Optimal Sampling Strategies for Multiscale Stochastic Processes

Vinay J. Ribeiro,[†] *Rudolf H. Riedi,*[†‡] *Richard G. Baraniuk* [†]

Department of Electrical and Computer Engineering[†] and Department of Statistics[‡]

Rice University, 6100 Main Street, Houston, TX 77005, USA

Abstract

This paper studies multiscale stochastic processes which are random processes organized on the nodes of a tree. The random variables at different levels on the tree represent time series of samples of a stochastic process at different temporal or spatial *scales*. We focus on classes of multiscale processes with additional statistical structure connecting scales and seek an optimal linear estimator of coarse scale nodes using an *incomplete* set of nodes at a finer time scale. We prove that the optimal solution for any tree with so-called independent innovations is readily given by a polynomial-time algorithm which we term the water-filling algorithm. The optimal solutions vary dramatically with the correlation structure of the multiscale process. For so-called scale-invariant trees and processes with positive correlation progression through scales, uniformly spaced leaves are optimal and clustered leaves are the worst possible. For processes with negative correlation progression, uniformly spaced leaves are the worst possible. Our results have implications for network traffic estimation, sensor network design, and environmental monitoring.

Keywords: multiscale stochastic processes, networks, long-range-dependence, sampling, optimal, trees, sensor networks

Supported by NSF grants ANI–9979465, ANI–0099148, and ANI-0338856, DoE SciDAC grant DE-FC02-01ER25462, DARPA/AFRL grant F30602–00–2–0557, Texas ATP 003604-0036-2003, and the Texas Instruments Leadership University program.

Email: {vinay, riedi, richb}@rice.edu; web: dsp.rice.edu, spin.rice.edu.

I. INTRODUCTION

A common theme in many practical applications is the estimation of the global average of some physical process (over space or time) from measured point samples of the process. Examples of such processes are environmental quantities such as temperature, pressure and pollution levels, agricultural quantities such as the insect population density, and network quantities such as data traffic loads [Pedigo & Buntin, 1993, Carter & Crovella, 1996].

Typically each sample has an associated measurement cost. Thus while increased sampling improves the accuracy of the estimate of the global average, it simultaneously increases the total measurement cost. In this paper we study solutions that optimally balance this trade-off in estimation accuracy versus cost. We limit our analysis to certain hierarchical statistical processes called *multiscale stochastic processes*.

A multiscale stochastic process consists of a set of univariate random variables which are organized like the nodes of a tree (see Fig. 1) [Willsky, 2002]. In typical scenarios, different levels on the tree represent different *scales* in time or space. Nodes at the bottom are called *leaves* and the topmost node is the *root*. We associate each node with the average of a physical process over some region with nodes at higher scales corresponding to larger regions. For example each leaf may correspond to the average temperature over a square mile and the root node to the average temperature over a thousand square miles.

Multiscale stochastic processes can vary in their *topology*, that is the number of nodes at each scale and the connectivity between nodes, as well as in the joint probability distribution of the nodes. We consider two types of multiscale stochastic processes in subsequent sections: *independent innovations trees* [Chou et al., 1994, Willsky, 2002] and *covariance trees* [Ma & Ji, 1998, Riedi et al., 1999]. In independent innovations trees, nodes at one scale are related to nodes at the next lower scale through independent additive innovations. In covariance trees pairs of leaf nodes with the same distance between them have the same covariance.

The question we address in this paper is: *Among all possible sets of leaves of size n, which set provides the best linear minimum mean squared error (LMMSE) for the root?* Equivalently, where must we strategically sample the process in order to estimate its global average optimally?

We describe an efficient $O(nN + R)$ algorithm (with N the tree-depth and R the total number of tree nodes) to determine an optimal leaf set of size n for independent innovations trees called *water-filling*. Exploiting key statistical properties of independent innovations trees we establish that water-filling is optimal. Note that the general problem of determining the n random variables that provide the best linear estimate of another random variable is NP-hard [Davis, 1994]. In contrast, our setting is one example of

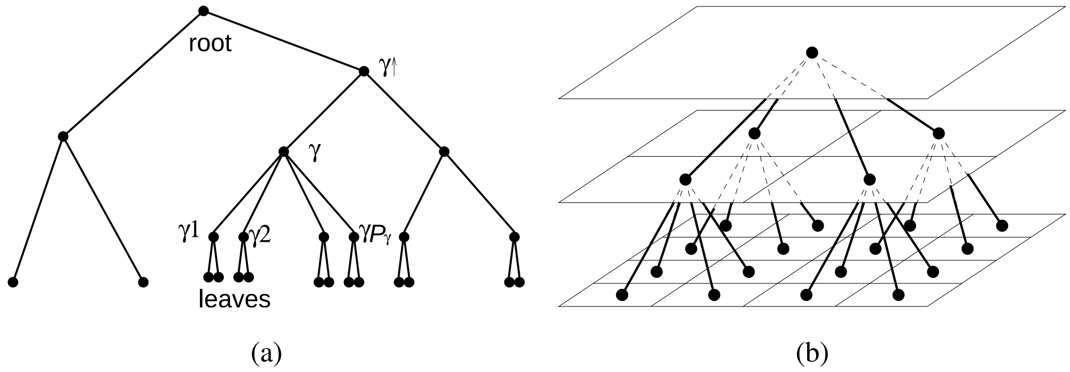


Fig. 1. *(a) A multiscale stochastic process. Nodes at the bottom are called leaves and the topmost node the root. (b) Multiscale quad-tree model of a spatial physical process. Nodes at lower levels in the tree correspond to averages of a physical process over smaller rectangular areas.*

this general problem which is solved in polynomial-time by the water-filling algorithm.

Using the water-filling algorithm we prove that uniformly spaced leaf nodes are optimal for certain independent innovations trees called scale-invariant trees as well as for covariance trees with positive correlation progression. The optimal solution can, however, change with the correlation structure of the tree. In fact for covariance trees with negative correlation progression we prove that uniformly spaced leaf nodes give the *worst* possible LMMSE! For covariance trees with positive correlation progression we prove that the set of clustered leaf nodes provides the worst possible LMMSE.

Our findings have implications in a range of different applications. One example is *sensor networks*. Such networks consist of electronic sensors that are distributed over a large region to measure some physical process. In the context of our work the area is subdivided hierarchically into sub-areas with the nodes of a tree representing the average of the physical process in different sub-areas (see Fig. 1(b)). Every sensor gives a local sample of the process which we model as a leaf node of the tree. We consider the case where the goal of the network is to measure the global average of the process which we model as the root of the tree. For economic reasons it is desirable to design sensor networks that use a small number of sensors, yet cover a large area with sufficiently accurate estimates [Willett et al., 2004]. An ideal design of such networks simply places sensors at locations corresponding to an optimal leaf set given a budget of number of sensors to be used.

Our results also have implications for techniques that infer internal properties of the Internet [Brownlee & kc claffy, 2004]. Such inference techniques are necessary because various technical and commercial factors render the direct measurement and dissemination of information about internal properties of the Internet prohibitive. One such technique uses the delays of injected *probe* packets to infer the volumes

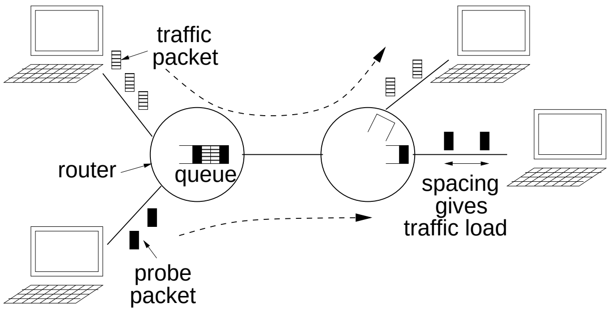


Fig. 2. *Probing for competing traffic. The spacing between probe packets is proportional to the volume of competing traffic arriving between them.*

of cross-traffic. Several networking applications benefit from cross-traffic inference including congestion control, streaming applications and admission control [Tuan & Park, 1999, Chou & Miao, 2001, Breslau et al., 2000]. Consider a simple network path consisting of two Y-junctions. At the first junction, the path to be probed joins with competing traffic at a router with constant service rate (see Fig. 2). At the second junction, the competing traffic separates again from the given path.

Our aim is to estimate the total competing traffic volume entering the queue in time interval $[0, \tau]$. We model this total amount over a large time interval as the root of a multiscale tree. As we inject two closely spaced probe packets, also known as a *packet-pair*, on the given path their spacing after the queue directly reflects the amount of competing traffic arriving between them. We model this amount arriving at a small time interval as a leaf node. Obviously one can estimate the total traffic in time interval $[0, \tau]$ by saturating the queue with packet-pairs for the entire time interval. This solution is highly impractical since we would use up the entire bandwidth with probe packets and congest the network! We are thus forced to make estimates using only a few probe packet-pairs. Our results provide the optimal spacing between packet-pairs.

The paper is organized as follows. Section II describes various multiscale stochastic processes used in the paper. In Section III we describe the water-filling technique to obtain optimal solutions for independent innovations trees. We then prove optimal and worst case solutions for covariance trees in Section IV. Through numerical experiments in Section V we demonstrate that optimal solutions for multiscale processes can vary depending on their topology and correlation structure. We describe related work on optimal sampling in Section VI. We summarize the paper and discuss future work in Section VII. The proofs and pseudo-code for water-filling can be found in the Appendices.

II. Multiscale Stochastic Processes

Trees occur naturally in many applications as an efficient data structure with simple dependence structure. Of particular interest are trees which arise from representing and analyzing stochastic processes and time series on different time scales. Examples include the trees of wavelet coefficients as well as the tree of scaling coefficients for the Haar wavelet [Willsky, 2002, Ma & Ji, 1998, Riedi et al., 1999]. In this section we describe various trees and related background material relevant to this paper.

A. Terminology and Notation

A tree is a special graph, i.e., a set of nodes together with a list of pairs of nodes which can be pictured as directed edges pointing from one node to another with the following special properties (see Fig. 1). (1) There is a unique node called the *root* to which no edge points to. (2) There is exactly one edge pointing to any node other than the root. The starting node of the edge is called *parent*, the end point is called *child*. (3) The tree is connected, meaning that it is possible to reach any node from the root by following edges.

These simple rules imply that there are no cycles in the tree, in particular, there is exactly one way to reach a node from the root. Otherwise, rule (2) would be violated. Consequently, unique addresses can be assigned to the nodes which also reflect the level of a node in the tree. The topmost node is the root whose address we denote by ø. Given an arbitrary node γ, its child nodes are said to be one level lower in the tree and are addressed by γk $(k = 1, 2, \ldots, P_\gamma)$, where $P_\gamma \geq 0$. Each node is thus a concatenation of the form ø$k_1 k_2 \ldots k_j$, or $k_1 k_2 \ldots k_j$ for short, where j is the node's *scale* or depth in the tree. The largest scale of any node in the tree is called the *depth* of the tree.

Nodes with no child nodes are termed *leaves* or *leaf nodes*. As usual, we denote the number of elements of a set of leaf nodes L by $|L|$. We define the operator $\uparrow$ such that $\gamma k \uparrow = \gamma$. Thus, the operator $\uparrow$ takes us one level higher in the tree to the parent of the current node. Nodes that can be reached by repeated $\uparrow$ operations are called *ancestors* of γ. We term γ a *descendant* of all of its ancestors.

The set of nodes and edges formed by γ and all its descendants is termed the *tree of* γ. Clearly, it satisfies all rules of a tree. Let L_γ denote the subset of L that belong to the tree of γ. Let $\mathcal{N}_\gamma$ be the total number of leaves of the tree of γ.

To every node γ we associate a single (univariate) random variable V_γ. The case where each node is associated with a multivariate random variable is beyond the scope of this paper. For the sake of brevity we often refer to V_γ as simply "the node V_γ" rather than "the random variable associated with node γ."

B. Covariance trees

Covariance trees are multiscale stochastic processes defined on the basis of the covariance between the leaf nodes which is purely a function of the *distance* between them. Examples of covariance trees are the Wavelet-domain Independent Gaussian model and the Multifractal Wavelet models proposed for network traffic [Ma & Ji, 1998, Riedi et al., 1999]. Precise definitions follow. We denote *variance*, *covariance*, and *expectation* by "var", "cov", and "$\mathbb{E}$" respectively.

Definition 1: *The* distance *between two leaf nodes is the scale of their lowest common ancestor.*

Definition 2: *A* covariance tree *is a multiscale stochastic process with two properties.*
(1) The covariance of any two leaf nodes depends only on their distance. In other words, if the leaves γ' and γ have distance l then

$$\operatorname{cov}(V_\gamma, V_{\gamma'}) =: c_l. \tag{1}$$

(2) All leaf nodes are at the same scale N and the root is equally correlated with all leaves.

In this paper we consider covariance trees of two classes: trees with positive correlation progression and trees with negative correlation progression.

Definition 3: *A covariance tree has a* positive correlation progression *if $c_l > c_{l-1} > 0$ for $l = 1, \ldots, N-1$. A covariance tree has a* negative correlation progression *if $c_l < c_{l-1}$ for $l = 1, \ldots, N-1$.* Intuitively in trees with positive correlation progression leaf nodes "closer" to each other in the tree are more strongly correlated than leaf nodes "farther apart."

Our results take on a special form for covariance trees that are also symmetric trees.

Definition 4: *A* symmetric tree *is a multiscale stochastic process in which P_γ, the number of child nodes of V_γ, is purely a function of the scale of γ.*

C. Independent innovations trees

Independent innovations trees are particular multiscale stochastic processes defined as follows.

Definition 5: *An* independent innovations tree *is a multiscale stochastic process in which each node V_γ, excluding the root, is defined through*

$$V_\gamma := \varrho_\gamma V_{\gamma\uparrow} + W_\gamma. \tag{2}$$

Here, ϱ_γ is a scalar and W_γ is a random variable independent of $V_{\gamma\uparrow}$ as well as of $W_{\gamma'}$ for all $\gamma' \neq \gamma$. The root node is independent of W_γ for all γ. In addition $\varrho_\gamma \neq 0$, $\operatorname{var}(W_\gamma) > 0\ \forall\gamma$ and $\operatorname{var}(V_\varnothing) > 0$ which together guarantee that $\operatorname{var}(V_\gamma) > 0\ \forall\gamma$ as well as the linear independence of any set of tree nodes.

The fact that each node is the sum of a scaled version of its parent and an independent random variable makes these trees amenable to analysis [Chou et al., 1994, Willsky, 2002]. We prove optimality results for independent innovations trees in Section III. Our results take on a special form for scale-invariant trees defined below.

Definition 6: *A* scale-invariant tree *is an independent innovations tree which is symmetric and where* ϱ_γ *and the distribution of* W_γ *are purely functions of the scale of* γ.

While independent innovations trees are not covariance trees in general, it is easy to see that scale-invariant trees are indeed covariance trees with positive correlation progression.

III. Optimal leaf sets for independent innovations trees

In this section we determine the optimal leaf sets of independent innovations trees to estimate the root. We first describe the concept of water-filling which we later use to prove optimality results. We also present an efficient numerical method to obtain the optimal solutions.

A. Water-filling

Concave optimization is a key ingredient of our analysis.

Definition 7: *A real function* ψ *defined on the set of integers* $\{0, 1, \ldots, M\}$ *is* concave *if*

$$\psi(x+1) - \psi(x) \geq \psi(x+2) - \psi(x+1), \text{ for } x = 0, \ldots, M-2. \tag{3}$$

The optimization problem we are faced with can be cast as follows. Given integers $P \geq 2$, $M_k > 0$ ($k = 1, \ldots, P$) and $n \leq \sum_{k=1}^{P} M_k$ consider the discrete space

$$\Delta_n(M_1, \ldots, M_P) := \left\{ X = [x_k]_{k=1}^{P} : \sum_{k=1}^{P} x_k = n, x_k \in \{0, \ldots, M_k\}, k = 1, 2, \ldots, P \right\}. \tag{4}$$

Given non-decreasing, concave functions ψ_k ($k = 1, \ldots, P$) with domains $\{0, \ldots, M_k\}$ we are interested in

$$h(n) := \max \left\{ \sum_{k=1}^{P} \psi_k(x_k) \; : \; X \in \Delta_n(M_1, \ldots, M_P) \right\}. \tag{5}$$

In the context of optimal estimation on a tree, P will play the role of the number of children that a parent node has, M_k the total number of leaf node descendants of the k-th child, and ψ_k the reciprocal of the optimal LMMSE of estimating the parent node given x_k leaf nodes in the tree of its k-th child.

The following iterative procedure constitutes a greedy algorithm that solves the optimization problem (5). Form vectors $G^{(n)} = [g_k^{(n)}]_{k=1}^{P}$, $n = 0, \ldots, \sum_k M_k$ as follows:

Step (i): Set $g_k^{(0)} = 0, \forall k$.

Step (ii): Set

$$g_k^{(n+1)} = \begin{cases} g_k^{(n)} + 1, & k = m \\ g_k^{(n)}, & k \neq m \end{cases} \tag{6}$$

where

$$m \in \arg\max_k \left\{ \psi_k \left(g_k^{(n)} + 1 \right) - \psi_k \left(g_k^{(n)} \right) : g_k^{(n)} < M_k \right\}. \tag{7}$$

The procedure described in Steps (i) and (ii) is termed *water-filling* because it resembles the solution to the problem of filling buckets with water to maximize the sum of the heights of the water levels. Buckets are narrow at the bottom and monotonously widen towards the top. Initially all buckets are empty (compare Step (i)). At each step we are allowed to pour one unit of water into any one bucket with the goal of maximizing the sum of water levels. Intuitively at any step we must pour the water into that bucket which will give the maximum increase in water level among all the buckets not yet full (compare Step (ii)). Variants of this water-filling procedure appear as solutions to different information theoretic and communication problems [Cover & Thomas, 1991].

Lemma 8: *The function $h(n)$ is non-decreasing and concave. In addition, $h(n) = \sum_k \psi_k \left(g_k^{(n)} \right)$ where $g_k^{(n)}$ is defined through water-filling.*

The proof is in the Appendix A.

When all functions ψ_k in Lemma 8 are identical, the maximum of $\sum_{k=1}^{P} \psi_k(x_k)$ is achieved by choosing the x_k's to be "near-equal". The following Corollary states this rigorously.

Corollary 9: *If $\psi_k = \psi$ for all $k = 1, 2, \ldots, P$ with ψ non-decreasing and concave, then*

$$h(n) = \left(P - n + P \left\lfloor \frac{n}{P} \right\rfloor \right) \psi \left(\left\lfloor \frac{n}{P} \right\rfloor \right) + \left(n - P \left\lfloor \frac{n}{P} \right\rfloor \right) \psi \left(\left\lfloor \frac{n}{P} \right\rfloor + 1 \right). \tag{8}$$

The maximizing values of the x_k are apparent from (8). In particular, if n is a multiple of P then this reduces to

$$h(n) = P\psi \left(\frac{n}{P} \right). \tag{9}$$

The proof is in Appendix A.

Corollary 9 is key to proving our results for scale-invariant trees.

B. Optimal leaf sets through recursive water-filling

Our goal is to determine the choice of n leaf nodes that gives the best linear minimum mean-square error (LMMSE) of the root. The LMMSE of V_γ given L_γ is defined as

$$\mathcal{E}(V_\gamma|L_\gamma) := \min_{\boldsymbol{\alpha}} \mathbb{E}(V_\gamma - \boldsymbol{\alpha}^T L_\gamma)^2, \tag{10}$$

where, in an abuse of notation, $\boldsymbol{\alpha}^T L_\gamma$ denotes a linear combination of the elements of L_γ with coefficients $\boldsymbol{\alpha}$. Crucial to our proofs is the fact that [Chou et al., 1994, Willsky, 2002]

$$\frac{1}{\mathcal{E}(V_\gamma|L_\gamma)} + \frac{P_\gamma - 1}{\mathrm{var}(V_\gamma)} = \sum_{k=1}^{P_\gamma} \frac{1}{\mathcal{E}(V_\gamma|L_{\gamma k})}. \tag{11}$$

Denote the set consisting of all subsets of leaves of the tree of γ of size n by $\Lambda_\gamma(n)$. Motivated by (11) we introduce

$$\mu_\gamma(n) := \max_{L \in \Lambda_\gamma(n)} \mathcal{E}^{-1}(V_\gamma|L) \tag{12}$$

and define

$$\mathcal{L}_\gamma(n) := \{L \in \Lambda_\gamma(n) : \mathcal{E}^{-1}(V_\gamma|L) = \mu_\gamma(n)\}. \tag{13}$$

Restated, our goal is to determine one element of $\mathcal{L}_\emptyset(n)$.

To allow a recursive approach through scale we generalize (12) and (13) by defining

$$\mu_{\gamma,\gamma'}(n) := \max_{L \in \Lambda_{\gamma'}(n)} \mathcal{E}^{-1}(V_\gamma|L). \tag{14}$$

and

$$\mathcal{L}_{\gamma,\gamma'}(n) := \{L \in \Lambda_{\gamma'}(n) : \mathcal{E}^{-1}(V_\gamma|L) = \mu_{\gamma,\gamma'}(n)\}. \tag{15}$$

Of course, $\mathcal{L}_\gamma(n) = \mathcal{L}_{\gamma,\gamma}(n)$. For the recursion, we are mostly interested in $\mathcal{L}_{\gamma,\gamma k}(n)$, i.e., the optimal estimation of a parent node from a sample of leaf nodes of one of its children. The following will be useful notation

$$X^* := \arg \max_{X \in \Delta_n(\mathcal{N}_{\gamma 1}, \ldots, \mathcal{N}_{\gamma P_\gamma})} \sum_{k=1}^{P_\gamma} \mu_{\gamma,\gamma k}(x_k). \tag{16}$$

Using (11) we can decompose the problem of determining $L \in \mathcal{L}_\gamma(n)$ into smaller problems of determining $L^{(k)} \in \mathcal{L}_{\gamma,\gamma k}(x_k^*)$ as stated in the next theorem.

Theorem 10: *For an independent innovations tree with $X^* = [x_k^*]_{k=1}^{P_\gamma}$ defined as in (16), $\bigcup_{k=1}^{P_\gamma} L^{(k)} \in \mathcal{L}_\gamma(n)$. Moreover, $\mathcal{L}_{\gamma k}(n) = \mathcal{L}_{\gamma k,\gamma k}(n) = \mathcal{L}_{\gamma,\gamma k}(n)$. Also $\mu_{\gamma,\gamma k}(n)$ is a positive, non-decreasing, and concave function of n, $\forall k, \gamma$.*

The proof is in the Appendix A.

Theorem 10 provides the following recursive technique to construct $L \in \mathcal{L}_\gamma(n)$. Starting at γ we move downward determining how many of the n leaf nodes of $L \in \mathcal{L}_\gamma(n)$ lie in the trees of the different descendants of γ till we reach the bottom. Assume for the moment that we are given the functions $\mu_{\gamma,\gamma k}(n)$ for all γ. We present an efficient scheme to compute $\mu_{\gamma,\gamma k}(n)$ in Appendix C.

Scale-Recursive Water-filling scheme $\gamma \to \gamma k$

Step (a): Split n leaf nodes between the trees of γk, $k = 1, 2, \ldots, P_\gamma$.

First determine how to split the n leaf nodes between the trees of γk by maximizing $\sum_{k=1}^{P_\gamma} \mu_{\gamma,\gamma k}(x_k)$ over $X \in \Delta_n(\mathcal{N}_{\gamma 1}, \ldots, \mathcal{N}_{\gamma P_\gamma})$ (see (16)). The split is given by X^* which is easily obtained using the water-filling procedure for concave functions (defined in (6)) since $\mu_{\gamma,\gamma k}(n)$ is concave for all k. Determine $L^{(k)} \in \mathcal{L}_{\gamma,\gamma k}(x_k^*)$ since $L = \bigcup_{k=1}^{P_\gamma} L^{(k)} \in \mathcal{L}_\gamma(n)$.

Step (b): Split x_k^* nodes between the trees of child nodes of γk.

To our good fortune $L^{(k)} \in \mathcal{L}_{\gamma,\gamma k}(x_k^*)$ if and only if $L^{(k)} \in \mathcal{L}_{\gamma k}(x_k^*)$. Thus repeat Step (a) with $\gamma = \gamma k$ and $n = x_k^*$ to construct $L^{(k)}$. Stop when we have reached the bottom of the tree.

Efficient Implementation

We present an efficient implementation of the scale-recursive water-filling algorithm in Appendix C. This implementation first computes $L \in \mathcal{L}_\gamma(n)$ for $n = 1$ and then inductively obtains the same for larger values of n. Given $L \in \mathcal{L}_\gamma(n)$ we obtain $L \in \mathcal{L}_\gamma(n+1)$ as follows. Note from Step (a) above that we use water-filling to determine how to split the n leaves at γ. We are now required to split $n+1$ leaves at γ. We easily obtain this from the earlier split of n leaves using (6). The water-filling technique maintains the split of n leaf nodes at γ while adding just one leaf node to the tree of one of the child nodes (say $\gamma k'$) of γ. We thus have to perform Step (b) only for $k = k'$. In this way the new leaf node "percolates" down the tree till we find its location at the bottom of the tree. The pseudo-code for determining $L \in \mathcal{L}_\gamma(n)$ given var(W_γ) for all γ as well as the proof that the recursive water-filling algorithm is polynomial-time is in Appendix C.

C. Uniform leaf nodes are optimal for scale-invariant trees

The symmetry in scale-invariant trees forces the optimal solution to take a particular form irrespective of the variances of the innovations W_γ. We use the following notion of uniform split to prove that in a scale-invariant tree a more or less equal spread of sample leaf nodes gives the best linear estimate of the root.

Definition 11: *Given a scale-invariant tree, a vector of leaf nodes L has* uniform split *of size n at node γ if $|L_\gamma| = n$ and $|L_{\gamma k}|$ is either $\lfloor \frac{n}{P_\gamma} \rfloor$ or $\lfloor \frac{n}{P_\gamma} \rfloor + 1$ for all values of k. It follows that $\#\{k : |L_{\gamma k}| = \lfloor \frac{n}{P_\gamma} \rfloor + 1\} = n - P_\gamma \lfloor \frac{n}{P_\gamma} \rfloor$.*

Definition 12: *Given a scale-invariant tree, a vector of leaf nodes is called a* uniform leaf sample *if it has a uniform split at all tree nodes.*

The next theorem gives the optimal leaf node set for scale-invariant trees.

Theorem 13: *Given a scale-invariant tree, the uniform leaf sample of size n gives the best LMMSE estimate of the tree-root among all possible choices of n leaf nodes.*

Proof: For a scale-invariant tree, $\mu_{\gamma,\gamma k}(n)$ is identical for all k given any location γ. Corollary 9 and Theorem 10 then prove the theorem. $\square$

IV. Covariance Trees

In this section we prove optimal and worst case solutions for covariance trees. For the optimal solutions we leverage our results for independent innovations trees and for the worst case solutions we employ eigenanalysis. We begin by formulating the problem.

A. Problem formulation

Let us compute the LMMSE error of estimating the root $V_\emptyset$ given a set of leaf nodes L of size n. Because of our assumption of a covariance tree, the correlation between any leaf node and the root node is identical. We denote this by ρ. Denote an $l \times j$ matrix with all elements equal to 1 by $\mathbf{1}_{l\times j}$. It is well known [Stark & Woods, 1986] that the optimal linear estimate of $V_\emptyset$ given L (assuming zero-mean random variables) is given by $\rho \mathbf{1}_{1\times n} S_L^{-1} L$, where S_L is the covariance matrix of L and that the resulting LMMSE is

$$\begin{aligned}\mathcal{E}(V_\emptyset|L) &= \mathrm{var}(V_\emptyset) - \mathrm{cov}(L, V_\emptyset)^T S_L^{-1} \mathrm{cov}(L, V_\emptyset) \\ &= \mathrm{var}(V_\emptyset) - \rho^2 \mathbf{1}_{1\times n} S_L^{-1} \mathbf{1}_{n\times 1}. \qquad (17)\end{aligned}$$

Clearly obtaining the optimal and worst case choice for L is equivalent to maximizing and minimizing the sum of the elements of S_L^{-1}. The exact value of ρ does not affect the solution. We assume that no element of L can be expressed as a linear combination of the others which implies that S_L in invertible.

B. Optimal solutions

We use our results of Section III for independent innovations trees to determine the optimal solutions for covariance trees. Note from (17) that the estimation error for a covariance tree is a function only of the covariance *between leaf nodes*. Exploiting this fact, we first construct an independent innovations tree whose leaf nodes have the same correlation structure as that of the covariance tree and then prove that both trees must have the same optimal solution. Previous results then provide the optimal solution for the independent innovations tree which is also the optimal for the covariance tree.

Definition 14: *A* matched innovations tree *of a given covariance tree with positive correlation progression is an independent innovations tree with the following properties. It has (1) the same topology (2) and the same correlation structure between leaf nodes as the covariance tree, and (3) the root is equally correlated with all leaf nodes (though the exact value of the correlation between the root and a leaf node may differ from that of the covariance tree).*

All covariance trees with positive correlation progression have corresponding matched innovations trees. We construct a matched innovations tree for a given covariance tree as follows. Consider an independent innovations tree with the same topology as the covariance tree. Set $\varrho_\gamma = 1$ for all γ,

$$\mathrm{var}(V_\emptyset) = c_0, \tag{18}$$

and

$$\mathrm{var}(W^{(j)}) = c_j - c_{j-1},\ j = 1, 2, \dots, N, \tag{19}$$

where c_j is the covariance of leaf nodes of the covariance tree with distance j and $\mathrm{var}(W^{(j)})$ is the common variance of all innovations of the independent innovations tree at scale j. Call c'_j the covariance of leaf nodes with distance j in the independent innovations tree. From (2) we have

$$c'_j = \mathrm{var}(V_\emptyset) + \sum_{k=1}^{j} \mathrm{var}\left(W^{(k)}\right),\ \ j = 1, \dots, N. \tag{20}$$

We thus obtain $c'_j = c_j$ for all j which proves that this independent innovations tree is the required matched innovations tree.

The next lemma relates the optimal solutions of a covariance tree and its matched innovations tree.

Lemma 15: *A covariance tree with positive correlation progression and its matched innovations tree have the same optimal leaf sets.*

Proof: Note that (17) applies to any tree whose root is equally correlated with all its leaves. This includes both the covariance tree and its matched innovations tree. From (17) we see that the choice of L that maximizes the sum of elements of S_L^{-1} is optimal. Since S_L^{-1} is identical for both the covariance tree and its matched innovations tree for any choice of L, they must have the same optimal solution. □

For a symmetric covariance tree that has positive correlation progression, the optimal solution takes on a specific form irrespective of the actual covariance between leaf nodes.

Theorem 16: *Given a symmetric covariance tree that has positive correlation progression, the uniform leaf sample of size n gives the best LMMSE of the tree-root among all possible choices of n leaf nodes.*

Proof: Form a matched innovations tree using the procedure outlined previously. This tree is by construction a scale-invariant tree. The result then follows from Theorem 13 and Lemma 15. □

While the uniform leaf sample is the optimal solution for a symmetric covariance tree with positive correlation progression, it is surprisingly the worst case solution for certain trees with a different correlation structure which we prove next.

C. Worst case solutions

The *worst case solution* is any choice of $L \in \Lambda_{\emptyset}(n)$ that maximizes $\mathcal{E}(V_{\emptyset}|L)$. We now highlight the fact that the optimal and worst case solutions can change dramatically depending on the correlation structure of the tree. Of particular relevance to our discussion is the set of *clustered leaf nodes* defined as follows.

Definition 17: *The set consisting of all leaf nodes of the tree of V_γ is called the set of* clustered leaves *of γ.*

We provide the worst case solutions for covariance trees in which every node (with the exception of the leaves) has the same number of child nodes. The following theorem summarizes our results.

Theorem 18: *Consider a covariance tree of depth N in which every node (excluding the leaves) has the same number of child nodes σ. Then for leaf sets of size σ^p, $p = 0, 1, \ldots, N$ the worst case solution when the tree has positive correlation progression is given by the sets of clustered leaves of γ, where γ is at scale $N - p$. The worst case solution is given by the sets of uniform leaf nodes when the tree has negative correlation progression.*

The proof is in Appendix B.

Theorem 18 gives us the intuition that "more correlated" leaf nodes give worse estimates of the root. In the case of covariance trees with positive correlation progression, clustered leaf nodes are strongly correlated when compared to uniform leaf nodes. The opposite is true in the negative correlation progression case. Essentially if leaf nodes are highly correlated then they contain more redundant information which leads to poor estimation of the root.

While we have proved the optimal solution for covariance trees with positive correlation progression we have not yet proved the same for those with negative correlation progression. Based on the intuition just gained we make the following conjecture.

Conjecture 19: For a covariance tree that is negatively correlated, the sets of clustered leaf nodes correspond to the optimal solution.

Using numerical techniques we support this conjecture in the next section.

V. NUMERICAL RESULTS

In previous sections we proved that uniform leaf nodes were optimal for scale-invariant trees. In this section, using the scale-recursive water-filling algorithm we evaluate the optimal leaf sets for independent innovations trees that are not scale-invariant. In addition we provide numerical support for Conjecture 19.

A. Independent innovations trees: scale-recursive water-filling

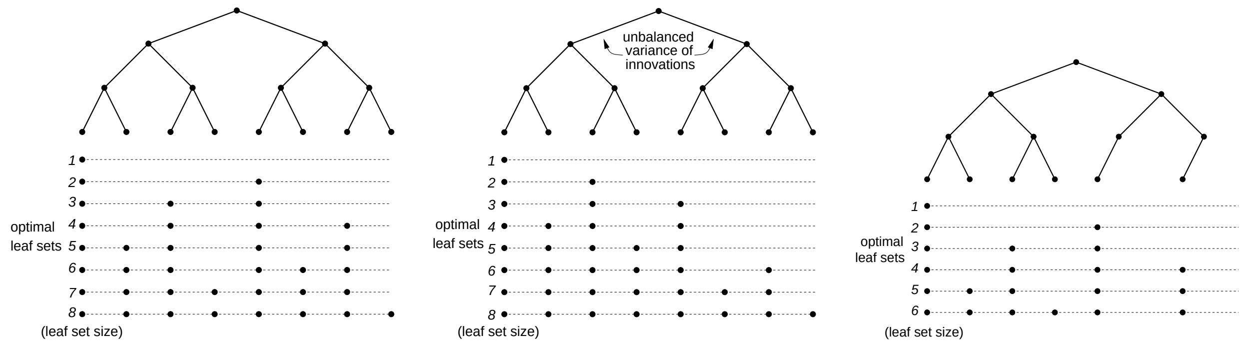
In this section we find the optimal leaf nodes to estimate the tree root for different independent innovations trees using the water-filling algorithm described in Section III. We consider trees with depth $N = 3$ and in which all nodes have at most two child nodes. The results demonstrate that the optimal leaf sets are a function of the correlation structure and topology of the multiscale trees.

In Fig. 3(a) we plot the optimal leaf node sets of different sizes for a scale-invariant tree. As expected the uniform leaf nodes sets are optimal.

We consider a symmetric tree in Fig. 3(b), that is a tree in which all nodes have the same number of children (excepting leaf nodes). All parameters are constant within each scale except for the variance of the innovations W_γ at scale 1. The variance of the innovation on the right side is five times larger than the variance of the innovation on the left. Observe that leaves on the left of the tree are now preferable to those on the right and hence dominate the optimal sets. Comparing this result to Fig. 3(a) we see that the optimal sets are dependent on the correlation structure of the tree.

In Fig. 3(c) we consider the same tree as in Fig. 3(a) with two leaf nodes missing. These two leaves do not belong to the optimal leaf sets of size $n = 1$ to $n = 5$ in Fig. 3(a) but are elements of the optimal sets for $n = 6$ to 8. As a result the optimal sets of size 1 to 5 in Fig. 3(c) are identical to those in Fig. 3(a) whereas that for $n = 6$ differs. This result demonstrates that the optimal sets depend on the topology of the tree.

Our results have important implications for practical applications because situations arise in practice



(a) Scale-invariant tree (b) Tree with unbalanced variance (c) Tree with missing leaves

Fig. 3. *Optimal leaf node sets for three different independent innovations trees: (a) scale-invariant tree, (b) symmetric tree with unbalanced variance of innovations at scale* 1, *and (c) tree with missing leaves at the finest scale. Observe that the uniform leaf node sets are optimal in (a) as expected. In (b), however, the nodes on the left half of the tree are more preferable to those on the right. In (c) the solution is similar to (a) for optimal sets of size* $n = 5$ *or lower but changes for* $n = 6$ *due to the missing nodes.*

where we must model physical processes using trees with different correlation structures and topologies. For example, if the process to be measured is non-stationary over space then the multiscale tree may be unbalanced as in Fig. 3(b). In sensor networks it may not be possible to place sensors in certain locations due to physical constraints. We would thus have to exclude certain leaf nodes in our analysis as in Fig. 3(c).

B. Covariance trees: Optimal and worst cases

This section provides numerical support for Conjecture 19 that states that the clustered node sets are optimal for covariance trees with negative correlation progression. We employ the wavelet-domain independent Gaussian (WIG) tree which is a covariance tree in which each node has $\sigma = 2$ child nodes [Ma & Ji, 1998]. We verify our claim using a WIG model of depth $N = 6$ possessing an fractional Gaussian noise-like correlation structure corresponding to Hurst parameter $H = 0.8$ and $H = 0.3$. To be precise, we choose the WIG model parameters such that the variance of nodes at scale j is $2^{-2jH} \cdot \mathrm{constant}$ (see [Ma & Ji, 1998] for further details). Note that $H > 0.5$ corresponds to positive correlation progression while $H \leq 0.5$ corresponds to negative correlation progression.

Fig. 4 compares the LMMSE of the estimated root node (normalized by the variance of the root) of the uniform and clustered sampling patterns. Since an exhaustive search of all possible patterns is computationally expensive (for example there are over 10^{18} ways of choosing 32 leaf nodes from among 64) we instead compute the LMMSE for 10^4 randomly selected patterns. Observe that the clustered pattern gives the smallest LMMSE for the tree with negative correlation progression in Fig. 4(a) supporting our Conjecture 19 while the uniform pattern gives the smallest LMMSE for the positively correlation progression one in Fig. 4(b) corroborating Theorem 16. As proved in Theorem 18, the clustered and uniform patterns give the worst LMMSE for the positive and negative correlation progression cases respectively.

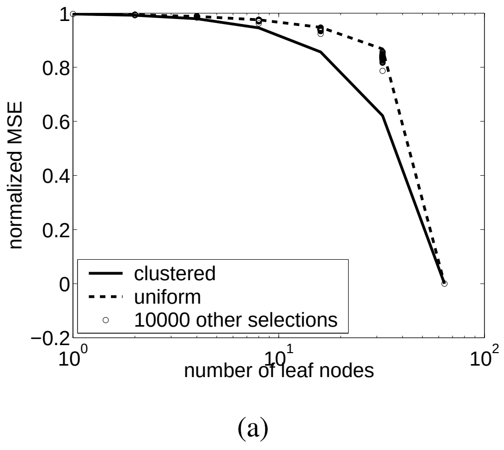


Fig. 4. *Comparison of probing schemes for a WIG model with (a) negative correlation progression and (b) positive correlation progression. Observe that the clustered nodes are optimal in (a) while the uniform is optimal in (b). The uniform and the clustered sets give the worst performance in (a) and (b) respectively, which confirms our theoretical results.*

Our results confirm that the optimal solution is strongly dependent on the correlation structure of the tree. In many applications such as Internet traffic sampling using probe packets, one expects a positive correlation progression. Thus schemes that send a stream of back-to-back probe packets will give poor estimates of the average traffic rate. However in exceptional cases where the traffic could possess negative correlation progression, we conjecture that the same schemes will be optimal!

VI. Related Work

Earlier work has studied the problem of designing optimal samples of size n to linearly estimate the global average of a process. For a one dimensional process which is wide-sense stationary with positive and convex correlation it was shown that systematic sampling (uniform patterns with random starting points) is optimal within a class of unbiased estimators [Hájek, 1959].

For a two dimensional process on an $n_1 \times n_2$ grid with positive and convex correlation it was shown that an optimal sampling scheme does not lie in the class of schemes that ensure equal inclusion probability of

$n/(n_1 n_2)$ for every point on the grid [Bellhouse, 1977]. Here an "optimal scheme" refers to a sampling scheme that achieves a lower bound on the error variance. The requirement of equal inclusion probability guarantees an unbiased estimator. The optimal schemes *within* certain sub-classes of this larger "equal inclusion probability" class were obtained using systematic sampling. More recent analysis refines these results to show that optimal designs do exist in the equal inclusion probability class for certain values of n, n_1, and n_2 and are obtained by Latin square sampling [Lawry & Bellhouse, 1992, Salehi, 2004].

Our results differs from the above works in that we provide optimal solutions for the entire class of linear estimators and study a different set of random processes.

A related problem of optimally sampling a process to minimize the entropy of the posterior process has been studied in the context of computer experiment design [Xu, 1999]. Two conditions that are sufficient for a sample to be *universally optimal* are that its elements are pair-wise equidistant and that this distance equal the maximum average distance of all possible sampling designs.

Other work on sampling fractional Brownian motion to estimate its Hurst parameter demonstrated that geometric sampling is superior to uniform sampling [Vidàcs & Virtamo, 1999].

Recent work compared different probing schemes for traffic estimation through numerical simulations [He & Hou, 2003]. It was shown that a scheme which used uniformly spaced probes outperformed other schemes that used clustered probes. These results are similar to our findings for independent innovation trees and covariance trees with positive correlation progression.

VII. CONCLUSIONS

This paper has addressed the problem of obtaining optimal leaf sets to estimate the root of two multi-scale stochastic processes: independent innovations trees and covariance trees. Our findings have a wide range of applications including the design of sensor networks and Internet inference schemes.

We have proved for an independent innovations tree that the optimal solution can be obtained in polynomial-time by the water-filling algorithm. Our results show that the optimal solutions can vary drastically depending on the correlation structure of the tree. For covariance trees with positive correlation progression as well as scale-invariant trees we obtained that uniformly spaced leaf nodes are optimal. However, uniform leaf nodes give the worst estimates for covariance trees with negative correlation progression. Numerical experiments support our conjecture that clustered nodes provide the optimal solution for covariance trees with negative correlation progression.

This paper raises several interesting questions for future research. While the general problem of determining the n best random variables to linearly estimate another random variable is an NP-hard problem, we devised a polynomial-time algorithm to solve one problem of this type, namely determining the optimal leaf set for an independent innovations tree. Clearly, the structure of independent innovations trees was an important factor that enabled a fast algorithm. The question arises as to whether there are similar problems that have polynomial-time solutions.

We have proved optimal results for covariance trees by reducing the problem to one for independent innovations trees. Such techniques of reducing one optimization problems to another problem that has an efficient solution can be very powerful. If a problem can be reduced to one of determining optimal leaf sets for independent innovations trees in polynomial-time, then its solution is also polynomial-time. Which other problems are malleable to this reduction is an open question.

Sensor networks face several challenging research problems. We have only partially solved the issues of sensor placement. Ideally this problem must be solved in conjunction with other important problems such as developing optimal communication strategies and protocols.

Inference of the Internet's internal properties is a rapidly developing field. While cross-traffic inference fits into our paradigm of estimation on trees, other inference problems require a more general framework. For example, a network operator may want to know the optimal n locations in his network to capture detailed traffic traces. Detailed traffic traces are, however, too complicated to be represented by a single random variable on a tree as in our paradigm. Generalizing our theory to arbitrary statistical models and optimality criteria is part of future work.

APPENDIX A

Proof of Lemma 8

We first prove the following statement.

Claim (1): If there exists $X^* = [x_k^*] \in \Delta_n(M_1, \ldots, M_P)$ that has the following property:

$$\psi_i(x_i^*) - \psi_i(x_i^* - 1) \geq \psi_j(x_j^* + 1) - \psi_j(x_j^*), \tag{21}$$

$\forall i \neq j$ such that $x_i^* > 0$ and $x_j^* < M_j$, then

$$h(n) = \sum_{k=1}^{P} \psi_k(x_k^*). \tag{22}$$

We then prove that such an X^* always exists and can be constructed using the water-filling technique.

Consider any $\widehat{X} \in \Delta_n(M_1, \ldots, M_P)$. Using the following steps, we transform the vector $\widehat{X}$ two elements at a time to obtain X^*.

Step 1: (Initialization) Set $X = \widehat{X}$.

Step 2: If $X \neq X^*$ then there exists a pair i, j such that $x_i \neq x_i^*$ and $x_j \neq x_j^*$ since the elements of both X and X^* sum up to n. Without loss of generality assume that $x_i < x_i^*$ and $x_j > x_j^*$. This assumption implies that $x_i^* > 0$ and $x_j^* < M_j$. Now form vector Y such that

$$\begin{aligned} y_i &= x_i + 1 \\ y_j &= x_j - 1 \\ y_k &= x_k, \ \ (k \neq i, j). \end{aligned} \tag{23}$$

From (21) and the concavity of ψ_i and ψ_j we have

$$\begin{aligned} \psi_i(y_i) - \psi_i(x_i) &= \psi_i(x_i + 1) - \psi_i(x_i) \\ &\geq \psi_i(x_i^*) - \psi_i(x_i^* - 1) \\ &\geq \psi_j(x_j^* + 1) - \psi_j(x_j^*) \\ &\geq \psi_j(x_j) - \psi_j(x_j - 1) \\ &\geq \psi_j(x_j) - \psi_j(y_j). \end{aligned} \tag{24}$$

As a consequence

$$\begin{aligned} \textstyle\sum_k (\psi_k(y_k) - \psi_k(x_k)) &= \psi_i(y_i) - \psi_i(x_i) + \psi_j(y_j) \\ &\quad -\psi_j(x_j) \\ &\geq 0. \end{aligned} \tag{25}$$

Step 3: If $Y \neq X^*$ then set $X = Y$ and repeat Step 2, otherwise stop.

After performing the above steps at most $\sum_k M_k$ times, $Y = X^*$ and (25) gives

$$\sum_k \psi_k(x_k^*) = \sum_k \psi_k(y_k) \geq \sum_k \psi_k(\widehat{x}_k). \tag{26}$$

This proves Claim (1).

Indeed for any $\widetilde{X} \neq X^*$ satisfying (21) we must have $\sum_k \psi_k(\widetilde{x}_k) = \sum_k \psi_k(x_k^*)$.

We now prove the following claim by induction.

Claim (2): $G^{(n)} \in \Delta_n(M_1, \ldots, M_P)$ and that $G^{(n)}$ satisfies (21).

(Initial Condition) The claim is trivial for $n = 0$.

(Induction Step) Clearly from (6) and (7)

$$\sum_k g_k^{(n+1)} = 1 + \sum_k g_k^{(n)} = n + 1, \tag{27}$$

and $0 \leq g_k^{(n+1)} \leq M_k$. Thus $G^{(n+1)} \in \Delta_{n+1}(M_1, \ldots, M_P)$. We now prove that $G^{(n+1)}$ satisfies property (21). We need to consider pairs i, j as in (21) for which either $i = m$ or $j = m$ because all other cases directly follow from the fact that $G^{(n)}$ satisfies (21).

Case (i) $j = m$, where m is defined as in (7). Assuming that $g_m^{(n+1)} < M_m$, for all $i \neq m$ such that $g_i^{(n+1)} > 0$ we have

$$\begin{aligned}\psi_i\left(g_i^{(n+1)}\right) - \psi_i\left(g_i^{(n+1)} - 1\right)&\\ &= \psi_i\left(g_i^{(n)}\right) - \psi_i\left(g_i^{(n)} - 1\right)\\ &\geq \psi_m\left(g_m^{(n)} + 1\right) - \psi_m\left(g_m^{(n)}\right)\\ &\geq \psi_m\left(g_m^{(n)} + 2\right) - \psi_m\left(g_m^{(n)} + 1\right)\\ &= \psi_m\left(g_m^{(n+1)} + 1\right) - \psi_m\left(g_m^{(n+1)}\right). \qquad (28)\end{aligned}$$

Case (ii) $i = m$. Consider $j \neq m$ such that $g_j^{(n+1)} < M_j$. We have from (7) that

$$\begin{aligned}\psi_m\left(g_m^{(n+1)}\right) - \psi_m\left(g_m^{(n+1)} - 1\right)&\\ &= \psi_m\left(g_m^{(n)} + 1\right) - \psi_m\left(g_m^{(n)}\right)\\ &\geq \psi_j\left(g_j^{(n)} + 1\right) - \psi_j\left(g_j^{(n)}\right)\\ &= \psi_j\left(g_j^{(n+1)} + 1\right) - \psi_j\left(g_j^{(n+1)}\right). \qquad (29)\end{aligned}$$

Thus Claim (2) is proved.

It only remains to prove the next claim.

Claim (3): $h(n)$, or equivalently $\sum_k \psi_k(g_k^{(n)})$, is non-decreasing and concave.

Since ψ_k is non-decreasing for all k, from (6) we have that $\sum_k \psi_k(g_k^{(n)})$ is a non-decreasing function of n. We have from (7)

$$\begin{aligned} h(n+1) - h(n) &= \sum_k \left(\psi_k(g_k^{(n+1)}) - \psi_k(g_k^{(n)})\right)\\ &= \max_{k: g_k^{(n)} < M_k} \left\{\psi_k(g_k^{(n)} + 1) - \psi_k(g_k^{(n)})\right\}. \qquad (30)\end{aligned}$$

From the concavity of ψ_k and the fact that $g_k^{(n+1)} \geq g_k^{(n)}$ we have that

$$\psi_k(g_k^{(n)} + 1) - \psi_k(g_k^{(n)}) \geq \psi_k(g_k^{(n+1)} + 1) - \psi_k(g_k^{(n+1)}), \qquad (31)$$

for all k. Thus from (30) and (31), $h(n)$ is concave. □

Proof of Corollary 9

Set $x_k^* = \left\lfloor \frac{n}{P} \right\rfloor$ for $1 \leq k \leq P - n + P\left\lfloor \frac{n}{P} \right\rfloor$ and $x_k^* = 1 + \left\lfloor \frac{n}{P} \right\rfloor$ for all other k. Then $X^* = [x_k^*] \in \Delta_n(M_1, \ldots, M_P)$ and X^* satisfies (21) from which the result follows. □

The following two lemmas are required to prove Theorem 10.

Lemma 20: *Given independent random variables A, W, F, define B and C through $B := \zeta A + W$ and $C := \eta B + F$ where ζ, η are constants. We then have the result*

$$\frac{\text{var}(A)}{\text{cov}(A,C)^2} \cdot \frac{\text{cov}(B,C)^2}{\text{var}(B)} = \frac{\zeta^2 + \text{var}(W)/\text{var}(A)}{\zeta^2} \geq 1. \tag{32}$$

Proof: Without loss of generality assume all random variables have zero mean. We have

$$\begin{aligned} \text{cov}(C,B) &= \mathbb{E}(\eta B^2 + FB) \\ &= \eta \text{var}(B), \end{aligned} \tag{33}$$

$$\begin{aligned} \text{cov}(A,C) &= \mathbb{E}((\eta(\zeta A + W) + F)A) \\ &= \mathbb{E}(\zeta\eta A^2 + \eta W A + FA) \\ &= \zeta\eta \text{var}(A), \end{aligned} \tag{34}$$

and

$$\text{var}(B) = \mathbb{E}(\zeta^2 A^2 + W^2 + 2\zeta A W) = \zeta^2 \text{var}(A) + \text{var}(W). \tag{35}$$

Thus from (33), (34) and (35)

$$\begin{aligned} \frac{\text{cov}(B,C)^2}{\text{var}(B)} \cdot \frac{\text{var}(A)}{\text{cov}(A,C)^2} &= \frac{\eta^2 \text{var}(B)}{\zeta^2 \eta^2 \text{var}(A)} \\ &= \frac{\zeta^2 + \text{var}(W)/\text{var}(A)}{\zeta^2} \\ &\geq 1. \end{aligned} \tag{36}$$

$\square$

Lemma 21: *Given a positive function $\rho_i, i \in \mathbb{Z}$ and constant $\alpha > 0$ such that*

$$r_i := \frac{1}{1 - \alpha \rho_i} \tag{37}$$

is positive, concave, and non-decreasing, we have that

$$q_i := \frac{1}{1 - \beta \rho_i} \tag{38}$$

is also positive, concave, and non-decreasing for all β such that $0 < \beta \leq \alpha$.

Proof: Define $\kappa_i := \rho_i - \rho_{i-1}$. Since ρ_i is positive and r_i is positive and non-decreasing, $\alpha\rho_i < 1$ and ρ_i must increase with i, that is $\kappa_i \geq 0$. This combined with the fact that $\beta\rho_i \leq \alpha\rho_i < 1$ guarantees that q_i must be positive and non-decreasing.

It only remains to prove the concavity of q_i. From (37)

$$\begin{aligned} r_{i+1} - r_i &= \frac{1}{1-\alpha\rho_{i+1}} - \frac{1}{1-\alpha\rho_i} \\ &= \frac{\alpha(\rho_{i+1}-\rho_i)}{(1-\alpha\rho_{i+1})(1-\alpha\rho_i)} \\ &= \alpha\kappa_{i+1}r_{i+1}r_i. \end{aligned} \tag{39}$$

We are given that r_i is concave, that is

$$\begin{aligned} 0 &\geq (r_{i+2}-r_{i+1}) - (r_{i+1}-r_i) \\ &= \alpha r_i r_{i+1}\left[\kappa_{i+2}\left(\frac{1-\alpha\rho_i}{1-\alpha\rho_{i+2}}\right) - \kappa_{i+1}\right]. \end{aligned} \tag{40}$$

Since $r_i > 0\ \forall i$, we must have

$$\left[\kappa_{i+2}\left(\frac{1-\alpha\rho_i}{1-\alpha\rho_{i+2}}\right) - \kappa_{i+1}\right] \leq 0. \tag{41}$$

Similar to (40) we have that

$$(q_{i+2}-q_{i+1}) - (q_{i+1}-q_i) = \beta q_i q_{i+1}\left[\kappa_{i+2}\left(\frac{1-\beta\rho_i}{1-\beta\rho_{i+2}}\right) - \kappa_{i+1}\right]. \tag{42}$$

Since $q_i > 0\ \forall i$, for the concavity of q_i it suffices to show

$$\left[\kappa_{i+2}\frac{1-\beta\rho_i}{1-\beta\rho_{i+2}} - \kappa_{i+1}\right] \leq 0. \tag{43}$$

Now

$$\begin{aligned} &\frac{1-\alpha\rho_i}{1-\alpha\rho_{i+2}} - \frac{1-\beta\rho_i}{1-\beta\rho_{i+2}} \\ &\qquad = \frac{(\alpha-\beta)(\rho_{i+2}-\rho_i)}{(1-\alpha\rho_{i+2})(1-\beta\rho_{i+2})} \\ &\qquad \geq 0. \end{aligned} \tag{44}$$

Then (41) and (44) combined with the fact that $\kappa_i \geq 0,\ \forall i$ proves (43). □

Proof of Theorem 10

We split the theorem into three claims.

Claim (1): $L^* := \cup_k L^{(k)}(x_k^*) \in \mathcal{L}_\gamma(n)$.

From (11), (12), and (14) we obtain

$$\begin{aligned} \mu_\gamma(n) + \frac{P_\gamma - 1}{\text{var}(V_\gamma)} &= \max_{L\in\Lambda_\gamma(n)} \sum_{k=1}^{P_\gamma} \mathcal{E}^{-1}(V_\gamma|L_{\gamma k}) \\ &\leq \max_{X\in\Delta_n(\mathcal{N}_{\gamma 1},\ldots,\mathcal{N}_{\gamma P_\gamma})} \sum_{k=1}^{P_\gamma} \mu_{\gamma,\gamma k}(x_k). \end{aligned} \tag{45}$$

Clearly $L^* \in \Lambda_\gamma(n)$. We then have from (11) and (12)

$$\begin{aligned} \mu_\gamma(n) + \frac{P_\gamma - 1}{\text{var}(V_\gamma)} &\geq \mathcal{E}^{-1}(V_\gamma|L^*) + \frac{P_\gamma - 1}{\text{var}(V_\gamma)} \\ &= \sum_{k=1}^{P_\gamma} \mathcal{E}^{-1}(V_\gamma|L^*_{\gamma k}) \\ &= \sum_{k=1}^{P_\gamma} \mu_{\gamma,\gamma k}(x_k^*) \\ &= \max_{X\in\Delta_n(\mathcal{N}_{\gamma 1},\ldots,\mathcal{N}_{\gamma P_\gamma})} \sum_{k=1}^{P_\gamma} \mu_{\gamma,\gamma k}(x_k). \end{aligned} \tag{46}$$

Thus from (45) and (46) we have

$$\mu_\gamma(n) = \mathcal{E}^{-1}(V_\gamma|L^*) = \max_{X\in\Delta_n(\mathcal{N}_{\gamma 1},\ldots,\mathcal{N}_{\gamma P_\gamma})} \sum_{k=1}^{P_\gamma} \mu_{\gamma,\gamma k}(x_k) - \frac{P_\gamma - 1}{\text{var}(V_\gamma)}, \tag{47}$$

which proves Claim (1).

Claim (2): If $L \in \mathcal{L}_{\gamma k}(n)$ then $L \in \mathcal{L}_{\gamma,\gamma k}(n)$ and vice versa.

Denote an arbitrary leaf node of the tree of γk as C. Then V_γ, $V_{\gamma k}$, and C are related through

$$V_{\gamma k} = \varrho_{\gamma k} V_\gamma + W_{\gamma k}, \tag{48}$$

and

$$C = \eta V_{\gamma k} + F \tag{49}$$

where η and $\varrho_{\gamma k}$ are scalars and $W_{\gamma k},\ F$ and V_γ are independent random variables. We note that by definition $\text{var}(V_\gamma) > 0\ \forall\gamma$ (see Definition 5). From Lemma 20 we have

$$\begin{aligned} \frac{\text{cov}(V_{\gamma k}, C)}{\text{cov}(V_\gamma, C)} &= \left(\frac{\text{var}(V_{\gamma k})}{\text{var}(V_\gamma)}\right)^{1/2} \left(\frac{\varrho_{\gamma k}^2 + \frac{\text{var}(W_{\gamma k})}{\text{var}(V_\gamma)}}{\varrho_{\gamma k}^2}\right)^{1/2} \\ &=: \xi_{\gamma,k} \\ &\geq \left(\frac{\text{var}(V_{\gamma k})}{\text{var}(V_\gamma)}\right)^{1/2}. \end{aligned} \tag{50}$$

From (50) we see that $\xi_{\gamma,k}$ is not a function of C.

Denote the covariance between V_γ and leaf node vector $L = [l_i] \in \Lambda_{\gamma k}(n)$ as $\Theta_{\gamma,L} = [\text{cov}(V_\gamma, l_i)]^T$. Then (50) gives

$$\Theta_{\gamma k,L} = \xi_{\gamma,k}\Theta_{\gamma,L}. \tag{51}$$

From (17) we have

$$\mathcal{E}(V_\gamma|L) = \text{var}(V_\gamma) - \varphi(\gamma, L) \tag{52}$$

where $\varphi(\gamma, L) = \Theta_{\gamma,L}^T S_L^{-1}\Theta_{\gamma,L}$. Note that $\varphi(\gamma, L) \geq 0$ since S_L^{-1} is positive semi-definite. Using (51) we similarly get

$$\mathcal{E}(V_{\gamma k}|L) = \text{var}(V_{\gamma k}) - \frac{\varphi(\gamma, L)}{\xi_{\gamma,k}^2}. \tag{53}$$

From (52) and (53) we see that $\mathcal{E}(V_\gamma|L)$ and $\mathcal{E}(V_{\gamma k}|L)$ are both minimized over $L \in \Lambda_{\gamma k}(n)$ by the same leaf vector that maximizes $\varphi(\gamma, L)$. This proves Claim (2).

Claim (3): $\mu_{\gamma,\gamma k}(n)$ is a positive, non-decreasing, and concave function of $n, \forall k, \gamma$.

We start at a node γ at one scale from the bottom of the tree and then move up the tree.

Initial Condition: Note that $V_{\gamma k}$ is a leaf node. From (2) and (52) we obtain

$$\mathcal{E}(V_\gamma|V_{\gamma k}) = \text{var}(V_\gamma) - \frac{(\varrho_{\gamma k}\text{var}(V_\gamma))^2}{\text{var}(V_{\gamma k})} \leq \text{var}(V_\gamma). \tag{54}$$

For our choice of γ, $\mu_{\gamma,\gamma k}(1)$ corresponds to $\mathcal{E}^{-1}(V_\gamma|V_{\gamma k})$ and $\mu_{\gamma,\gamma k}(0)$ corresponds to $1/\text{var}(V_\gamma)$. Thus from (54), $\mu_{\gamma,\gamma k}(n)$ is positive, non-decreasing, and concave (trivially since n takes only two values here).

Induction Step: Given that $\mu_{\gamma,\gamma k}(n)$ is a positive, non-decreasing, and concave function of n for $k = 1, \ldots, P_\gamma$, we prove the same when γ is replaced by $\gamma\uparrow$. Without loss of generality choose k such that $(\gamma\uparrow)k = \gamma$. From (12), (14), (52), (53) and Claim (2), we have for $L \in \mathcal{L}_\gamma(n)$

$$\mu_\gamma(n) \quad = \quad \frac{1}{\text{var}(V_\gamma)} \cdot \frac{1}{1 - \frac{\varphi(\gamma,L)}{\text{var}(V_\gamma)}} \tag{55}$$

and

$$\mu_{\gamma\uparrow,k}(n) \quad = \quad \frac{1}{\text{var}(V_{\gamma\uparrow})} \cdot \frac{1}{1 - \frac{\varphi(\gamma,L)}{\xi_{\gamma\uparrow,k}^2\text{var}(V_{\gamma\uparrow})}}. \tag{56}$$

From (47), the assumption that $\mu_{\gamma,\gamma k}(n)\ \forall k$ is a positive, non-decreasing, and concave function of n, and Lemma 8 we have that $\mu_\gamma(n)$ is a non-decreasing and concave function of n. Note that by definition (see (12)) $\mu_\gamma(n)$ is positive. This combined with (2), (55), (56), (50) and Lemma 21, then prove that $\mu_{\gamma\uparrow,k}(n)$ is also positive, non-decreasing, and concave. $\square$

APPENDIX B

We now prove a lemma which we then use to prove Theorem 18.

As a first step we compute the leaf arrangements L which maximize and minimize the sum of all elements of $S_L = [s_{i,j}(L)]$. We restrict our analysis to a covariance tree with depth N and in which each node (excluding leaf nodes) has σ child nodes. We introduce some notation. Define

$$\Gamma^{(u)}(p) := \{L : L \in \Lambda_{\emptyset}(\sigma^p) \text{ and } L \text{ is a uniform leaf node set}\} \tag{57}$$

and

$$\Gamma^{(c)}(p) := \{L : L \text{ is a clustered leaf set of a node at scale } N-p\} \tag{58}$$

for $p = 0, 1, \dots, N$. We number nodes at scale m in an arbitrary order from $q = 0, 1, \dots, \sigma^m - 1$ and refer to a node by the pair (m, q).

Lemma 22: *Assume a positive correlation progression. Then,* $\sum_{i,j} s_{i,j}(L)$ *is minimized over* $L \in \Lambda_{\emptyset}(\sigma^p)$ *by every* $L \in \Gamma^{(u)}(p)$ *and maximized by every* $L \in \Gamma^{(c)}(p)$*. For a negative correlation progression,* $\sum_{i,j} s_{i,j}(L)$ *is maximized by every* $L \in \Gamma^{(u)}(p)$ *and minimized by every* $L \in \Gamma^{(c)}(p)$*.*

Proof: Set p to be an arbitrary element in $\{1, \dots, N-1\}$. The case of $p = 0$ and $p = N$ is trivial. Let $\vartheta_m = \#\{s_{i,j}(L) \in S_L : s_{i,j}(L) = c_m\}$ be the number of elements of S_L equal to c_m. Define $a_m := \sum_{k=0}^{m} \vartheta_k, m \geq 0$ and set $a_{-1} = 0$. Then

$$\begin{aligned}
\sum_{i,j} s_{i,j} &= \sum_{m=0}^{N} c_m \vartheta_m = \sum_{m=0}^{N-1} c_m (a_m - a_{m-1}) + c_N \vartheta_N \\
&= \sum_{m=0}^{N-1} c_m a_m - \sum_{m=-1}^{N-2} c_{m+1} a_m + c_N \vartheta_N \\
&= \sum_{m=0}^{N-2} (c_m - c_{m+1}) a_m + c_{N-1} a_{N-1} - c_0 a_{-1} \\
&\qquad\qquad + c_N \vartheta_N \\
&= \sum_{m=0}^{N-2} (c_m - c_{m+1}) a_m + \text{constant},
\end{aligned} \tag{59}$$

where we used the fact that $a_{N-1} = a_N - \vartheta_N$ is a constant independent of the choice of L, since $\vartheta_N = \sigma^p$ and $a_N = \sigma^{2p}$.

We now show that $L \in \Gamma^{(u)}(p)$ maximizes $a_m, \forall m$ while $L \in \Gamma^{(c)}(p)$ minimizes $a_m, \forall m$.

First we prove the results for $L \in \Gamma^{(u)}(p)$. Note that L has one element in the tree of every node at scale p.

Case (i) $m \geq p$. Since every element of L has distance at most $p-1$ with all other elements, $a_m = \sigma^p$ which is the maximum value it can take. Case (ii) $m < p$ (assuming $p > 0$). Let the number of elements of L belonging to the sub-tree of $V_{(m+1,q)}$ be $g_q, q = 0, \dots, \sigma^{m+1} - 1$. We have

$$a_m = \sum_{q=0}^{\sigma^{m+1}-1} g_q(\sigma^p - g_q) = \frac{\sigma^{2p+1+m}}{4} - \sum_{q=0}^{\sigma^{m+1}-1} (g_q - \sigma^p/2)^2 \tag{60}$$

since every element of L in the tree of $V_{(m+1,q)}$ must have distance *at most* m with all nodes *not* in the same tree but must have distance *at least* $m+1$ with all nodes *within* the same tree.

The choice of g_q's is constrained to lie on the hyperplane $\sum_q g_q = \sigma^p$. Obviously the quadratic form of (60) is maximized by the point on this hyperplane closest to the point $(\sigma^p/2, \dots, \sigma^p/2)$ which is $(\sigma^{p-m-1}, \dots, \sigma^{p-m-1})$. This is clearly achieved by $L \in \Gamma^{(u)}(p)$.

Now we prove the results for $L \in \Gamma^{(c)}(p)$.

Case (i) $m < N - p$. We have $a_m = 0$, the smallest value it can take.

Case (ii) $N - p \leq m < N$. Consider leaf node $l_i \in L$ which without any loss of generality belongs to the tree of $V_{(m+1,0)}$. Let $a_m(l_i)$ be the number of elements of L to which l_i has distance less than or equal to m. Now since l_i has distance less than or equal to m only with those elements of L not in the same tree, we must have $a_m(l_i) \geq \sigma^p - \sigma^{N-m-1}$. Since $L \in \Gamma^{(c)}(p)$ achieves this lower bound for $a_m(l_i), \forall i$ and $a_m = \sum_i a_m(l_i)$, $L \in \Gamma^{(c)}$ minimizes a_m in turn. □

Let us now study, to what extent the above results transfer to the actual matrix of interest S_L^{-1}. We start with a useful formula.

Lemma 23: *Denote the eigenvalues of S_L by $\lambda_j, j = 1, \dots, \sigma^p$. Assume that no leaf node of the tree can be expressed as a linear combination of other leaf nodes, implying that $\lambda_j > 0, \forall j$. Set $D_L = [d_{i,j}]_{\sigma^p \times \sigma^p} := S_L^{-1}$. Then there exist positive numbers f_i with $f_1 + \ldots + f_p = 1$ such that*

$$\sum_{i,j=1}^{\sigma^p} s_{i,j} = \sigma^p \sum_{j=1}^{\sigma^p} f_j \lambda_j, \text{ and} \tag{61}$$

$$\sum_{i,j=1}^{\sigma^p} d_{i,j} = \sigma^p \sum_{j=1}^{\sigma^p} f_j / \lambda_j. \tag{62}$$

Furthermore, for both special cases, $L \in \Gamma^{(u)}(p)$ and $L \in \Gamma^{(c)}(p)$, we may choose the weights f_j such that only one is non-zero.

Proof: Since the matrix S_L is real and symmetric there exists an orthonormal eigenvector matrix $U = [u_{i,j}]$ that diagonalizes S_L, that is $S_L = U \Xi U^T$ where Ξ is diagonal with eigenvalues $\lambda_j, j = 1, \dots, \sigma^p$.

Define $w_j := \sum_i u_{i,j}$. Then

$$\begin{aligned}\sum_{i,j} s_{i,j} &= \mathbf{1}_{1\times\sigma^p} S_L \mathbf{1}_{\sigma^p\times 1} \\ &= (\mathbf{1}_{1\times\sigma^p} U)\Xi(\mathbf{1}_{1\times\sigma^p} U)^T \\ &= [w_1 \dots w_{\sigma^p}]\Xi[w_1 \dots w_{\sigma^p}]^T \\ &= \sum_j \lambda_j w_j^2.\end{aligned}$$

Further, since $U^T = U^{-1}$ we have

$$\sum_j w_j^2 = (\mathbf{1}_{1\times\sigma^p} U)(U^T \mathbf{1}_{\sigma^p\times 1}) = \mathbf{1}_{1\times\sigma^p} I \mathbf{1}_{\sigma^p\times 1} = \sigma^p. \tag{63}$$

Setting $f_i = w_i^2/\sigma^p$ establishes (61). Using the decomposition $S_L^{-1} = (U^T)^{-1}\Xi^{-1}U^{-1} = U\Xi^{-1}U^T$ similarly gives (62).

Consider the case $L \in \Gamma^{(u)}(p)$.

Since $L = [l_i]$ consists of a symmetrical set of leaf nodes (the set of distances between any element l_i and the rest does not depend on i) the sum of the covariances of a leaf node l_i with its fellow leaf nodes does not depend on i, and we can set:

$$\lambda^{(u)} := \sum_{j=1}^{\sigma^p} s_{i,j}(L) = c_N + \sum_{m=1}^{p} \sigma^{p-m} c_m. \tag{64}$$

With the sum of the elements of any row of S_L being identical, the vector $\mathbf{1}_{\sigma^p\times 1}$ is an eigenvector of S_L with eigenvalue $\lambda^{(u)}$ equal to (64).

Recall that we can always choose a basis of orthogonal eigenvectors which includes $\mathbf{1}_{\sigma^p\times 1}$ as the first basis vector. It is well known that the rows of the corresponding basis transformation matrix U will then be exactly these normalized eigenvectors. Since they are orthogonal to $\mathbf{1}_{\sigma^p\times 1}$, the sum of their coordinates w_j ($j = 2, \dots, \sigma^p$) must be zero. Thus, all f_i but f_1 vanish. (The last claim follows also from the observation that the sum of coordinates of the normalized $\mathbf{1}_{\sigma^p\times 1}$ equals $w_1 = \sigma^p\sigma^{-p/2} = \sigma^{p/2}$; due to (63) $w_j = 0$ for all other j.)

Consider the case $L \in \Gamma^{(u)}(p)$.

The reasoning is similar to the above and we can define

$$\lambda^{(c)} := \sum_{j=1}^{\sigma^p} s_{i,j}(L) = c_N + \sum_{m=1}^{p} \sigma^{m} c_{N-m}. \tag{65}$$

□

Proof of Theorem 18

Due to the special form of the covariance vector $\operatorname{cov}(L, V_{\emptyset})= \rho\mathbf{1}_{1\times\sigma^k}$ we observe from (17) that minimizing the LMMSE $\mathcal{E}(V_{\emptyset}|L)$ over $L \in \Lambda_{\emptyset}(n)$ is equivalent to maximizing $\sum_{i,j} d_{i,j}(L)$ the sum of the elements of S_L^{-1}.

Note that the weights f_i as well as the eigenvalues λ_i of Lemma 23 depend on the arrangement of the leaf nodes L. To avoid confusion, we denote by λ_i the eigenvalues of S_L for an arbitrary fixed set of leaf nodes L, and by $\lambda^{(u)}$ and $\lambda^{(c)}$ the only relevant eigenvalues of $L \in \Gamma^{(u)}(p)$ and $L \in \Gamma^{(c)}(p)$ according to (64) and (65).

Assume a positive correlation progression, and let L be an arbitrary set of σ^p leaf nodes. Lemma 22 and Lemma 23 then imply that

$$\lambda^{(u)} \;\leq\; \sum_j \lambda_j f_j \;\leq\; \lambda^{(c)}. \tag{66}$$

Since S_L is positive definite, we must have $\lambda_j > 0$. We may then interpret the middle expression as an expectation of the positive "random variable" λ with discrete law given by f_i. Jensen's inequality applies with the convex function $1/x$ $(x > 0)$ and yields

$$\sum_j (1/\lambda_j) f_j \;\geq\; \frac{1}{\sum_j \lambda_j f_j} \;\geq\; \frac{1}{\lambda^{(c)}}. \tag{67}$$

In other words, $\sum_{i,j} d_{i,j}$ is minimized by $L \in \Gamma^{(c)}(p)$; that is, clustering the nodes in the tree gives the worst LMMSE.

A similar argument holds for the negative correlation progression case which proves the Theorem. □

APPENDIX C: PSUEDO-CODE AND COMPUTATIONAL COMPLEXITY OF WATER-FILLING ALGORITHM FOR INDEPENDENT INNOVATIONS TREES

structure node{

  var;   /* variance of node */

  M;   /* $\mathcal{N}_\gamma$, where γ is this node */

  N;   /* $|L_\gamma|$ where L is the current leaf node set */

  curr_u;   /* $\mu_\gamma(N)$ */

  next_u;   /* $\mu_\gamma(N+1)$ */

  curr_v;   /* $\mu_{\gamma\uparrow,k}(N)$, where $\gamma k = \gamma$ */

  next_v;   /* $\mu_{\gamma\uparrow,k}(N+1)$ */

  num_child;   /* P_γ */

  child;   /* pointer to location of first child, $\gamma 1$ */

  cov_fac;   /* $\xi^2_{\gamma\uparrow,k}$ where $\gamma\uparrow k = \gamma$ */

  parent;   /* pointer to parent, $\gamma\uparrow$ */

  const;   /* ϱ_γ */

  var_W;   /* var(W_γ) */

  max_v_loc;/* child number with highest (next_v-curr_v) */

  }

*node create_nodes(){

/*create an array of "node" structures which sets the parent, child, num_child, var_W, and const fields for all nodes. Initialize the root node variance. Return a pointer to the root location.*/

}

init(node *node){

  node *parent, *child;   /*pointers to parent and child of node*/

  parent=node→parent;   /*set parent to parent of current node

  child=node→child;   /*set child to first child of current node*/

   node→N=0;

   if (node ≠ root){   /* if the node is not the root */

    node→var=((node→const)^2 * parent→var) +node→var_W ;

    node→cov_fac=(node→var/parent→var)*[[(node→const)^2+(node→var_W/parent→var)]

```
                /(node→const)^2] ;        /* set node variance */
            node→curr_v=1/parent→var;
            }/* end if (node≠ root) */

        node→curr_u=1/node→var;

        if (node→child ≠ 0){        /* if not a leaf node */
            for k=1 to node→num_child {
                init(child+k-1);        /* initialize all children */
                node→M+=(child+k-1)→M;
                }        /*end for loop*/

            max_v_loc=find_max_v(node);        /*find child with maximum (next_v-curr_v) */
            node→max_v_loc=max_v_loc;
            node→next_u=node→curr_u+(child+max_v_loc-1)→next_v-(child+max_v_loc-1)→curr_v;
            if (node≠root){
                node→next_v=1/(parent→var - ((1/node→cov_fac)*(node→var-(1/node→next_u))));
            } ;        /*end if (node ≠ root) */
    } ;        /* end if (node→child ≠ 0)*/
    else {        /*else if a leaf node*/
            node→M=1;
            node→next_v=1/(parent→var- ((node→const*parent→var)^2/node→var));
            } ;
    return ;
}
  int find_max_v(node *node){        /* find child with maximum (next_v - curr_v) */
    int i;
    double max=0.0;
    int max_v_loc;        /*child number*/
    node *child;        /*pointer to child*/


        child=node→child;

        for i=1 to node→num_child {
```

```
        if (child→next_v - child→curr_v→ max){
            max=child→next_v - child→curr_v;
            max_v_loc=i;
            child++;        /*move to next child node*/
        }       /*end if*/
    }       /*end for loop*/
}

/*returns pointer to the next optimal leaf node */
find_next_leaf(node *node){
    node *parent, *child;       /*pointers to parent and child*/
    parent=node→parent;
    child=node→child;
    node→N++;

    if (node→num_child=0){       /*if a leaf node */
        best_child=node;       /* return this leaf node */
    }
    else {
        best_child=find_next_leaf(child+max_v_loc-1);
        node→curr_u=node→next_u;
        node→curr_v=node→next_v;
        if (node→N < node→M) {       /* if nodes remaining in sub-tree*/
            max_v_loc=find_max_v(node);       /*find child with maximum (next_v-curr_v) */
            node→max_v_loc=max_v_loc;
            node→next_u=node→curr_u+(child+max_v_loc-1)→next_v-(child+max_v_loc-1)→curr_v;
            if (node≠root){
                node→next_v=1/(parent→var - ((1/node→cov_fac)*(node→var-(1/node→next_u))));
            } ;       /* end if (node ≠ root) */
        };       /*end if (node→N < node→M) */
    } /*end if (node→num_child=0) */       return(best_child);
}

  leaves best_leaves(int n){ /* Determine the best n leaf nodes to estimate the root*/
    node *root;
```

```
    root=create_nodes();      /* create array of node structures */
    init(root);      /* initialize the structure array starting with root */
    leaves=empty set;      /* leaves contains the set of leaf indices selected so far*/
    for i=1 to n
        leaves=leaves∪find_next_leaf(root);      /* each leaf entry consists of a set of child
                                                    indices of nodes connecting the root to the leaf*/
    return(leaves);
}
```

Computational Complexity: Denote the number of nodes in the tree by R and the depth of the tree by N. The complexity of `create_nodes()` and `init(root)` is $O(R)$ since they create and initialize R structures of type `node`. The complexity of `find_next_leaf(root)` is $O(N)$ since it updates a fixed amount of information at each node along a path from the root to a leaf along of length N. The overall complexity of the water-filling algorithm is thus $O(R + nN)$. If every node in the tree has the same number of children then N is approximately $\log(R)$ thus giving an overall complexity of $O(R + n\log(R))$.

REFERENCES

[Bellhouse, 1977] Bellhouse, D. R. (1977). Some optimal designs for sampling in two dimensions. *Biometrika* 64(3), pp. 605–611.

[Breslau et al., 2000] Breslau, L., Knightly, E., Shenker, S. & Stoica, I. (2000). Endpoint admission control: Architectural issues and performance. *ACM SIGCOMM* , pp. 57–69.

[Brownlee & kc claffy, 2004] Brownlee, N. & kc claffy (2004). Internet measurement. *IEEE Internet Computing* 8(5), pp. 30–33.

[Carter & Crovella, 1996] Carter, R. L. & Crovella, M. E. (1996). Measuring bottleneck link speed in packet-switched networks. *Performance Evaluation (Proceedings of Performance'96)* 27&28, pp. 297–318.

[Chou et al., 1994] Chou, K. C., Willsky, A. S. & Benveniste, A. (1994). Multiscale recursive estimation, data fusion, and regularization. *IEEE Trans. on Automatic Control* 39(3), pp. 464–478.

[Chou & Miao, 2001] Chou, P. & Miao, Z. (2001). Rate-distortion optimized streaming of packetized media. *Microsoft Research Technical Report MSR-TR-2001-35*.

[Cover & Thomas, 1991] Cover, T. M. & Thomas, J. A. (1991). *Information Theory*. Wiley Interscience.

[Davis, 1994] Davis, G. (1994). Adaptive nonlinear approximations. *Ph.D. thesis, New York University*.

[Hájek, 1959] Hájek, J. (1959). Optimum strategy and other problems in probability sampling. *Căsopis Pěst. Mat.* 84, pp. 387–423. Also available in *Collected Works of Jaroslav Hájek - With Commentary* by M. Hušková, R. Beran and V. Dupač, Wiley, 1998.

[He & Hou, 2003] He, G. & Hou, J. C. (2003). On exploiting long-range dependency of network traffic in measuring cross-traffic on an end-to-end basis. *IEEE INFOCOM*.

[Lawry & Bellhouse, 1992] Lawry, K. A. & Bellhouse, D. R. (1992). Relative efficiency of certian randomization procedures in an $n \times n$ array when spatial correlation is present. *Jour. Statist. Plann. Inference* 32, pp. 385–399.

[Ma & Ji, 1998] Ma, S. & Ji, C. (1998). Modeling video traffic in the wavelet domain. *IEEE INFOCOM* , pp. 201–208.

[Pedigo & Buntin, 1993] Pedigo, L. P. & Buntin, G. D., eds (1993). *Handbook of sampling methods for arthropods in agriculture*. CRC Press, Boca Raton, FL.

[Riedi et al., 1999] Riedi, R., Crouse, M. S., Ribeiro, V. & Baraniuk, R. G. (1999). A multifractal wavelet model with application to TCP network traffic. *IEEE Trans. on Information Theory* 45(3), pp. 992–1018.

[Salehi, 2004] Salehi, M. M. (2004). Optimal sampling design under a spatial correlation model. *Jour. of Statistical Planning and Inference* 118, pp. 9–18.

[Stark & Woods, 1986] Stark, H. & Woods, J. W. (1986). *Probability, random processes, and estimation theory for engineers*. Prentice-Hall.

[Tuan & Park, 1999] Tuan, T. & Park, K. (1999). Multiple time scale congestion control for self-similar network traffic. *Performance Evaluation* 36, pp. 359–386.

[Vidàcs & Virtamo, 1999] Vidàcs, A. & Virtamo, J. T. (1999). ML estimation of the parameters of fBm traffic with geometrical sampling. *COST257* 99(14).

[Willett et al., 2004] Willett, R., Martin, A. & Nowak, R. (2004). Backcasting: adaptive sampling for sensor networks. *Information Processing in Sensor Networks (IPSN)*.

[Willsky, 2002] Willsky, A. (2002). Multiresolution markov models for signal and image processing. *Proceedings of the IEEE* 90(8), pp. 1396–1458.

[Xu, 1999] Xu, H. (1999). Universally optimal designs for computer experiments. *Statistica Sinica* 9, pp. 1083–1088.