Fast Algorithms for Optimal Link Selection in Large-scale Network Monitoring

Michael G. Kallitsis, Stilian Stoev and George Michailidis
Department of Statistics, University of Michigan, Ann Arbor
{mgkallit, sstoev, gmichail}@umich.edu

Abstract

The robustness and integrity of IP networks require efficient tools for traffic monitoring and analysis, which scale well with traffic volume and network size. We address the problem of *optimal large-scale* monitoring of computer networks under resource constraints. Specifically, we consider the task of *selecting* the "best" subset of at most K links to monitor, so as to *optimally predict* the traffic load at the remaining ones. Our notion of optimality is quantified in terms of the statistical error of network traffic predictors. The *optimal monitoring* problem at hand is akin to certain combinatorial constraints, which render the algorithms seeking the exact solution impractical. We develop a number of *fast algorithms* that improve upon existing algorithms in terms of computational complexity and accuracy. Our algorithms exploit the geometry of *principal component analysis*, which also leads us to new types of theoretical bounds on the prediction error. Finally, these algorithms are amenable to randomization, where the best of several parallel independent instances often yields the exact optimal solution. Their performance is illustrated and evaluated on simulated and real-network traces.

I. Introduction

A. Motivation

ADVANCES in high-throughput technologies have led to unprecedented growth of network traffic due to a large host of applications such as Web, IP telephony, cloud computing, social networking, audio/video streaming, etc. Network monitoring aims to sample traffic data on short time-scales and quickly analyze them. A natural application of online monitoring is *anomaly detection* [1], [2]. This requires periodic traffic flow measurements on a *large set* of links, a costly and computationally challenging task. Therefore, since global monitoring of large-scale networks involves large traffic volumes,

This research was partially supported by NSF grants DMS-0806094 and DMS-1106695 at the University of Michigan.

it becomes impractical and often impossible due to resource constraints. For example, limited bandwidth prevents all data from being sent to a centralized coordinator (*e.g.,* Cisco's Netflow Collector [3]). Further, typically only part of the network may be accessible for monitoring at a given time due to engineering or management constraints. Therefore, it is important to be able to statistically predict the traffic loads on all links in the network based on information obtained from a limited set of observation sites.

Recently, [4] introduced methodology for global network traffic modeling, which utilizes the routing and other network-specific structural information derived from sampled NetFlow measurements. Once calibrated, these models can be fit by monitoring small subsets of links in the network and used to solve the *network kriging* problem [5]. This modeling approach allows for *fast online* prediction of the traffic volume, by utilizing measurements on a small set of links. Certainly, the accuracy of the prediction heavily depends on the selection of links to be monitored. One fundamental problem in this context is how to best allocate the monitors so as to minimize the *prediction uncertainty*.

In this work, we focus on the *optimal monitoring* problem for wired IP networks. Given a global traffic model in terms of the *covariance matrix* of all links in the network, the goal is to find the optimal set of K links to monitor so as to optimally predict link utilization on the remaining ones (see Figures 1 and 8). The underlying network covariance matrix may be obtained through a network-specific traffic model as in [4] or by other methods, *e.g.* from extensive off-line analysis of historical traffic data (see Section III). Here, we shall assume that this covariance is a known and given "input" for our algorithms[1].

To informally introduce the problem under study, consider the toy example in Fig. 1, based on the Internet2 topology [6]. Traffic flows through the highlighted links. Suppose we have the following 5 network flows with the same statistical characteristics, denoted as source–destination (S, D) pairs: (SALT, NEWY), (SALT, ATLA), (HOUS, NEWY), (HOUS, ATLA) and (KANS, ATLA). Assuming that prediction uncertainty is quantified by the mean prediction error of the unobserved links (i.e. A-optimality, see Section III-B), we pose the following question: Suppose one is allowed to monitor only $K = 2$ links, which two links should be chosen? One can easily guess that the first member of that set should be link "c". This is because link "c" serves the most flows; indeed, link "c" belongs to the optimal set (in general, more than one might exist). The other link can be either "d" or "e". Finally, the less obvious choice $\{$"d", "e"$\}$ is also an optimal solution.

The situation becomes more involved if we assign link preferences. For example, a network operator could face different scenarios and may assign distinct link weights/priorities (see section VII-E), such as monitoring links which can be more error prone, or links that are more susceptible to network anomalies,

[1]We exclude pathological or trivial cases. *E.g.*, in a network where all origin destination flows are served by individual dedicated links, the dependence between links becomes trivial and the network kriging problem is of no practical interest..

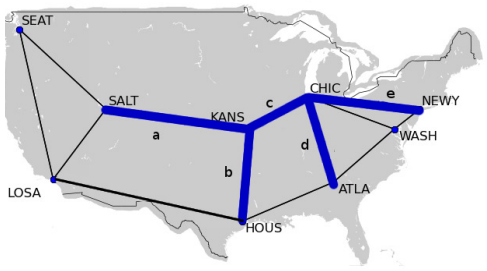


Fig. 1. Toy example on a subnetwork of the Internet2 topology. For the full-topology, real-world application see Section VII.

or even not wanting to observe inexpensive links, etc. Considering again our toy topology, link "a" could join the observed set depending on its priority. To complicate matters, note that these decisions should be made online and fast. In real-world networks of hundreds of links servicing thousands of flows, where network conditions change frequently, exact algorithms – which amount to complete combinatorial searches – are prohibitively expensive. Hence, only fast heuristic algorithms are practical; finding such algorithms consists the main topic of this paper.

B. Contributions

Our focus is the problem of efficiently selecting the subset of network links to monitor which yields best overall prediction for the remaining ones. Specifically, we aim to minimize a functional of the prediction error, which represents the uncertainty on the information that is obtained. There are several popular criteria for measuring the uncertainty, such as entropy (also known as D-optimality, see [7]), mutual information [8], trace (A-optimality) and spectral norm (E-optimality) of the prediction error covariance matrix [9]. In this work, we focus on A- and E-optimality.

The combinatorial optimization problem at hand – formally introduced in Section III – belongs to the family of *subset selection* problems, which are NP-hard problems in general. For example, min- and max-entropy (D-optimality) sampling are shown to be NP-hard problems in [7] by reduction from the well-known NP-hard problems of CLIQUE and STABLE SET [10]. Related problems such as *sparse approximation* [11], *subset selection for regression* [12] and *sensor placement* using the criterion of mutual information [8] are also proven to be NP-hard. Therefore, even though the NP-hardness of our problem remains an open question (as is the *column subset selection problem* [13], [14], a closely related one), there is strong evidence that it is NP-hard. In principle, exact solutions can be obtained by using an *integer program* formulation. However, the problem becomes prohibitively expensive for even moderate-size settings. For example, an exhaustive search implementation in CPLEX [15] running on a medium-sized computer cluster required several hours to complete the solution for a 26 link problem with a "budget" of $K = 12$. Other methods, such as the mixed-integer program with constraint generation proposed in [16], are also computationally expensive and impractical. Branch and bound methods could

be applied (see [7]), but the huge number of "branching" choices makes the procedure unattractive for online implementations in large-scale networks. These challenges emphasize the need for *fast* heuristic algorithms for solving the optimal prediction problem. The main contributions of our study are:

1) We introduce novel approximation algorithms whose efficiency can allow network operators to solve the optimal monitoring problem *fast* and practically *online*. We juxtapose these algorithms against naïve greedy implementations and against a selection algorithm proposed in [5]. The evaluation results suggest qualitative and computational advantages of our proposed methods against the aforementioned algorithms (*e.g.,* see Figures 6 and 7). Further, we show that *kriging*-based traffic prediction outperforms estimation methods based on diffusion wavelets [17] (see Figure 8).
2) By exploiting the geometry of the objective functions used, together with connections to *principal component analysis* (PCA) we obtain *prediction error bounds* that do not stipulate submodularity of the objective functions as in existing work (*e.g.* see [8]). Furthermore, these bounds help us assess the quality of our approximate solution with respect to the optimal one, which is often unknown and practically unobtainable for large networks.
3) Randomized versions of the proposed algorithms can be implemented in a parallel fashion. This leads to further improvements in practice, often yielding the optimal solution (see Fig. 2(a)).

The paper is organized as follows: in Section II we outline the related work. In Section III we formally define the network monitoring problem. In Section IV we present the connections with PCA and derive our PCA-based lower bound on the performance of our algorithms, which in fact applies to any other algorithm. Section V introduces and analyzes our three approximation algorithms, and concludes with possible extensions to randomized implementations. Section VI discusses theoretical performance guarantees for the error reduction achieved at each step. Next, in Section VII, we evaluate our methods in a variety of network scenarios, including the real-world datasets obtained from the Internet2 network [6] and the Cooperative Association for Internet Data Analysis (CAIDA) [18]. We also provide insights on how a network operator can choose the budget K, and how *weighted* link monitoring can be implemented.

II. Literature Survey

Statistical prediction for the sake of network monitoring, named as network kriging, appeared in [5]. The authors developed a framework in which global flow characteristics can be predicted from a small sample of flow measurements. Their problem seeks the "best" network paths to monitor and they propose a fast approximation algorithm based on SVD and QR factorizations [19]. Efficient network monitoring of end-to-end performance metrics also appears in [17]. The presented procedure utilizes diffusion wavelets to capture spatial and temporal correlations in traffic measurements. To perform this analysis, a path

selection problem similar to the one in [5] needs to be addressed, and the QR-based method is again employed. The authors in [20], [21] follow a combined approach in which they first seek the links at which network monitors should be activated, and then aim to choose an optimal packet sampling scheme so that the traffic estimation error is minimized. They propose fast algorithms that take advantage of the fact that the problem at hand can be reduced to a second-order cone programming problem [20], [21]. Optimal sampling schemes for link traffic is also discussed in [22], [9], where a state-space model is engaged that captures the dynamics of link traffic.

In [8], near-optimal sensor placement for temperature monitoring is examined, which has a similar formulation to our problem. The authors propose heuristics that outperform the classical implementation of the greedy algorithm that appears in [23]. However, these methods apply only in special cases such as when the covariance matrix of the joint Gaussian distribution of the temperatures has "low-bandwidth"[2]. The structure of the covariance matrix is also exploited in [12] where the problem of *subset selection for regression* is studied. The efficiency of the algorithms algorithms in [12] relies, however, on the special cases of "low-bandwidth" and "tree covariance" graphs. We note, though, that such special cases do not commonly arise in real-world network monitoring applications, and thus the efficiency of the algorithms in [8], [12] is practically unsatisfactory for large-scale communication infrastructures. This is because the topology and routing of real-life networks often lead to covariance matrices with complex structure. A network session may flow through a large number of links that can be geographically distributed (see Fig. 1 for a large-scale network topology). Hence, the inherent statistical independence between distant locations that may be apparent in sensor networks, giving rise to "low-bandwidth" covariance matrices, is generally absent in IP wired networks.

Although there has been a vast amount of work on problems of similar flavor (see also [24], [25], [26]), optimal monitoring in large-scale networks poses new challenges. As mentioned above, there is no natural sparsity of the covariance structure between the links in the network. At the same time, approaching the problem from a combinatorial perspective leads to computational challenges similar to existing NP-hard problems. This motivated us to explore a new approach based on the geometry of PCA.

III. Problem Formulation

A. Large-scale Network Monitoring

Consider a communication network of N nodes and L links. The total number of traffic flows, i.e. source and destination (S, D) pairs, is denoted by J. Traffic is routed over the network along predefined

[2]A covariance matrix Σ has bandwidth β when the variables can be ordered in such a way that $\Sigma_{ij} = 0$ when $|j - i| > \beta$.

paths described by a routing matrix $R = (r_{\ell,j})_{L\times J}$, with $r_{\ell,j} = 1$, when route j uses link ℓ and 0 otherwise. Let $\boldsymbol{x}(t) = (x_j(t))_{j=1}^{J}$ and $\boldsymbol{y}(t) = (y_\ell(t))_{\ell=1}^{L}$, $t = 1, 2, \cdots$ be the vector time series[3] of traffic traversing all J routes and L links, respectively. We shall ignore network delays and adopt the assumption of *instantaneous propagation*. This is reasonable when traffic is monitored at a time-scale coarser than the round-trip time of the network, which is the case in our setting. We thus obtain that the link and route level traffic are related through the fundamental *routing equation*[4]

$$\boldsymbol{y}(t) = R\boldsymbol{x}(t). \tag{1}$$

In [27], we proposed a global mechanistic model for the traffic on an entire network. The traffic flow along each route was represented as a composition of multiple long-range dependent On/Off traces describing the behavior of individual users. Such On/Off processes have been popular and successful models for the pattern of traffic generated by various protocols, services and applications (*e.g.,* peer-to-peer, file transfers, VoIP, etc.). It is well-known that the composition of multiple independent traces of this type yields *long-range dependent* models, that are well-approximated by *fractional Gaussian noise* (see *e.g.* [28]). On the other hand, using NetFlow data, we found that the traffic flows $x_j(t)$ across different routes j are relatively weakly correlated[5] (in j). Thus, *routing* (see Eq. (1)) becomes the primary cause of statistical dependence between the traffic traces $y_\ell(t)$ across different links ℓ in the network. In particular, the greater the number of common flows that pass through two given links, the greater the correlation between the traffic loads on these links. The dependence of $y_\ell(t)$ across "space" (links) ℓ and time t can be quantified and succinctly described in terms of the *functional fractional Brownian motion* (see [27]).

In [4], we developed further statistical methodology that allows one to estimate (using NetFlow data) the structure of the means $\boldsymbol{\mu_x} = \mathbb{E}\boldsymbol{x}(t)$, the flow-covariances $\Sigma_{\boldsymbol{x}} := \mathbb{E}(\boldsymbol{x}(t) - \boldsymbol{\mu_x})(\boldsymbol{x}(t) - \boldsymbol{\mu_x})^T$ and their relationship for all routes in the network. This leads to a practical factor model for the link loads $\boldsymbol{y}(t)$, which can be estimated *online* from traffic measurements of just a few links. The estimated model can in turn be used to perform *network prediction*, discussed next.

[3]Here, time is discrete and traffic loads are measured in bytes or packets per unit time, over a time scale greater than the round-trip time of the network.

[4] Note that in backbone IP networks the routing matrix R does not change often. Further, in wired networks, routing matrices can be readily obtained by periodically using Cisco's Discovery Protocol (CDP) or by checking the routing tables. See the Supplemental material for more information.

[5]Except in periods of congestion where the TCP feedback mechanism induces dependence between the *forward* and *reverse* flows (see [27] and also [22]).

B. Link Selection for Optimal Prediction

We focus on instantaneous prediction, known as *kriging*. In this case, one can measure the *aggregated* traffic (i.e., traffic of all network flows traversing the given link) on a set of K *observed* links $\boldsymbol{y}_o(t) = (y_\ell(t))_{\ell\in\mathcal{O}}$, $\mathcal{O} \subseteq \mathcal{L} := \{1, \cdots, L\}$ (*e.g.* see Fig. 8). The goal is then to predict the traffic carried on the unobserved links $\boldsymbol{y}_u(t) = (y_\ell(t))_{\ell\in\mathcal{U}}$, $\mathcal{U} := \{1, \cdots, L\} \setminus \mathcal{O}$, at the same time t. (For an example of temporal prediction, see [27]; henceforth, we often omit the argument t.) For link loads being jointly Gaussian distributed[6], the *ordinary kriging* estimate given by

$$\widehat{\boldsymbol{y}}_u = \boldsymbol{\mu}_u + \Sigma_{uo}\Sigma_{oo}^{-1}(\boldsymbol{y}_o - \boldsymbol{\mu}_o), \tag{2}$$

is the best linear unbiased predictor (BLUP) for $\boldsymbol{y}_u$ via $\boldsymbol{y}_o$, where $\boldsymbol{\mu}_{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{\mu}_u \\ \boldsymbol{\mu}_o \end{pmatrix}$ and $\Sigma_{\boldsymbol{y}} = \begin{pmatrix} \Sigma_{uu} & \Sigma_{uo} \\ \Sigma_{ou} & \Sigma_{oo} \end{pmatrix}$ are the partitions of the mean and the *covariance* of $\boldsymbol{y} = \boldsymbol{y}(t)$ into blocks corresponding to the unobserved (u) and observed (o) links. The estimation of $\boldsymbol{\mu}_{\boldsymbol{y}} = R\boldsymbol{\mu}_{\boldsymbol{x}}$ and $\Sigma_{\boldsymbol{y}} = R\Sigma_{\boldsymbol{x}}R^T$ are important open problems, which were partially addressed in [27] and [4] (recall the discussion at the end of subsection III-A). For the purpose of this work, we shall assume that $\boldsymbol{\mu}_{\boldsymbol{y}}$ and $\Sigma_{\boldsymbol{y}}$ are *known*. We refer the reader to [4], [27] and the Supplemental material of this paper for details on how they can be obtained. Assuming multivariate normality, we also have[7]

$$\boldsymbol{y}_u|\boldsymbol{y}_o \sim \mathcal{N}(\widehat{\boldsymbol{y}}_u, \Sigma_{uu} - \Sigma_{uo}\Sigma_{oo}^{-1}\Sigma_{ou}), \tag{3}$$

where $\widehat{\boldsymbol{y}}_u$ is given by (2). In this case, the BLUP $\widehat{\boldsymbol{y}}_u \equiv \mathbb{E}(\boldsymbol{y}_u|\boldsymbol{y}_o)$ is also the mean squared optimal predictor.

Let the error covariance matrix corresponding to the observed set $\mathcal{O}$ be:

$$\Sigma_{\text{err}}(\mathcal{O}) := \mathbb{E}[(\widehat{\boldsymbol{y}}_u - \boldsymbol{y}_u)(\widehat{\boldsymbol{y}}_u - \boldsymbol{y}_u)^T|\boldsymbol{y}_o] = \mathbb{E}[(\widehat{\boldsymbol{y}}_u - \boldsymbol{y}_u)(\widehat{\boldsymbol{y}}_u - \boldsymbol{y}_u)^T] = \Sigma_{uu} - \Sigma_{uo}\Sigma_{oo}^{-1}\Sigma_{ou}, \tag{4}$$

where the second equality follows from (3). The objective functions considered in this study are:

(i) *A-optimality:*

$$\text{trace}(\Sigma_{\text{err}}(\mathcal{O})) \equiv \sum_{l\in\mathcal{U}} \text{Var}(y_l|\boldsymbol{y}_o) = \mathbb{E}\|\widehat{\boldsymbol{y}}_u - \boldsymbol{y}_u\|^2, \tag{5}$$

where $\|\cdot\|$ stands for the Euclidean norm;

[6]At each time instance, one can view the traffic volumes of each individual flow as independent r.v. with finite variance, and, therefore, by the *Central Limit Theorem* the sum of these individual flows can be approximated as a Gaussian r.v. (see also [4]).

[7]Henceforth, we shall assume that the matrix A has full row rank. Otherwise, our results can be shown to hold *mutatis mutandis* with $\Sigma_{oo}^{-1} = (A_o A_o^T)^{-1}$ viewed as the Moore-Penrose generalized inverse.

(ii) *E-optimality:*

$$\rho(\Sigma_{\text{err}}(\mathcal{O})) = \|\Sigma_{\text{err}}(\mathcal{O})\|_2, \tag{6}$$

where $\|\cdot\|_2$ stands for the spectral matrix norm, i.e., largest eigenvalue of $\Sigma_{\text{err}}(\mathcal{O})$. Then, the *optimal monitoring design* problem is given by:

Problem (Optimal Monitoring Design) *Find the optimal set* $\mathcal{O}^* \subseteq \mathcal{L} := \{1, 2, \ldots, L\}$ *such that:*

$$Z(\mathcal{O}^*) = \min_{\mathcal{O}\subseteq\mathcal{L},\, s.t.\, |\mathcal{O}|=K} f(\Sigma_{\text{err}}(\mathcal{O})), \tag{7}$$

where $Z(\mathcal{O}) = f(\Sigma_{\text{err}}(\mathcal{O}))$ *is the prediction error when monitoring the set of links* $\mathcal{O} \subseteq \mathcal{L}$ *and* $f(\cdot)$ *is the optimality criterion (*e.g. trace *or* spectral norm*).*

The *greedy heuristic* is a well-known method for finding approximate solutions to (7). Starting from an empty set $\mathcal{O}$, this heuristic amounts to incrementally adding to $\mathcal{O}$ the link that minimizes the prediction error Z (or equivalently, maximizes the error reduction). Let $\delta_j(\mathcal{O}) = Z(\mathcal{O}) - Z(\mathcal{O} \bigcup \{j\})$ be the error reduction when adding element j to the set $\mathcal{O}$. The formal algorithm is as follows.

Greedy Heuristic (Nemhauser *et al.* [23])

1) Let $\mathcal{O}^0 = \emptyset$, $\mathcal{N}^0 = \mathcal{L}$ *and set* $k = 1$.

2) At iteration k, *select* $i_k \in \mathcal{N}^{k-1}$ *such that*

$$i_k \in \arg\max_{i\in\mathcal{N}^{k-1}} \delta_i(\mathcal{O}^{k-1}) \tag{8}$$

with ties settled arbitrarily.

3) If $\delta_{i_k}(\mathcal{O}^{k-1}) \leq 0$ *then stop. Otherwise, set* $\mathcal{O}^k = \mathcal{O}^{k-1} \bigcup \{i_k\}$ *and* $\mathcal{N}^k = \mathcal{N}^{k-1} \setminus \{i_k\}$.

4) If $k = K$ *stop and output* $\mathcal{O}^K$. *Otherwise, set* $k = k + 1$ *and go to step 2).*

A naïve implementation of the described greedy algorithm for A- and E-optimality criteria has a complexity of $O(K^2L^3)$ and $O(KL^4)$, respectively. This is because operations such as matrix inversion, matrix multiplication and the calculation of the trace or spectral norm are involved whenever $\delta_i(\mathcal{O}^{k-1})$ is calculated (see (8) and Eqs.(3)-(6)). In our problem, however, there is a natural geometric structure related to PCA that can be used for developing fast heuristics. The connections to PCA also yield worst-case, lower bounds on the error in (7), which are of independent interest. These bounds are practical alternatives of the well-known $(1 - 1/e)$ approximation guarantees of [23] for polynomial-time, greedy heuristics. Unfortunately, these guarantees hold when the objective function is *submodular* [23], a property not satisfied by our objective functions. We present these new results next.

IV. A Principal Component Analysis Perspective

A. A geometric view of optimal prediction

We discuss next how our problem (7) relates to PCA. The covariance matrix $\Sigma_{\boldsymbol{y}}$ could be obtained via the routing equation (1) and the statistical characteristics of the J flows, i.e. $\Sigma_{\boldsymbol{y}} = R\Sigma_{\boldsymbol{x}}R^T$ or could be directly available through historical data from past link measurements (for more details on estimating the covariance see [4]). Without loss of generality, we decompose $\Sigma_{\boldsymbol{y}}$ using *singular value decomposition* or *Cholesky decomposition*, as $\Sigma_{\boldsymbol{y}} = AA^T$ for some $L \times J$ matrix A. In practice, we often readily have such a decomposition, with $A = R\Sigma_{\boldsymbol{x}}^{1/2}$, where R is the routing matrix and $\Sigma_{\boldsymbol{x}}$ is modeled as a diagonal or banded matrix (see [4] for a discussion on the statistical properties of the flows in the network). The row-vectors of the matrix A will be denoted as $\boldsymbol{a}_\ell \in \mathbb{R}^J,\ \ell = 1, \cdots, L$.

For convenience, we let $\Sigma_{\text{err}}(\mathcal{O}) = \mathbb{E}[(\boldsymbol{y} - \widetilde{\boldsymbol{y}})(\boldsymbol{y} - \widetilde{\boldsymbol{y}})^T]$, with $\boldsymbol{y}^T = (\boldsymbol{y}_u^T, \boldsymbol{y}_o^T)$ and $\widetilde{\boldsymbol{y}}^T = (\widehat{\boldsymbol{y}}_u^T, \boldsymbol{y}_o^T)$ be the error covariance matrix including both the observed and unobserved links. Using (4) one can show that $\Sigma_{\text{err}}(\mathcal{O}) = \Sigma_{\boldsymbol{y}} - \Sigma_o\Sigma_{oo}^{-1}\Sigma_o^T$, where $\Sigma_o = AA_o^T,\ \Sigma_{oo} = A_oA_o^T$, and $A_o = (a_{ij})_{i\in\mathcal{O}, j\in\mathcal{J}}$ is a submatrix of A with rows corresponding to the set of observed links $\mathcal{O}$. In practice, $\Sigma_{\boldsymbol{y}}$ is typically non-singular.

One thus obtains:

$$\Sigma_{\text{err}}(\mathcal{O}) = A(I_J - A_o^T(A_oA_o^T)^{-1}A_o)A^T = A(I_J - P_{\mathcal{O}})A^T, \tag{9}$$

where $P_{\mathcal{O}} := A_o^T(A_oA_o^T)^{-1}A_o$. The matrix $P_{\mathcal{O}}$ is the projection matrix onto the space $W_{\mathcal{O}} := \text{Range}(A_o^T)$ spanned by the row vectors $\{\boldsymbol{a}_i : \boldsymbol{a}_i \in \mathbb{R}^J, i \in \mathcal{O}\}$ of A_o. Therefore, $(I_J - P_{\mathcal{O}})$ is the projection matrix onto the orthogonal complement $\text{Range}(A_o^T)^\perp$.

Appendix A shows that $\|A - AP_{\mathcal{O}}\|_F^2 = \text{trace}(A(I_J - P_{\mathcal{O}})A^T)$ and $\|A - AP_{\mathcal{O}}\|_2^2 = \rho(A(I_J - P_{\mathcal{O}})A^T)$, where $\|\cdot\|_F$ denotes the Frobenius[8] norm. These facts together with (9) imply that the problem in (7) is not different than the combinatorial problem:

$$\mathcal{O}^* := \underset{\mathcal{O}\subseteq\{1,\cdots,L\}, |\mathcal{O}|=K}{\arg\min} \|A - AP_{\mathcal{O}}\|_\xi^2, \tag{10}$$

where $\|\cdot\|_\xi$ is the spectral norm for $\xi = 2$ and the Frobenius norm for $\xi = F$.

Problem (10) has a nice *geometric* interpretation. It seeks the "optimal" subspace $W_{\mathcal{O}} := \text{Range}(A_o^T)$ such that the distance, under the Frobenius[9] or spectral norm, of the matrix A to its row-wise projection $AP_{\mathcal{O}}$ is minimized.

[8] By definition $\|B\|_F := \sqrt{\text{trace}(BB^T)}$.

[9] The notion of distance is even more transparent in the Frobenius case – see Relation (30) in Appendix A.

B. PCA-based lower bounds

Observe that in (10), projection is restricted to subspaces spanned by subsets of K rows of A. Should one relax this constraint and optimize over arbitrary K-dimensional subspaces of $\mathbb{R}^J$, one would achieve a lower bound for the objective function. In this case, PCA analysis shows that the optimal space W^* is given by $P_K = V_K V_K^T$ where $V_K = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K)$ is the matrix of the K principal eigenvectors of $A^T A$ (see Appendix A). The above PCA-geometric observation gives the following *lower bounds* for the prediction error (see [19], [13] and Appendix A for proofs).

Theorem 1 (PCA lower bound for trace)**.** *Let K be the number of links allowed to be monitored. Let, also, $A^T A = V D V^T$ be the singular value decomposition (SVD) of $A^T A$ with $D = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_K, ..., \lambda_J)$, where $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_K \geq \cdots \geq \lambda_J \geq 0$. Then, the prediction error $\mathrm{trace}(\Sigma_{\mathrm{err}}(\mathcal{O}))$ is bounded as follows:*

$$\sum_{i=K+1}^{J} \lambda_i = \|A - AP_K\|_F^2 \leq \mathrm{trace}(\Sigma_{\mathrm{err}}(\mathcal{O})). \tag{11}$$

Theorem 2 (PCA lower bound for spectral norm)**.** *Similarly, the prediction error $\rho(\Sigma_{\mathrm{err}}(\mathcal{O}))$ for E-optimality has the following lower bound:*

$$\lambda_{K+1} = \|A - AP_K\|_2^2 \leq \rho(\Sigma_{\mathrm{err}}(\mathcal{O})). \tag{12}$$

The geometric structure of the problem also suggests efficient heuristics, discussed in detail in the next section. For example, we can think of a sequential "greedy" method that picks the space $W_{\mathcal{O}}$ that has smallest *"angle"* with the space W^*. Note, though, that the spaces $W_{\mathcal{O}}$ in (10) should be spanned by K vectors $\boldsymbol{a}_i$, $i \in \mathcal{L}$ (*i.e.*, corresponding to K observed links). Thus, in principle, the PCA-based lower bounds are strict, yet extremely useful since these bounds also hold for the exact optimal solution $\mathcal{O}^*$. Therefore, a small relative gap between the error of a heuristic and the PCA lower bound implies a good approximation to the value $Z(\mathcal{O}^*)$ of the optimal solution (for example, see Fig. 4).

V. Efficient Algorithms

The "naïve" implementation of the greedy heuristic is not feasible for large-scale networks. This section presents fast algorithms that substantially reduce this computational complexity (*e.g.,* see Fig. 5 for the speedup achieved). Motivated by the discussion at the end of the previous section, one idea is to first pick a link $i_1 \in \mathcal{L}$ for which the vector $\boldsymbol{a}_{i_1}$ is "closest" to the first PCA component $\mathbf{v}_1$. If $K = 1$, the procedure ends, and one can show – by (9) and Theorem 3 of the sequel – that this selection is very close to the *optimal* choice where the notions of "close" and "optimal" depend on the type of norm or optimality criterion. If $K > 1$, we "subtract" the effect of the chosen link i_1 by considering only the

orthogonal projections of the $\boldsymbol{a}_i$'s for the remaining links onto the space $\text{span}\{\boldsymbol{a}_{i_1}\}^{\perp}$. We then construct a new matrix A with rows given by the above projections and repeat the procedure iteratively K times.

Formally, the matrix A is updated as follows. Assume the iterative procedure selects link i_k at step k to be added to the set of selected links $\mathcal{O}$. We update A with the rule

$$A^{(k)} = A^{(k-1)}\Big(I_J - \frac{\boldsymbol{a}_{i_k}^{(k-1)}\boldsymbol{a}_{i_k}^{(k-1)^T}}{\|\boldsymbol{a}_{i_k}^{(k-1)}\|^2}\Big), \tag{13}$$

with $A^{(0)} = A$ and the row-vectors of $A^{(k-1)}$ being $\boldsymbol{a}_i^{(k-1)} \in \mathbb{R}^J, i \in \{1, 2, \ldots, L\}$. The following proposition justifies this method, by establishing that the proposed procedure can be used to sequentially update the prediction error covariance $\Sigma_{\text{err}}(\mathcal{O})$. This sequential projection property is the key behind the computational efficiency of the proposed heuristics presented in the sequel, since it allows us to avoid expensive operations such as matrix inversions. Its proof is given in Appendix B.

Proposition 1 (Sequential Computation of $\Sigma_{\text{err}}(\mathcal{O})$)**.** Let $\mathcal{O}^k$ be the set of selected links at the end of step k, and $A^{(k)}$ the iteratively updated matrix, as shown in Eq. (13). Then,

$$A^{(k)}A^{(k)^T} = \Sigma_{\text{err}}(\mathcal{O}^k) \text{ where } \Sigma_{\text{err}}(\mathcal{O}^k) \equiv A(I_J - A_{o_k}^T(A_{o_k}A_{o_k}^T)^{-1}A_{o_k})A^T. \tag{14}$$

A. Detailed description of algorithms

Our first algorithm, named PCAPH, is well suited for both *A- and E-optimality*. Essentially we implement the geometric idea discussed above where, at step k, we select the link $i_k \in \mathcal{L}$ whose row-vector $\boldsymbol{a}_{i_k}$ is closest to the principal component of the current version of A. We can choose between two options for vector proximity; the smallest angle or the longest projection. We decided to work with the latter. We obtain the principal component using the *power method* [19]. The steps of the algorithm are:

Algorithm 1 (PCA Projection Heuristic - PCAPH)**.** *Let* $\mathcal{O}^0 = \emptyset$ *and* $A^{(0)} = A$*. Set* $k = 1$.

1) POWER METHOD STEP: *Using the* power method *obtain a fast approximation to the principal eigenvector* $\mathbf{v}_1$ *of* $A^{(k-1)^T}A^{(k-1)}$.

2) SELECTION: *At iteration* k*, choose:*

$$i_k \in \underset{i \in \mathcal{L}\setminus\mathcal{O}^{k-1}}{\arg\max} |\mathbf{v}_1^T \boldsymbol{a}_i^{(k-1)}|$$

where $\boldsymbol{a}_i^{(k-1)} \in \mathbb{R}^J, i \in \{1, 2, \ldots, L\}$ *are the row-vectors of* $A^{(k-1)}$*. Put* i_k *in the list of links to be monitored, i.e.,* $\mathcal{O}^k := \mathcal{O}^{k-1} \bigcup \{i_k\}$.

3) PROJECTION/ERROR REDUCTION: *Update matrix* $A^{(k)}$. *The rows of the matrix* $A^{(k)}$ *are the orthogonal projections of the rows of* $A^{(k-1)}$ *onto* $(\text{span}\{\boldsymbol{a}_{i_k}^{(k-1)}\})^{\perp}$. *Formally,*

$$A^{(k)} = A^{(k-1)}\Big(I_J - \frac{\boldsymbol{a}_{i_k}^{(k-1)}\boldsymbol{a}_{i_k}^{(k-1)^T}}{\|\boldsymbol{a}_{i_k}^{(k-1)}\|^2}\Big). \tag{15}$$

4) Set $k = k + 1$. *If* $k < K$, *go to step 1).*

As shown in Section VII, this strategy usually yields a monitoring design with slightly larger prediction error than the greedy strategy, but is orders of magnitude faster in execution time. Specifically, step 1) requires $O(mJL)$ computations, where m is the number of iterations the power method executes ($m \ll \min\{L, J\}$). We need $O(LJ)$ computations per iteration for the matrix-vector multiplication in the power method loop. Steps 2) and 3) require $O(LJ)$ operations.

Lemma 1. *The complexity of the PCA Projection Heuristic is* $O(mKLJ)$.

The next algorithm is named FGE and is tailored for *A-optimality*. It represents a fast implementation of the classical greedy algorithm, since it avoids calculating the inverse of the covariance matrix Σ_{oo}. Instead, at each iteration we seek the column vector that maximizes the error reduction. This is equivalent to finding the vector that maximizes the squares of the projections of the remaining vectors onto itself. For A-optimality, one can show that the *error reduction* at each step k, given that links $\{i_1, ..., i_{k-1}\}$ were already chosen, is equal to:

$$R_k(i) := R_{\{i_1,...,i_{k-1}\}}(i) = \sum_{j=1}^{L} \frac{|\boldsymbol{a}_j^{(k-1)^T}\boldsymbol{a}_i^{(k-1)}|^2}{\|\boldsymbol{a}_i^{(k-1)}\|^2}, \tag{16}$$

with $i \in \mathcal{L} \setminus \{i_1, ..., i_{k-1}\}$. In words, it equals the sum of the squares of the projections to space $\text{span}\{\boldsymbol{a}_i^{(k-1)}\}$. This step is accomplished by the first step of the algorithm. The second step, updates the matrix $A^{(k)}$.

Algorithm 2 (Fast Greedy Exact - FGE)**.** *Let* $\mathcal{O}^0 = \emptyset$ *and* $A^{(0)} = A$. *Set* $k = 1$.

1) SELECTION: *At iteration* k, *choose:*

$$i_k \in \underset{i \in \mathcal{L}\setminus\mathcal{O}^{k-1}}{\arg\max} \sum_{j=1}^{L} \frac{|\boldsymbol{a}_j^{(k-1)^T}\boldsymbol{a}_i^{(k-1)}|^2}{\|\boldsymbol{a}_i^{(k-1)}\|^2} \tag{17}$$

where $\boldsymbol{a}_i^{(k-1)} \in \mathbb{R}^J, i \in \{1, 2, \ldots, L\}$ *are the row vectors of* $A^{(k-1)}$. *Set* $\mathcal{O}^k = \mathcal{O}^{k-1} \bigcup \{i_k\}$.

2) PROJECTION/ERROR REDUCTION: *Do step 3) of Algorithm 1.*

3) Set $k = k + 1$. *If* $k < K$, *go to step 1).*

Step 1) is a "greedy step" since it picks the link that reduces the error the most. It requires $O(JL^2)$ operations while, step 2) requires $O(LJ)$ operations after suitably rearranging the order of operations.

Lemma 2. *The computational complexity of the Fast Greedy Exact algorithm is* $O(KJL^2)$.

Remark 1. *PCAPH relies on the fast performance of the power method, whose convergence speed depends on the ratio* $|\lambda_2|/|\lambda_1|$ *of the matrix under study. Thus, one might think that the performance of PCAPH may deteriorate when that ratio is close to* 1. *However, it is not affected because: a) We are only interested in approximating the principal eigenvector so a few iterations of the power method suffice, and b) other iterative methods could be used instead, such as the Rayleigh quotient iteration [19] that has a cubic convergence speed when an approximate eigenvector is provided (say, from the power method).*

For the E-optimality criterion, we present a very fast implementation of the greedy heuristic, namely FGR. Relation (9) and Proposition 1 allow us to avoid computationally expensive operations like matrix inversion and singular value decomposition, which leads to drastic improvements in performance. The next algorithm was motivated by the following characterization of the largest eigenvalue [19]:

$$\rho(\Sigma_{\text{err}}(\mathcal{O})) := \lambda_1(\Sigma_{\text{err}}(\mathcal{O})) = \max_{\boldsymbol{z}\in\mathbb{R}^L, \|z\|=1} \boldsymbol{z}^T \Sigma_{\text{err}}(\mathcal{O}) \boldsymbol{z}. \tag{18}$$

Algorithm 3 (Fast Greedy Randomized - FGR)**.** *Let* $\mathcal{O}^0 = \emptyset$ *and* $A^{(0)} = A$. *Set* $k = 1$.

1) INITIALIZATION: *Generate* m *independent, normally distributed random vectors* $\mathbf{x}_i \in \mathbb{R}^L, i = 1, 2, \ldots, m$ *from* $\mathcal{N}(0, I_L)$ *and set* $\mathbf{z}_i := \mathbf{x}_i / \|\mathbf{x}_i\|$.

2) SAVINGS STEP: *At iteration* k, $k = 1, 2, \ldots, K$, *calculate:*

$$c_i := (\mathbf{z}_i^T A^{(k-1)}) \cdot (A^{(k-1)^T} \mathbf{z}_i), \forall i = 1, 2, \ldots, m. \tag{19}$$

3) SELECTION: *At iteration* k, *select:*

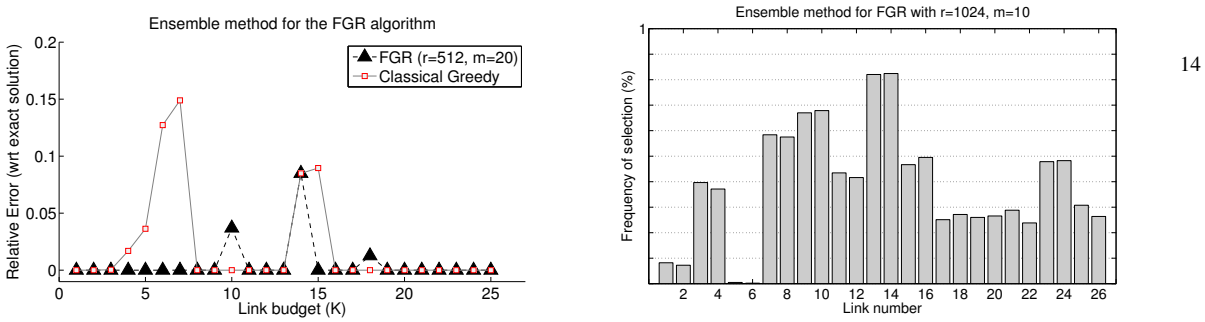
$$j_k \in \arg\min_{j \in \mathcal{L} \setminus \mathcal{O}^{k-1}} \Big\{ \max_{i=1,\ldots,m} [c_i - \frac{\mathbf{z}_i^T \mathbf{b}_j^{(k-1)} \mathbf{b}_j^{(k-1)^T} \mathbf{z}_i}{\|\boldsymbol{a}_j^{(k-1)}\|^2}] \Big\}, \tag{20}$$

where $\mathbf{b}_j^{(k-1)} := A^{(k-1)} \boldsymbol{a}_j^{(k-1)}$ *and* $\boldsymbol{a}_j^{(k-1)} \in \mathbb{R}^J, j \in \{1, 2, \ldots, L\}$ *are the column vectors of* $A^{(k-1)^T}$. *This corresponds to finding the link* j *that minimizes the error* $\|A^{(k-1)}(I - \frac{\boldsymbol{a}_j^{(k-1)} \boldsymbol{a}_j^{(k-1)^T}}{\|\boldsymbol{a}_j^{(k-1)}\|^2}) A^{(k-1)^T}\|_2$. *Set* $\mathcal{O}^k = \mathcal{O}^{k-1} \bigcup \{j_k\}$.

4) PROJECTION/ERROR REDUCTION: *Update matrix* $A^{(k)}$. *I.e.,*

$$A^{(k)} = A^{(k-1)} \Big(I_J - \frac{\boldsymbol{a}_{j_k}^{(k-1)} \boldsymbol{a}_{j_k}^{(k-1)^T}}{\|\boldsymbol{a}_{j_k}^{(k-1)}\|^2} \Big). \tag{21}$$

5) Set $k = k + 1$. *If* $k < K$, *go to step 2).*



(a) Relative error (w.r.t. optimal solution)

(b) Link selection (for $K = 10$)

Fig. 2. Ensemble (distributed implementation) of FGR algorithm for Internet2 topology. (Left) Ensemble of FGR yields the exact optimal solution 88% of the time, whereas the classical greedy 76%. (Right) Link selection frequency. Links that connect West to East sites (*e.g.* links 13, 14 and 9, 10 that correspond to KANS–CHIC and SALT–KANS) are selected the most often by our randomized algorithm. Information from links 5, 6 is redundant, so these links are almost never selected.

In step 1), we randomly sample m unit vectors $\mathbf{z}_i$ such that $\mathbf{z}_i \in \mathbb{R}^L, i = 1, 2, \ldots, m, \|\mathbf{z}_i\| = 1$. We use these vectors in step 2) and 3) to approximate the maximum in (18) and hence the largest eigenvalue. In step 3), we choose the vector (link) that minimizes the error expressed through the largest eigenvalue. Finally, in step 4), we update matrix A for use in the next iteration. Step 2) requires[10] $O(mLJ)$ operations and step 3) $O(mL^2J)$.

Lemma 3. *The computational complexity of the Fast Greedy Randomized algorithm is* $O(mKL^2J)$.

The following proposition establishes a theoretical bound on the quality of approximating $\tilde{\lambda}_1$, the largest eigenvalue of a matrix Σ. The proof is given in the Appendix. It requires some prior information for the eigenvalues of Σ. Such information might be gathered by computing the eigenvalues *once*, before the algorithm starts. That is, we may compute the eigenvalues only for the initial matrix $A^{(0)}A^{(0)^T}$. Note that approximation of the largest eigenvalue is needed whenever the *Selection* step of Algorithm 3 is executed.

Proposition 2. Let λ_1 be the true eigenvalue, and $\tilde{\lambda}_1$ the approximated one using the heuristic described in Algorithm 3. Let also m be the number of random vectors $\mathbf{z}_i \in \mathbb{R}^L$ used in the heuristic. The eigenvalues of matrix Σ are $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_L$, and $c_1 \geq c_2 \geq \ldots c_L$ with $c_i = \lambda_i/\lambda_1$ the normalized values thereof. For any positive scalar $\varepsilon > 0$, there exists an index t such that $c_i > 1 - \varepsilon$, $\forall i \leq t$ and $c_i \leq 1 - \varepsilon$

[10] Note that step 2) can be avoided, if we save the $c_i, i = 1, ..., m$ from the previous iteration. Indeed, at iteration $k+1$, we have $A^{(k)} = A^{(k-1)}\left(I_J - \frac{\boldsymbol{a}_{j_k}^{(k-1)}\boldsymbol{a}_{j_k}^{(k-1)^T}}{\|\boldsymbol{a}_{j_k}^{(k-1)}\|^2}\right)$. Therefore, for step 2) at iteration $k+1$, we have $c_i^{(k+1)} = (\mathbf{z}_i^T A^{(k)}) \cdot (A^{(k)^T}\mathbf{z}_i) = c_i^{(k)} - \frac{\mathbf{z}_i^T \mathbf{b}_{j_k}^{(k)} \mathbf{b}_{j_k}^{(k)^T} \mathbf{z}_i}{\|\boldsymbol{a}_{j_k}^{(k)}\|^2}$, where $\mathbf{b}_{j_k}^{(k)} = A^{(k)}\boldsymbol{a}_{j_k}^{(k)}$. This improvement leads to extra $O(mLJ)$ savings per iteration.

otherwise. Then,

$$\mathbf{P}\Big(\frac{\tilde{\lambda}_1}{\lambda_1} > 1-\varepsilon\Big) \geq 1 - \Big[F_{t,L-t}\Big(\frac{L-t}{t}\frac{1-\varepsilon}{c_t-(1-\varepsilon)}\Big)\Big]^m, \tag{22}$$

where $F_{\nu_1,\nu_2}(\cdot)$ is the cumulative distribution function for the F-distribution with parameters ν_1 and ν_2.

Applying Proposition 2 on the Internet2 network (see Section VII) that has $L = 26$ links we obtain the following insights: the probability of having a better than 63% precision ($\approx 1 - 1/e$) for λ_1 is 0.94 when we choose $m = 500,000$. A precision of 80% can be achieved with probability 0.95 when $m = 10^9$. However, in practice m could be orders of magnitude smaller. As shown in Section VII (see Fig. 5(b)) we get low prediction error with just $m = 100$. We obtain even further improvements when employing parallelization and the methods of *ensembling* discussed in the section that follows (see Tables I and II). The reason of dissimilarity between Proposition 2 and practical values for m is this. Our proposition is about approximating the spectral norm needed in step 3) of the algorithm; nevertheless, a crude approximation of that value is sufficient as long as the order by which links enter the selection set coincides with the order that an algorithm with sufficiently large m would generate. A theoretical justification of this argument is an open problem, but numerical justification is provided via the aforementioned tables and figures and additional results in Section VII.

B. Ensemble methods for randomized algorithms

Algorithm FGR can be characterized as a *randomized algorithm*. It uses m random vectors to approximate λ_1, needed to accomplish the *Selection* step. PCAPH may also be implemented in a randomized fashion; since it involves the power method for approximating $\mathbf{v}_1$, one can utilize a random vector for the method's initial value. This randomization gives rise to the idea of ensembling.

The main idea of the *ensemble* method is to pick a small m – which makes the algorithm much faster – and run several, say r, independent instances of the algorithm, in parallel. We then select the solution set that yields the minimum prediction error among the ensemble. Note that unlike the greedy approach, the resulting solution sets $\mathcal{O}^K$ are no longer nested, i.e. some links may be excluded from $\mathcal{O}^K$ and replaced with others as K grows. This helps avoiding one of the artificial constraints that greedy procedures impose on the solution sets.

Our experiments with the topology[11] of Fig. 1, indicate that for large enough r, ensemble methods can often come close and, in fact, yield the optimal solution of Problem (7). *E.g.*,, Fig. 2(a) shows the results of a distributed implementation of FGR with $r = 512$ and $m = 20$. Comparing with a single execution of the classical greedy (or even FGR with $r = 1$, $m \gg 20$), we note that the exact optimal solution is obtained 88% of the time. The solution can be obtained in minutes, rather than hours, needed for the integer programming formulation that yields the exact solution. Fig. 2(b) shows the selection frequency of each link when FGR is employed with $r = 1024, m = 10$. It can be seen that links bridging East with West locations of the topology have the highest probability of being included in the optimal solution.

Furthermore, as verified in Table I and Table II, randomization allows the network designer to adhere to smaller values for m when the option of parallel or distributed implementation of the algorithm is available. The two tables show the error of the FGR algorithm (for a link budget $K = 14$) for the Internet2 network (Table I) and a considerably larger network of $L = 195$ links and $J = 500$ flows (Table II). Several versions of FGR were run in Matlab on an 8-core computer, for different combinations of m and amount of parallelization r. Should enough resources be available, little time is needed for computing a monitoring set with less prediction error than the "naïve" implementation of the greedy heuristic. For example, for $m = 64$ and $r = 256$ we get a relative error of 4.8%, whereas greedy gives 8.5%.

Our numerical results suggest that, given a specific "resource/computation budget" (i.e., the product of m and r), a sound balance between m and r should be one that keeps the ratio m/r at around $32/128$ or $128/32$. Of course, if more resources are available and the amount of parallelization can be increased, both the efficiency and the accuracy of our randomized algorithm can be dramatically increased (see the rightmost column in the two tables). In general, the "optimal" ratio between m and r depends on the routing matrix A and the topology of the network (that translate to the "spectrum" of the matrix $A^T A$, see also subsection VII-A). It is wise, though, not to increase r excessively and decrease m to very low values, or vice versa.

VI. Performance Guarantees for Error Reduction

This section introduces performance guarantees for the error reduction (see Eq. (16)) that can be achieved by algorithms PCAPH and FGE under the A-optimality criterion. Similar error bounds, though

[11]The ID's of the 26 links of the Internet2 network are: 1 (2): Los Angeles → Seattle, 3 (4): Seattle → Salt Lake City, 5 (6): Los Angeles → Salt Lake City, 7 (8): Los Angeles → Houston, 9 (10): Salt Lake City → Kansas City, 11 (12): Kansas City → Houston, 13 (14): Kansas City → Chicago, 15 (16): Houston → Atlanta, 17 (18): Chicago → Atlanta, 19 (20): Chicago → New York, 21 (22): Chicago → Washington, 23 (24): Atlanta → Washington, 25 (26): Washington → New York. Odd Link ID's correspond to the uplink direction and the even to downlink; i.e. Link 7 is the Los Angeles to Houston link and Link 8 is the Houston to Los Angeles link.

TABLE I

ENSEMBLING FOR VARIOUS VALUES OF m AND r ON INTERNET2 NETWORK FOR $K = 14$. WE RAN 50 REPLICATIONS FOR EACH (m, r) PAIR TO OBTAIN THE RELATIVE MEAN ERROR (RME) W.R.T THE EXACT SOLUTION, AND 95% CONFIDENCE INTERVALS. CLASSICAL GREEDY'S RME IS 8.5%. THE EXPERIMENTS WERE RUN IN MATLAB ON AN 8-CORE COMPUTER.

(m,r)	**(32000,1)**	**(1024,4)**	**(512,8)**	**(256,16)**	**(128,32)**	**(64,64)**	**(32,128)**	**(16,256)**	**(8,512)**	**(4,1024)**	**(64,256)**
95% CI (ub)	10.6	8.6	8.5	8.1	8.0	7.8	7.8	7.8	8.1	7.8	5.1
RME (%)	**10.2**	**8.4**	**8.5**	**8.0**	**7.8**	**7.6**	**7.6**	**7.6**	**8.0**	**7.6**	**4.8**
95% CI (lb)	9.8	8.3	8.5	7.8	7.6	7.4	7.4	7.4	7.8	7.4	4.4

TABLE II

ENSEMBLING FOR VARIOUS VALUES OF m AND r ON A NETWORK WITH $L = 195$ LINKS AND $J = 500$ FLOWS FOR $K = 14$. FOR EACH (m, r) WE ILLUSTRATE THE ABSOLUTE PREDICTION ERROR, AND 95% CONFIDENCE INTERVALS. CLASSICAL GREEDY'S PREDICTION ERROR IS 30.58; FOR $m = 26$, $r = 256$ WE ACHIEVE A FURTHER ERROR REDUCTION CLOSE TO 5%.

(m,r)	**(100,1)**	**(1024,4)**	**(512,8)**	**(256,16)**	**(128,32)**	**(64,64)**	**(32,128)**	**(4,1024)**	**(64,256)**
95% CI (ub)	37.39	31.90	30.68	29.92	29.75	29.85	29.75	31.26	29.63
Error (%)	**32.18**	**31.17**	**30.29**	**29.78**	**29.69**	**29.70**	**29.68**	**30.72**	**29.57**
95% CI (lb)	26.97	30.44	29.90	29.64	29.63	29.55	29.61	30.17	29.52

very loose, are provided in [5]. Note that the $(1 - 1/e)$ bounds for greedy algorithms developed by Nemhauser/Wolsey [16] cannot be claimed, since the submodularity property does not hold for our objective functions. For proofs see Appendix C.

Theorem 3. *Suppose that at iteration k the observed set $\mathcal{O}^k = \{i_1, ..., i_k\}$, and the resulting matrix is $A^{(k)}$. Let also $\boldsymbol{a}_i^{(k)}$ be a candidate vector for selection by the algorithm on step $k+1$. Then, the error reduction (see Eq. (16)) at step $k+1$ is bounded below as follows:*

$$\sum_{j=1}^{L} \frac{|\boldsymbol{a}_j^{(k)^T} \boldsymbol{a}_i^{(k)}|^2}{\|\boldsymbol{a}_i^{(k)}\|^2} \geq (\gamma^{(k)})^2 \lambda_1^{(k)} - 2\gamma^{(k)} \sqrt{1 - (\gamma^{(k)})^2} \sqrt{\lambda_1^{(k)}} \sqrt{\sum_{j=1}^{L} \lambda_j^{(k)}}, \tag{23}$$

*where $\lambda_j^{(k)}$ is the jth largest eigenvalue of $A^{(k)^T} A^{(k)}$ (*i.e.*, after we have updated our matrix at the previous step k, see (13)), and $\gamma^{(k)}$ is the cosine (*i.e.*, see Supplementary section) between the principal eigenvector $\mathbf{v}_1^{(k)}$ of $A^{(k)^T} A^{(k)}$ and the selected vector $\boldsymbol{a}_i^{(k)}$, i.e.*

$$\gamma^{(k)} = \frac{|\mathbf{v}_1^{(k)^T} \boldsymbol{a}_i^{(k)}|}{\|\boldsymbol{a}_i^{(k)}\|}. \tag{24}$$

Theorem 4. *Suppose that the conditions of Theorem 3 hold. Then, at iteration $k+1$, the error reduction is bounded from above as follows:*

$$\sum_{j=1}^{L} \frac{|\boldsymbol{a}_j^{(k)T} \boldsymbol{a}_i^{(k)}|^2}{\|\boldsymbol{a}_i^{(k)}\|^2} \leq \lambda_2^{(k)} + (\gamma^{(k)})^2 \lambda_1^{(k)} + 2\gamma^{(k)} \sqrt{1-(\gamma^{(k)})^2} \sqrt{\lambda_1^{(k)}} \sqrt{\sum_{j=1}^{L} \lambda_j^{(k)}}, \tag{25}$$

*where $\lambda_j^{(k)}$ is the jth largest eigenvalue of $A^{(k)T}A^{(k)}$ (*i.e.*, after we have updated our matrix at the previous step k, see (13)), and $\gamma^{(k)}$ is as defined in Eq. (24).*

A demonstration of these Theorems for FGE is provided in the Supplemental material.

Remark 2 (Interpretation of Theorems 3 and 4)**.** *Say that the algorithm selects the link that corresponds to vector $\boldsymbol{a}_i^{(k)}$ which is at the same direction as $\mathbf{v}_1^{(k)}$. Then, $\gamma^{(k)} = 1$ and our bounds suggest a significant reduction, with value between $\lambda_1^{(k)}$ and $\lambda_1^{(k)} + \lambda_2^{(k)}$. On the other hand, if the algorithm selects a vector with $\gamma^{(k)} = 0$, then the best reduction we should expect to achieve is $\lambda_2^{(k)}$. This is because we are essentially selecting from vectors that are perpendicular to $\mathbf{v}_1^{(k)}$, and the "favorable choice" would be some vector that is co-directional with the second principal axis $\mathbf{v}_2^{(k)}$. For all other values of $\gamma^{(k)}$, the error reduction is in between the values given by the theorems, taking into account, of course, that it should be non-negative and not greater than the error reduction given by the PCA bound of Theorem 1.*

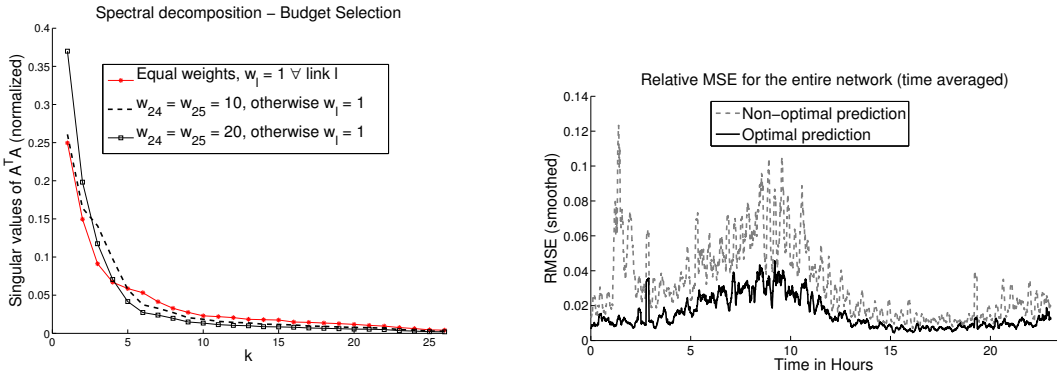
Remark 3 (E-optimality bounds)**.** *The E-optimality error reduction bound is given in Theorem 2 which claims that at iteration $k+1$ we can expect a reduction at most equal to $\lambda_1^{(k)}$ (see proof in Appendix A). In addition, we have the trivial lower bound that error reduction should be non-negative. At this moment, existence of tighter bounds remains an open problem.*

VII. Performance Evaluation

This section evaluates our algorithms. We start with a short discussion for the choice of budget K, and then proceed with various comparisons of the proposed algorithms, for different network sizes and topologies, against alternative methods including the ones studied in [5], [17].

A. Choosing the "budget" K

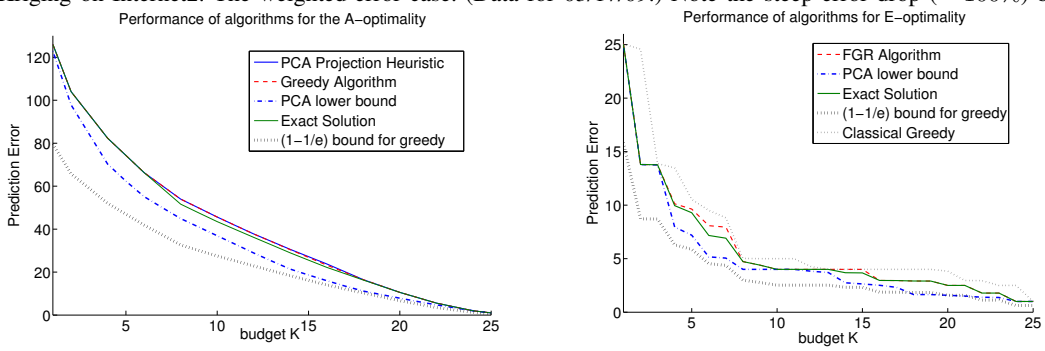
Thus far, we have assumed that the number of links to be monitored, namely the "budget" K, is known to the network operator. However, in practice, this is an unknown parameter dependent on the monetary budget available. We now address the question of choosing an adequate value for K, so that the network operator can spend the least amount of money while monitoring the network with sufficient accuracy. The answer comes from PCA via the spectral decomposition of the matrix A^TA. One wants to choose K according to the "spectrum" of the data. Specifically, K should be such that (see Section 6.1 [29] for



(a) Spectra of $A^T A$: Budget selection

(b) Weighted RMSE.

Fig. 3. Kriging on Internet2. The weighted-error case. (Data for 03/17/09.) Note the steep error drop ($\approx 100\%$) between the two pre



(a) A-optimal criterion

(b) E-optimal criterion

Fig. 4. Evaluation of fast algorithms on Internet2 topology, and comparison with the exact optimal solution. Further, the PCA lower bound is depicted, together with the loose (1-1/e) bound proposed by Nemhauser/Wolsey [16]. Note however that the bound of [16] cannot be claimed, due to absence of submodularity.

more details and other criteria):

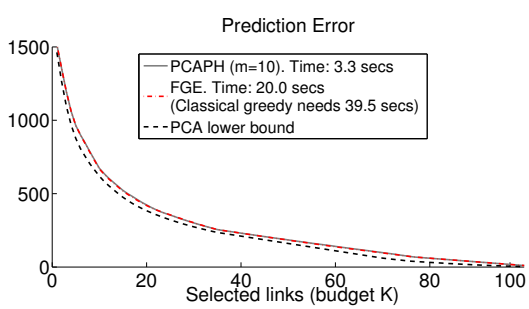
$$\sum_{i=1}^{K} \lambda_i \geq 0.80 \times \mathrm{trace}(A^T A), \tag{26}$$

where $(\lambda_1 \geq \cdots \geq \lambda_p)$, $p = \min\{L, J\}$, are the eigenvalues of the matrix $A^T A$.

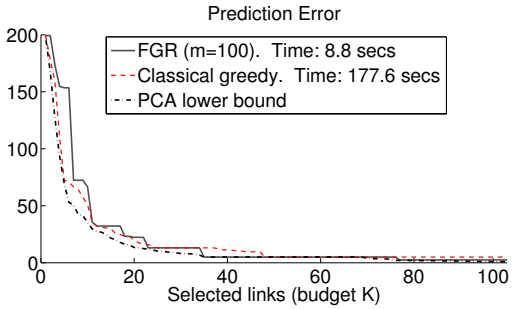
Fig. 3(a) shows the "spectra" of three signals, based on the Internet2 traffic data obtain on March 17, 2009. We plot the case of uniform priority, the case where links 24 and 25 are assigned weights $w_l = 10$, and the case where $w_l = 20$ for $l = 24, 25$. (More details on weighted link monitoring follow on subsection VII-E.) Note that as the importance of links increases the "energy" of the signal is concentrated on fewer singular values. This is intuitively appealing since it suggests that the prediction error will be reduced the most if we can afford a budget K that is at least as large as the number of the high importance links.

B. Heuristic Vs. Exact algorithms

We use a real-world network, namely Internet2, to evaluate our fast algorithms against the exact solution obtained via an exhaustive search. We also calculate and demonstrate the PCA lower bound. Internet2



(a) A-optimality



(b) E-optimality

Fig. 5. Prediction error on a network with $L = 195$ links, $J = 500$ flows. Note the dramatic decrease in computation time between the naïve greedy and the proposed algorithms.

(formerly Abilene) involves $L = 26$ links, $N = 9$ nodes and $J = 72$ routes (see [27], [6]). For simplicity, we assume that the flow-covariance matrix is $\Sigma_x = I_J$. In Fig. 4(a) we examine the trace criterion and in Fig. 4(b) the spectral norm one[12]. In both cases, the exhaustive search required several hours to converge to a solution (the implementation was done in Matlab using CPLEX). On the contrary, the heuristics (PCAPH and greedy algorithms) terminate in a few seconds and yield a solution very close to the optimal one. We also demonstrate the PCA lower bound of Theorems 1 and 2 accompanied by the $1 - 1/e$ bound proposed in [16] for minimization of submodular functions. Even though the latter cannot be claimed for our algorithms due to lack of submodularity, we note that our PCA-based bound is much tighter. In the sequel, we will make use of our PCA lower bounds to evaluate the quality of approximation in scenarios where the exact solution is not available.

C. Comparison of Algorithms

We next juxtapose the computational performance of the proposed algorithms against the classical greedy heuristic and the algorithm proposed in [5]. The comparisons against the naïve greedy are performed on a simulated network of $N = 100$ nodes. The topology is generated using preferential attachment, as described in [30]. In particular, we created a routing matrix R of $L = 195$ links and $J = 500$ flows (the methodology is explained below). For this case, we assume independent flows with unit variance, i.e. $\Sigma_{\boldsymbol{x}} = I_J$ and, thus, $A = R$ (see section IV-A). Fig. 5(a) illustrates the results for A-optimality. The proposed algorithms are notably faster than the naïve implementation of the greedy algorithm. Specifically, PCAPH is $10\times$ and FGE $2\times$ faster than classical greedy. The PCAPH algorithm significantly outperforms all other algorithms at the expense of having a slightly larger prediction error. Fig. 5(b) depicts the case of E-optimality. Again, our algorithm performs significantly faster ($20\times$ faster) than the naïve implementation of the greedy heuristic. We used $m = 100$ random vectors for the FGR

[12]These numerical results are also tabulated in the Supplementary information section.

algorithm. In both figures, we use the PCA lower bound to qualitatively assess the solutions of the algorithms, since obtaining the exact solution is not computationally feasible.

We then compare PCAPH (with $m = 5$) against the algorithm proposed in [5] for network kriging of end-to-end network properties. The algorithm makes heuristic use of QR-factorization with column pivoting (henceforth, named "QR") and is an adaptation of the method proposed in [19] for subset selection. The QR algorithm is also adapted in [17]. For a budget of K links and an input matrix A, with L rows and J columns, QR has a complexity of $O(L^2J) + O(K^2L)$. The first term involves the computation of SVD of A, and the second corresponds to the QR factorization step. Clearly, its complexity is similar to the one of PCAPH (see Lemma 1), and thus we evaluate the two methods on the A- and E-optimality criteria. The comparisons are based on real-world data obtained from CAIDA [18]. Comparisons of QR with FGE and FGR are given in the Supplemental material.

The results on *prediction accuracy* are illustrated in Fig. 6. The figure shows 18 different scenarios based on a real-world network topology obtained from CAIDA [18]. Using these dada, we constructed a different input matrix A for each scenario, as follows. We first "pruned" the large network topology given in [18] to smallest topologies of the $N = 50$ (Figs. 6(a) and 6(b)), $N = 75$ (Figs. 6(c) and 6(d)) and $N = 100$ (Figs. 6(e) and 6(f)) highest-degree nodes. Then, for all node pairs (each source-destination pair (S, D) corresponds to a flow) of these topologies, we calculate the shortest path using Dijkstra's algorithm. This step provides the routing matrix R for the three "pruned" networks. Finally, we simulate the covariance matrix Σ_x of the flows as described next. Each subfigure of Fig. 6 corresponds to 3 different flow covariance matrices: (i) in each leftmost subplot we assume independent flows, i.e., $\Sigma_x = I_J$, (ii) in each middle subplot we have flows with variance simulated from the Pareto distribution with parameter α, and we also assume a correlation ρ_1 between a forward flow and its reverse, and (iii) each rightmost subplot corresponds to a flow covariance matrix constructed as in (ii) and additionally we assume a correlation $\rho_2 \ll \rho_1$ between the remaining flows[13]. With these 3 different structures for Σ_x the comparisons of PCAPH against QR on the input matrix $A = R\Sigma_x^{1/2}$ involve cases with decreasing matrix sparsity. The specific simulation parameters (α, ρ_1, ρ_2) can be found in the Supplemental file.

We utilize a randomized version of PCAPH (i.e., randomization on the initial vector of the power method) and run it independently for $n = 50$ times. Fig. 6 shows the mean performance of PCAPH. More details including tables of 95% confidence intervals are given in the Supplemental material. Clearly, the approximation quality of PCAPH is superior than the one of QR. In all plots of Fig. 6, executed for budgets $K = 1, \ldots, 100$, the prediction error that PCAPH yields is much smaller than QR's error, except

[13]Structures (ii)-(iii) are not unreasonable cases in real-world, since in periods of congestion TCP introduces higher dependence between the forward flow and its reverse, and weaker dependence between all other network flows (see [31], [27], [22]).

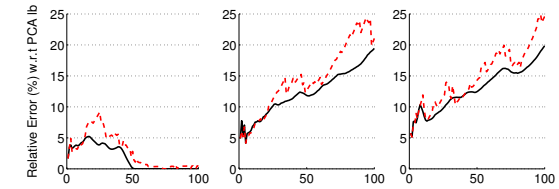
for some small K's for which the algorithms are comparable. The intuition is that at the beginning, both algorithms tend to choose the same links. Essentially, for small K's there are only few "good options", and hence the algorithms choose similar links (the ones offering the largest error reduction). Additionally, for small K's, the low relative error suggests that the PCA lower bound is close to the exact solution. This means that there is not much room for differences in the error between the two methods. However, as K increases and especially as the matrix A becomes less sparse, the advantages of PCAPH are evident.

The dominance of PCAPH over QR becomes even more pronounced when PCAPH with ensembling (with $r = 50$) is used, as Fig. 7(a) depicts. For Fig. 7(a) we calculate the ratio *relative error of QR (w.r.t. PCA lower bound)* over *relative error of PCAPH (w.r.t. PCA lower bound)* and plot the histogram of these quantities obtained for $K = 1, \ldots, 100$. PCAPH is much better (can be up to 1.6 times better for some cases of A-optimality, and up to 60 times better for some cases of E-optimality) or at least as good as QR in the majority of the cases examined (it is inferior only in 3 out of total 400 cases, see second from the left plot). The parameters for the scenarios used can be found in the Supplemental material.

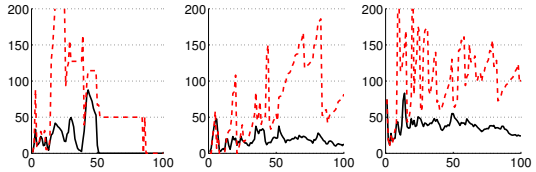
To illustrate the computational performance of the two algorithms we utilized the CAIDA-based scenario of $N = 100$ nodes. The results are shown in Figure 7(b). As expected from the theoretical bounds on their computational complexity, the two algorithms are scalable to large networks, and have comparable performance. Figure 7(b) shows, however, that PCAPH is better for values of K less than L/m. In practice, this means that PCAPH will computationally outperform QR since $L \gg K$ and this threshold will practically never be exceeded.

D. Network Kriging in practice

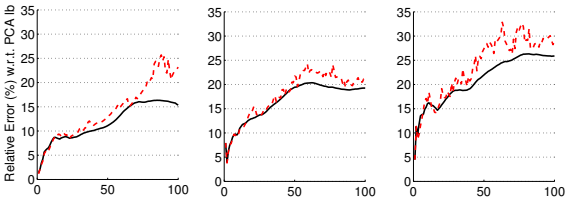
We illustrate next the importance of optimal selection applied in the context of network prediction. We used the real-world data collected from the Internet2 network (see [27], [6]) on March 17, 2009. We utilized the PCAPH algorithm to calculate the optimal set of links to be monitored. The estimation procedure for $\boldsymbol{\mu}_u$ and $\Sigma_{\boldsymbol{y}}$ is discussed in the Supplemental material. Fig. 8(a) depicts the true traffic on the CHIC (Chicago) to WASH (Seattle) link and the predicted traffic when the optimal set of links is used. This set includes links KANS→CHIC, CHIC→KANS, LOSA→HOUS, HOUS→LOSA and SALT→KANS (see Fig. 1). As one would expect, an optimal global view is attained by monitoring links that connect network sites between West and East (see also [22]). Fig. 8(b) shows the empirical relative mean squared error (ReMSE) for the whole network on the day of interest. We compare the quality of prediction when monitoring: a) a non-optimally chosen set, b) the optimal set used throughout the day, and c) the optimal set, periodically recalculated every 8 hours. The last method accounts the newest history available, dynamically re-estimates matrix $\Sigma_{\boldsymbol{y}}$ and calculates the new optimal set. ReMSE



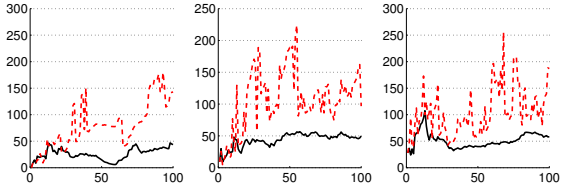
(a) A-optimality ($L = 204, J = 276$)



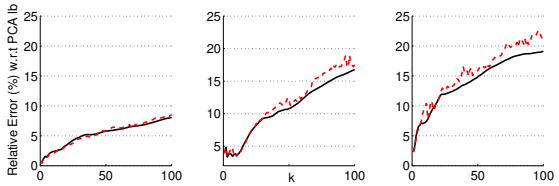
(b) E-optimality ($L = 204, J = 276$)



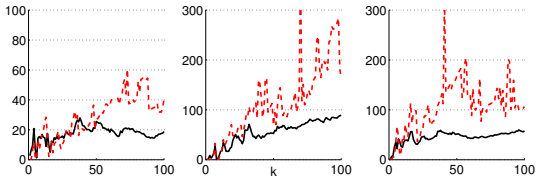
(c) A-optimality ($L = 404, J = 2360$)



(d) E-optimality ($L = 404, J = 2360$)

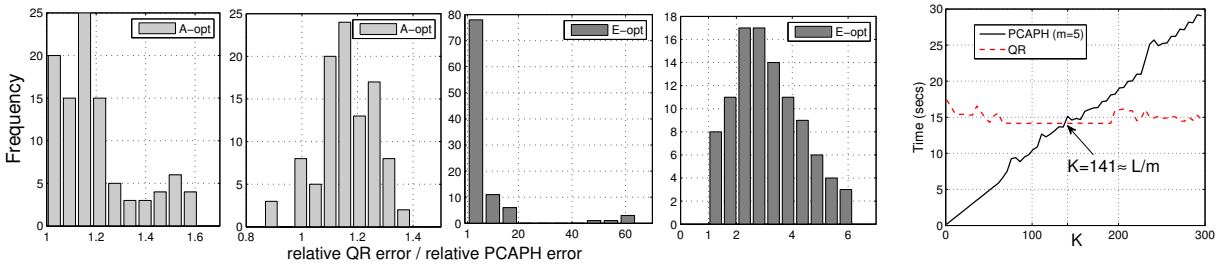


(e) A-optimality ($L = 752, J = 5266$)



(f) E-optimality ($L = 752, J = 5266$)

Fig. 6. PCAPH (solid-black) versus QR (dotted-red): Comparisons on approximation quality. Figures show the prediction error



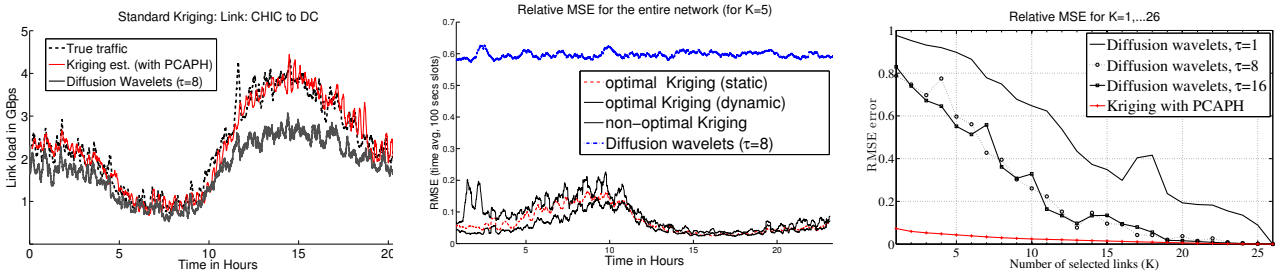
(a) PCAPH ensembling versus QR: Histogram

(b) Time (secs)

Fig. 7. (Left) PCAPH ensembling versus QR or the CAIDA dataset of $L = 404$ links, $J = 2360$ flows. Each histogram shows the ratio *relative QR error/relative PCAPH error* for 100 comparisons in total. (Right) Computational performance of PCAPH versus QR on the CAIDA scenario of $L = 752$, $J = 5266$.

is defined as, $ReMSE(t) = (\sum_{l\in\mathcal{U}}(\widehat{y}_l(t) - y_l(t))^2)/\sum_{l\in\mathcal{U}} y_l(t)^2)$. The ReMSE time average is 0.09, 0.07 and 0.056, respectively. This clearly shows the advantages of employing fast algorithms that can be dynamically used for optimal link monitoring.

Figure 8 also compares kriging-based prediction against diffusion wavelets methods studied in [17]. Using the Internet2 dataset, we tried the algorithms of Coates *et al.* [17] for several values of time diffusion τ. Our kriging estimator exhibits a much lower estimation error than the nonlinear estimator of [17] in terms of the ReMSE criterion. The superiority of kriging is not surprising. As Eq. (2) shows, under the Gaussian assumption the "best" predictor (*i.e.,* the conditional expectation) is a linear function



(a) Link CHIC-DC, with $K = 5$ (b) All links, with $K = 5$ (c) All links, all K

Fig. 8. Link volume estimation on Internet2. (Data for 03/17/09.) Optimal kriging substantially outperforms estimation methods based on diffusion wavelets [17].

of observations. On the other hand, the diffusion wavelet estimator of [17] is nonlinear. Further, the diffusion wavelet method, contrary to kriging, stipulates no knowledge for traffic mean and covariance. As explained in the Supplemental material, acquiring such knowledge is not unrealistic; we obtain $\boldsymbol{\mu}_u$ and $\Sigma_{\boldsymbol{y}}$ through a short "training" period. Alternatively, should this training procedure is not amenable at all links, one can still get accurate *standard kriging* [4], [27] predictions by employing a model for the mean and covariance. As shown in [4], [27], this is plausible via auxiliary information from NetFlow. In such cases, *standard kriging*'s error is expected to lie in-between *ordinary kriging*'s error (Figure 8) and the one attained by the wavelet method.

E. Weighted Link Monitoring

Our methods can be easily adjusted to handle cases where the relative importance of the links is not uniform. Such weighted design is particularly useful when network conditions drastically change in a dynamic manner. Recall that in the trace criterion case we have: $\mathrm{trace}(\mathbb{E}[(\widehat{\boldsymbol{y}} - \boldsymbol{y})(\widehat{\boldsymbol{y}} - \boldsymbol{y})^T]) = \mathbb{E}(\mathrm{trace}[(\widehat{\boldsymbol{y}} - \boldsymbol{y})(\widehat{\boldsymbol{y}} - \boldsymbol{y})^T]) = \mathbb{E}[\|\widehat{\boldsymbol{y}} - \boldsymbol{y}\|^2] = \mathbb{E}[(\widehat{\boldsymbol{y}} - \boldsymbol{y})^T(\widehat{\boldsymbol{y}} - \boldsymbol{y})]$. The weighted monitoring design problem aims to minimize

$$\mathbb{E}[(\widehat{\boldsymbol{y}} - \boldsymbol{y})^T G(\widehat{\boldsymbol{y}} - \boldsymbol{y})] = \sum_{\ell=1}^{L} w_\ell \mathbb{E}(\widehat{y}_\ell - y_\ell)^2, \tag{27}$$

where $G := \mathrm{diag}(w_1, \ldots, w_L)$ is the matrix assigning the link weights. After some algebra we get:

$$\mathbb{E}[(\widehat{\boldsymbol{y}} - \boldsymbol{y})^T G(\widehat{\boldsymbol{y}} - \boldsymbol{y})] = \mathbb{E}[(G^{1/2}\widehat{\boldsymbol{y}} - G^{1/2}\boldsymbol{y})^T(G^{1/2}\widehat{\boldsymbol{y}} - G^{1/2}\boldsymbol{y})] = \mathbb{E}[(\widehat{\boldsymbol{y}}^G - \boldsymbol{y}^G)^T(\widehat{\boldsymbol{y}}^G - \boldsymbol{y}^G)], \tag{28}$$

where $\widehat{\boldsymbol{y}}^G := G^{1/2}\widehat{\boldsymbol{y}}$ and $\boldsymbol{y}^G := G^{1/2}\boldsymbol{y}$. This shows that the trace-optimal solution $\widehat{\boldsymbol{y}}^G$ with the new "routing" matrix $A^G := G^{1/2}A$ optimizes (28). We can readily apply our fast algorithms to obtain $\widehat{\boldsymbol{y}}^G$. The optimal predictor for the *weighted* trace criterion in (27) is then obtained by $\widehat{\boldsymbol{y}} = G^{-1/2}\widehat{\boldsymbol{y}}^G$.

Fig. 3(b) shows the (weighted) prediction error (*i.e.*, a weighted ReMSE) for the scenario where the importance of monitoring links WASH $\rightarrow$ NEWY and WASH $\rightarrow$ ATLA elevates. We can model this by assigning unit weights to all other links, and a weight of ten to the important ones. Based on Eq. (26) and the spectrum shown in Fig. 3(a) we select a budget of $K = 7$. The solution given by PCAPH (which

actually coincides with the exact optimal solution) includes the high importance links, plus links LOSA – HOUS (both directions), KANS – CHIC (both directions) and SALT → KANS. Fig. 3(b) clearly shows that the traffic prediction based on this set of links is way more accurate than the one based on a randomly chosen set (that includes the important links too).

VIII. Discussion

Large-scale optimal monitoring is an important, yet computationally challenging problem, suitable for applications like anomaly detection in communication networks. The aim is to select the "best" links to monitor, so as to *optimally predict* the traffic volume at the remaining ones. Our notion of optimality is quantified in terms of the statistical error of network kriging predictors in the context of global network traffic models. We work with A- and E- optimality for the following reasons. A-optimality has a natural interpretation as the mean squared prediction error (see Eq. (5)). Moreover, A-optimality allows a weighted design, such as the one presented in Section VII-E. E-optimality, captures the "worst case" scenario, since it amounts to minimizing the maximum prediction error among all possible unit-norm weighted linear combinations of links. In particular, the E-optimality objective dominates the worst prediction error among all possible unobserved links. In addition to obtaining appealing intuitive understandings for A- and E-optimality criteria, these criteria were amenable to *fast* implementations, which can be applied to large-scale, dynamically evolving networks in real time. Randomized instances of the algorithms can even yield the optimal solution in a fraction of the time needed to obtain exact solutions using integer programming techniques. Moreover, the novel PCA-based error bounds we propose for A- and E-optimality are practical, network-specific bounds that help us assess the quality of approximation.

The proposed selection algorithms are well suited for kriging applications beyond traffic monitoring, such as end-to-end network delay prediction [5], [17]. Efficient algorithms for alternative criteria such as C-, D-optimality or mutual information, under the context of network kriging, are open research problems.

IX. Acknowledgements

We would like to thank the authors of [17] for kindly providing us their algorithm's source code.

Appendix A
PCA-based Results

Let $\Sigma_y = AA^T$ be the covariance matrix of the vector of all links $\boldsymbol{y}$. SVD of A yields $A = UD^{1/2}V^T$, with $D^{1/2} := \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$, $p = \min\{L, J\}$, where σ_i's are the singular values of A. Thus, $(\lambda_1 \geq \cdots \geq \lambda_p) \equiv (\sigma_1^2 \geq \cdots \geq \sigma_p^2)$ are the eigenvalues of the $J \times J$ matrix $\Omega \equiv A^T A$. Recall that the columns of A^T are denoted by $\boldsymbol{a}_\ell \in \mathbb{R}^J, \ell = 1, \cdots, L$. Then, we have:

Theorem 5 (Trace). *Let P_W denote the projection matrix onto a sub-space $W \subseteq \mathbb{R}^J$. Then,*

$$\min_{W\subseteq\mathbb{R}^J,\ \dim(W)=K} \sum_{\ell=1}^{L} \|\boldsymbol{a}_\ell - P_W \boldsymbol{a}_\ell\|^2 = \sum_{j=K+1}^{p} \lambda_j,$$

where the lower bound is achieved for the sub-space $W^* = \mathrm{span}(\mathbf{v}_1, \cdots, \mathbf{v}_K)$ of the eigenvectors corresponding to the largest K eigenvalues of Ω.

Theorem 6 (Spectral norm). *Let the SVD of matrix A be as above. Then,*

$$\min_{\mathrm{rank}(B)=K} \|A - B\|_2 = \|A - AP_K\|_2 \tag{29}$$

where P_K is the projection matrix $P_K = V_K V_K^T$. The columns of matrix V_K are the top K right singular vectors of A, i.e.[14]*, $P_K = V\mathrm{diag}(\mathbf{1}_K^T, \mathbf{0}_{L-K}^T)V^T$.*

The proof of the former theorem appears in [4] (Proposition 1) and [32] (p. 785, Theorem A.2). For the proof of the latter, see Theorem 2.5.3 in [19]. We use these two theorems to prove Theorems 1 and 2.

Proof of Theorem 1: We will use Theorem 5. Recall that $\Sigma_{\mathrm{err}}(\mathcal{O}) = A(I - P_{\mathcal{O}})A^T$, where $P_{\mathcal{O}} := A_o^T(A_o A_o^T)^{-1}A_o$. The projection matrix $(I - P_{\mathcal{O}})$ is symmetric and indempotent (*i.e.*, $(I - P_{\mathcal{O}}) = (I - P_{\mathcal{O}})^2$), so we have $\Sigma_{\mathrm{err}} = [(I - P_{\mathcal{O}})A^T]^T[(I - P_{\mathcal{O}})A^T]$, and therefore

$$\mathrm{trace}(\Sigma_{\mathrm{err}}(\mathcal{O})) = \sum_{\ell=1}^{L} \|(I - P_{\mathcal{O}})\boldsymbol{a}_\ell\|^2 = \sum_{\ell=1}^{L} \mathrm{dist}(\boldsymbol{a}_\ell, W_{\mathcal{O}})^2 \tag{30}$$

where $W_{\mathcal{O}} := \mathrm{span}(\boldsymbol{a}_\ell,\ \ell \in \mathcal{O})$ is the sub-space spanned by the vectors $\boldsymbol{a}_\ell$ corresponding to the observed links. The vector $(I - P_{\mathcal{O}})\boldsymbol{a}_\ell$ is the "perpendicular" dropped from point $\boldsymbol{a}_\ell$ to the hyperplane $\mathrm{Range}(A_o^T)$. Hence $\|(I - P_{\mathcal{O}})\boldsymbol{a}_\ell\|$ is the distance from $\boldsymbol{a}_\ell$ to $\mathrm{Range}(A_o^T) = \mathrm{span}(\boldsymbol{a}_\ell,\ \ell \in \mathcal{O})$, where $A_o^T = (\boldsymbol{a}_\ell)_{\ell\in\mathcal{O}}$.

Using Theorem 5 we see that the sum of (30) is minimized when the projection matrix $P_{\mathcal{O}}$ equals $P_K = V_K V_K^T$. Hence, $\mathrm{trace}(\Sigma_{\mathrm{err}}(\mathcal{O})) = \mathrm{trace}[A(I - P_{\mathcal{O}})A^T] \geq \|A - AP_K\|_F^2 = \sum_{i=K+1}^{p} \lambda_i$. ■

Proof of Theorem 2: Using Theorem 6, we need to show that: $\rho(\Sigma_{\mathrm{err}}(\mathcal{O})) = \rho[A(I - P_{\mathcal{O}})A^T] \geq \|A - AP_K\|_2^2 = \lambda_{K+1}$. We first calculate the lower bound when $P_K = V_K V_K^T$. We use the SVD of A, $A = UD^{1/2}V^T$ and the projection matrix $I - P_K = V\mathrm{diag}(\mathbf{0}_K^T, \mathbf{1}_{L-K}^T)V^T$. Thus, we have $A - AP_K = UD^{1/2}V^T V\mathrm{diag}(\mathbf{0}_K^T, \mathbf{1}_{L-K}^T)V^T = UD^{1/2}\mathrm{diag}(\mathbf{0}_K^T, \mathbf{1}_{L-K}^T)V^T = U\mathrm{diag}(\mathbf{0}_K^T, \sigma_{K+1}, \ldots, \sigma_L)V^T$. By the definition of spectral norm: $\|A - AP_K\|_2^2 = \rho\left((U\mathrm{diag}(\mathbf{0}_K^T, \sigma_{K+1}, \ldots, \sigma_L)V^T)^T \times U\mathrm{diag}(\mathbf{0}_K^T, \sigma_{K+1}, \ldots, \sigma_L)V^T\right) = \rho\left(V\mathrm{diag}(\mathbf{0}_K^T, \lambda_{K+1}, \ldots, \lambda_L)V^T\right) = \lambda_{K+1}$. Consequently, using also Theorem 6 we obtain: $\rho(\Sigma_{\mathrm{err}}(\mathcal{O})) = \rho[A(I - P_{\mathcal{O}})A^T] = \rho[(A - AP_{\mathcal{O}})(A - AP_{\mathcal{O}})^T] = \|A - AP_{\mathcal{O}}\|_2^2 \geq \|A - AP_K\|_2^2 = \lambda_{K+1}$. ■

[14] We symbolize the vector of ones of dimension k as $\mathbf{1}_k$ and the vector of zeros as $\mathbf{0}_k$.

APPENDIX B
PROPOSITION PROOFS

Proof of Proposition 1: The proof resembles the Gram–Schmidt orthogonalization procedure. We will show by induction that

$$A^{(k)}A^{(k)^T} = A(I - A_o^T(A_oA_o^T)^{-1}A_o)A^T. \tag{31}$$

For notational simplicity, let $\mathcal{O} := \mathcal{O}^k$, i.e., we drop the subscript k of the matrix A_{o_k}. It is easy to see that the sequential updates of the matrices $A^{(k)}$ in (13) can be represented in matrix form as follows:

$$A^{(k)} = AP_1P_2\cdots P_k, \tag{32}$$

where $P_1 = I - \frac{\boldsymbol{a}_{i_1}\boldsymbol{a}_{i_1}^T}{\|\boldsymbol{a}_{i_1}\|^2}$, $\cdots, P_k = I - \frac{\boldsymbol{a}_{i_k}^{(k-1)}\boldsymbol{a}_{i_k}^{(k-1)^T}}{\|\boldsymbol{a}_{i_k}^{(k-1)}\|^2}$. Here $\boldsymbol{a}_i^{(k)} \in \mathbb{R}^J$ denote the rows of the matrix $A^{(k)}$, where by convention $A^{(0)} = A$. Observe that P_k is the orthogonal projection matrix onto the space $(\text{span}\{\boldsymbol{a}_{i_k}^{(k-1)}\})^\perp$, i.e. the orthogonal complement of the one-dimensional space spanned by $\boldsymbol{a}_{i_k}^{(k-1)}$. Here $\boldsymbol{a}_{i_k}^{(k-1)}$ corresponds to the link i_k added to the set $\mathcal{O}$ on step k. This shows that on the $k-$th step *all* the rows of the matrix $A^{(k)}$ are *orthogonal* to $\text{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\} \equiv \text{span}\{\boldsymbol{a}_{i_1}^{(0)}, \cdots, \boldsymbol{a}_{i_k}^{(k-1)}\}$. The last subspace is generated by the vectors corresponding to the links $\{i_1, \cdots, i_k\}$ added on the first k steps. Now, to complete the proof, it is enough to show that

$$P_1P_2\cdots P_k = P_{\text{span}(\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k})^\perp}. \tag{33}$$

Indeed, we have that $P_{(\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k}\})^\perp} = I - A_o^T(A_oA_o^T)^{-1}A_o$, where $A_o^T = (\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k})$. Therefore, by (32) and (33), one obtains (31), which yields (14). We now prove (33) by induction.

Induction Basis: Relation (33) trivially holds for $k = 1$.

Induction Hypothesis: Suppose that (33) holds.

Induction Step: We will show that (33) holds with k replaced by $k+1$.

Note that by the induction hypothesis $\boldsymbol{a}_{i_{k+1}}^{(k)}$ is the orthogonal projection of $\boldsymbol{a}_{i_{k+1}}$ onto $(\text{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\})^\perp$. Therefore, $\text{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}, \boldsymbol{a}_{i_{k+1}}\} = \text{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\} \oplus \text{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}$, where $\oplus$ denotes sum of orthogonal subspaces of $\mathbb{R}^J$. This shows that $P_{\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k},\boldsymbol{a}_{i_{k+1}}\}} = P_{\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k}\}} + P_{\text{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}}$. Since $P_{W^\perp} = I - P_W$, we obtain

$$P_{(\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k},\boldsymbol{a}_{i_{k+1}}\})^\perp} = I - P_{\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k}\}} - P_{\text{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}}. \tag{34}$$

Note, however, that since $\boldsymbol{a}_{i_{k+1}}^{(k)} \perp \text{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\}$, we have $P_{\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k}\}}P_{\text{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}} = 0$, and the right-hand side of (34) equals $(I - P_{\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k}\}})(I - P_{\text{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}}) = P_{\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k}\}^\perp}P_{\text{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}^\perp}$. This, in view of the induction hypothesis and (34), implies that $P_{(\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k},\boldsymbol{a}_{i_{k+1}}\})^\perp} = P_1\cdots P_kP_{k+1}$, which completes the proof of the induction step. ■

Proof of Proposition 2: By definition, $\tilde{\lambda}_1$ is the following random variable:

$$\tilde{\lambda}_1 := \max\{\frac{\boldsymbol{x}_1^T}{\|\boldsymbol{x}_1\|}\Sigma\frac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|}, \ldots, \frac{\boldsymbol{x}_m^T}{\|\boldsymbol{x}_m\|}\Sigma\frac{\boldsymbol{x}_m}{\|\boldsymbol{x}_m\|}\}, \tag{35}$$

with $\boldsymbol{x}_i \in \mathbf{R}^L$, $i = 1, ..., m$ are independent random vectors from the multivariate normal distribution $\mathcal{N}(0, I_L)$. Let the SVD of Σ be $\Sigma = UDU^T$, with the columns of U being the singular vectors of Σ and $D = \mathrm{diag}(\lambda_1, ..., \lambda_{\mathrm{L}})$. We multiply each vector $\boldsymbol{x}_i$ with the orthogonal matrix P, with $P = U^T$ to get $\boldsymbol{w}_i = P\boldsymbol{x}_i$. Since this operation preserves inner products, angles and distances, the vectors $\boldsymbol{w}_i$ are also *iid* from the $\mathcal{N}(0, I_L)$ distribution. Hence, $\tilde{\lambda}_1 = \max_i\{\frac{\boldsymbol{x}_i^T}{\|\boldsymbol{x}_i\|}UDU^T\frac{\boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}\} = \max_i\{\frac{\boldsymbol{w}_i^T}{\|\boldsymbol{w}_i\|}D\frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}\} = \max_i\{\frac{\sum_{j=1}^L \lambda_j w_i^2(j)}{\|\boldsymbol{w}_i\|^2}\} = \lambda_1 \max_i\{\frac{\sum_{j=1}^L c_j w_i^2(j)}{\|\boldsymbol{w}_i\|^2}\}$.

Given $\varepsilon > 0$, the precision of approximation is: $\mathbf{P}(\tilde{\lambda}_1 > \lambda_1(1-\varepsilon)) = \mathbf{P}(\max_i \xi_i > 1-\varepsilon)$, with the random variables $\xi_i = \sum_{j=1}^L c_j w_i^2(j)/\|\boldsymbol{w}_i\|^2$, $i = 1, \ldots, m$ being independent and identically distributed. Proceeding, we have $\mathbf{P}(\max_i \xi_i > 1-\varepsilon) = 1 - \mathbf{P}(\max_i \xi_i \leq 1-\varepsilon) = 1 - [\mathbf{P}(\xi_1 \leq 1-\varepsilon)]^m$, with ξ_1 chosen without loss of generality. Explicitly writing ξ_1 we obtain, $\mathbf{P}(\xi_1 \leq 1-\varepsilon) = \mathbf{P}\Big(\sum_{j=1}^L c_j w_1^2(j) \leq (1-\varepsilon)\sum_{j=1}^L w_1^2(j)\Big) = \mathbf{P}\Big(\sum_{j=1}^t [c_j - (1-\varepsilon)]w_1^2(j) \leq \sum_{j=t+1}^L [(1-\varepsilon) - c_j]w_1^2(j)\Big) \leq \mathbf{P}\Big(\sum_{j=1}^t [c_j - (1-\varepsilon)]w_1^2(j) \leq \sum_{j=t+1}^L (1-\varepsilon)w_1^2(j)\Big) \leq \mathbf{P}\Big([c_t - (1-\varepsilon)]\sum_{j=1}^t w_1^2(j) \leq (1-\varepsilon)\sum_{j=t+1}^L w_1^2(j)\Big)$, where we used the fact that $c_i > 1-\varepsilon$ for all $i \leq t$, and $c_i \leq 1-\varepsilon$ otherwise, to obtain the last two inequalities.

The random variables $\sum_{j=1}^t w_1^2(j)$ have *chi-square* density with t-degrees of freedom (see [33], Section II.3). Similarly, he random variables $\sum_{j=t+1}^L w_1^2(j)$ have *chi-square* density with $(L-t)$-degrees of freedom. Using the fact (see [33], Section II.3) that the random variable $F = (X/\nu_1)/(Y/\nu_2)$ – with X, Y being *chi-squared* distributed with ν_1 and ν_2 degrees of freedom respectively – has the F-density with parameters ν_1 and ν_2, the result follows.

■

APPENDIX C
ERROR REDUCTION BOUNDS

Proof of Theorem 3: For ease of notation, we drop the superscript k. For the same reason, we introduce the operator $\langle \boldsymbol{e}_1, \boldsymbol{e}_2\rangle$ to represent the inner product between two vectors. Let $x_j := |\langle \boldsymbol{a}_j, \boldsymbol{a}_i\rangle|/\|\boldsymbol{a}_i\|$ be the length of the projection of any vector $\boldsymbol{a}_j$ to the selected vector $\boldsymbol{a}_i$. Also, let $y_j := |\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)}\rangle|$ be the projection length on $\mathbf{v}_1^{(k)}$. Note that $\boldsymbol{a}_i/\|\boldsymbol{a}_i\| = \langle \boldsymbol{a}_i/\|\boldsymbol{a}_i\|, \mathbf{v}_1^{(k)}\rangle\mathbf{v}_1^{(k)} + \boldsymbol{a}_i^\perp = \gamma\mathbf{v}_1^{(k)} + \boldsymbol{a}_i^\perp$. Proceeding, we have $x_j^2 = |\langle \boldsymbol{a}_j, \gamma\mathbf{v}_1^{(k)} + \boldsymbol{a}_i^\perp\rangle|^2 = |\gamma\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)}\rangle + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^\perp\rangle|^2 \geq |\gamma|\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)}\rangle| - |\langle \boldsymbol{a}_j, \boldsymbol{a}_i^\perp\rangle||^2 \geq \gamma^2\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)}\rangle^2 - 2\gamma|\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)}\rangle||\langle \boldsymbol{a}_j, \boldsymbol{a}_i^\perp\rangle| = \gamma^2 y_j^2 - 2\gamma y_j|\langle \boldsymbol{a}_j, \boldsymbol{a}_i^\perp\rangle|$, after substitution of the projection length y_j on the first principal axis. Observe that due to orthogonality, we have $\gamma^2 + \|\boldsymbol{a}_i^\perp\|^2 = 1$. Using the Cauchy – Bunyakovsky – Schwarz inequality we then bound the term $|\langle \boldsymbol{a}_j, \boldsymbol{a}_i^\perp\rangle|$ as $|\langle \boldsymbol{a}_j, \boldsymbol{a}_i^\perp\rangle| \leq \|\boldsymbol{a}_j\|\|\boldsymbol{a}_i^\perp\| = \|\boldsymbol{a}_j\|\sqrt{1-\gamma^2}$. Therefore, $x_j^2 \geq \gamma^2 y_j^2 - 2\gamma y_j\|\boldsymbol{a}_j\|\sqrt{1-\gamma^2}$. Now we sum over all $j \in \mathcal{L}$ to

get the error reduction for all links. We thus get,

$$\sum_{j=1}^{L} x_j^2 \geq \gamma^2 \sum_{j=1}^{L} y_j^2 - 2\gamma\sqrt{1-\gamma^2}\sum_{j=1}^{L} y_j\|\boldsymbol{a}_j\| = \gamma^2\lambda_1^{(k)} - 2\gamma\sqrt{1-\gamma^2}\sum_{j=1}^{L} y_j\|\boldsymbol{a}_j\|, \tag{36}$$

where in the last equation we used Theorem 1 for $K = 1$ on matrix $A^{(k)}$; thus, the PCA error reduction equals the largest eigenvalue, $\lambda_1^{(k)}$. We now isolate the term $\sum_{j=1}^{L} y_j\|\boldsymbol{a}_j\|$ and using again the Cauchy – Bunyakovsky – Schwarz bound we get $\sum_{j=1}^{L} y_j\|\boldsymbol{a}_j\| \leq \sqrt{\sum_{j=1}^{L}(y_j)^2}\sqrt{\sum_{j=1}^{L}\|\boldsymbol{a}_j\|^2} = \sqrt{\lambda_1^{(k)}}\sqrt{\text{trace}(A^{(k)^T}A^{(k)})} = \sqrt{\lambda_1^{(k)}}\sqrt{\sum_{j=1}^{L}\lambda_j^{(k)}}$. The result follows. ■

Proof of Theorem 4: Again, when there is no ambiguity we drop the superscript k. Let $x_j := |\langle \boldsymbol{a}_j, \boldsymbol{a}_i\rangle|/\|\boldsymbol{a}_i\|$ be the length of the projection of any vector $\boldsymbol{a}_j$ to the selected by the algorithm vector $\boldsymbol{a}_i$. Also, let $y_j := |\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)}\rangle|$ be the projection length on $\mathbf{v}_1^{(k)}$.

As before $\boldsymbol{a}_i/\|\boldsymbol{a}_i\| = \gamma\mathbf{v}_1^{(k)} + \boldsymbol{a}_i^{\perp}$. Proceeding, we have $x_j^2 = |\langle \boldsymbol{a}_j, \gamma\mathbf{v}_1^{(k)} + \boldsymbol{a}_i^{\perp}\rangle|^2 = |\gamma\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)}\rangle + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp}\rangle|^2 = \gamma^2\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)}\rangle^2 + 2\gamma\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)}\rangle\langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp}\rangle + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp}\rangle^2 = \gamma^2 y_j^2 + 2\gamma y_j \underbrace{\langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp}\rangle} + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp}\rangle^2 \leq \gamma^2 y_j^2 + 2\gamma \underbrace{y_j\|\boldsymbol{a}_j\|}\sqrt{1-\gamma^2} + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp}\rangle^2 \leq \gamma^2 y_j^2 + 2\gamma\sqrt{\lambda_1^{(k)}}\sqrt{\sum_{j=1}^{L}\lambda_j^{(k)}}\sqrt{1-\gamma^2} + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp}\rangle^2$, with the Cauchy – Bunyakovsky – Schwarz inequality used on the specified terms. Note that for $\phi > 0$, $\boldsymbol{a}_j = \phi\mathbf{v}_1^{(k)} + \boldsymbol{a}_j^{\perp}$, i.e. $\boldsymbol{a}_j^{\perp} \perp \mathbf{v}_1^{(k)}$. Then, $\langle \phi\mathbf{v}_1^{(k)} + \boldsymbol{a}_j^{\perp}, \boldsymbol{a}_i^{\perp}\rangle = \langle \boldsymbol{a}_j^{\perp}, \boldsymbol{a}_i^{\perp}\rangle$ because $\boldsymbol{a}_i^{\perp} \perp \mathbf{v}_1^{(k)}$ as well. Moreover, notice that $|\langle \boldsymbol{a}_j^{\perp}, \boldsymbol{a}_i^{\perp}\rangle| \leq |\langle \boldsymbol{a}_j^{\perp}, \boldsymbol{a}_i^{\perp}\rangle|/\|\boldsymbol{a}_i^{\perp}\|$ because $\|\boldsymbol{a}_i^{\perp}\| = \sqrt{1-\gamma^2} \leq 1$ (by construction of $\|\boldsymbol{a}_i^{\perp}\|$). These steps suggest that:

$$\sum_{j=1}^{L}\langle \boldsymbol{a}_j^{\perp}, \frac{\boldsymbol{a}_i^{\perp}}{\|\boldsymbol{a}_i^{\perp}\|}\rangle^2 \leq \sum_{j=1}^{L}\langle \boldsymbol{a}_j^{\perp}, \mathbf{v}_2^{(k)}\rangle^2 = \lambda_2^{(k)}. \tag{37}$$

This result follows from PCA. It basically says that the sum of the squares of the projection lengths of any vector $\boldsymbol{a}_j^{\perp}$ on $\boldsymbol{a}_i^{\perp}$ is bounded by the sum of the squares of the projections on the second principal axis, $\mathbf{v}_2^{(k)}$. In other words, $\mathbf{v}_2^{(k)}$ gives the maximum projection with respect to any other vector from $\text{Null}(\mathbf{v}_1^{(k)^T})$. Summing both sides of inequality $x_j^2 \leq \gamma^2 y_j^2 + 2\gamma\sqrt{\lambda_1^{(k)}}\sqrt{\sum_{j=1}^{L}\lambda_j^{(k)}}\sqrt{1-\gamma^2} + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp}\rangle^2$ over all $j = 1, ..., L$, and using (37), concludes the proof.

■

REFERENCES

[1] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 219–230, August 2004.

[2] I. C. Paschalidis and G. Smaragdakis, "Spatio-temporal network anomaly detection by assessing deviations of empirical measures," *IEEE/ACM Trans. Netw.*, vol. 17, pp. 685–697, June 2009.

[3] Cisco Systems, "Cisco IOS Netflow," www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html.

[4] S. Vaughan, J. and Stoev and G. Michailidis, "Network-wide statistical modeling and prediction of computer traffic," 2010, to appear in Technometrics. [Online]. Available: http://arxiv.org/pdf/1005.4641.pdf

[5] D. Chua, E. Kolaczyk, and M. Crovella, "Network kriging," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 12, pp. 2263 –2272, dec. 2006.

[6] Internet2, "*Internet2*." [Online]. Available: http://www.internet2.edu/observatory/

[7] C.-W. Ko, J. Lee, and M. Queyranne, "An Exact Algorithm for Maximum Entropy Sampling," *OPERATIONS RESEARCH*, vol. 43, no. 4, pp. 684–691, Jul. 1995. [Online]. Available: http://dx.doi.org/10.1287/opre.43.4.684

[8] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, pp. 235–284, June 2008.

[9] H. Singhal and G. Michailidis, "Optimal experiment design in a filtering context with application to sampled network data," *Ann. Appl. Stat.*, vol. 4, no. 1, pp. 78–93, 2010.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[11] A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss, "Approximation of functions over redundant dictionaries using coherence," in *Proc. of 14th annual ACM-SIAM symp. on Discrete algorithms*, ser. SODA '03, 2003, pp. 243–252.

[12] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *Proceedings of the 40th annual ACM symposium on Theory of computing*, 2008, pp. 45–54.

[13] C. Boutsidis, M. W. Mahoney, and P. Drineas, "Unsupervised feature selection for principal components analysis," in *Proceeding of the 14th ACM SIGKDD*, 2008, pp. 61–69.

[14] ——, "An improved approximation algorithm for the column subset selection problem," in *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '09, Philadelphia, PA, USA, 2009, pp. 968–977.

[15] CPLEX, "Ibm ilog cplex optimizer," http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[16] G. Nemhauser and L. Wolsey, "Maximizing submodular set functions: Formulations and analysis of algorithms," in *Annals of Discrete Mathematics*, 1981, vol. 59, pp. 279 – 301.

[17] M. Coates, Y. Pointurier, and M. Rabbat, "Compressed network monitoring for IP and all-optical networks," in *Proc. 7th ACM SIGCOMM*. ACM, 2007, pp. 241–252. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298340

[18] CAIDA, "CAIDA's Internet Topology Data Kit #0304. Cooperative Association for Internet Data Analysis, University of California, San Diego (UCSD), 2003," http://www.caida.org/tools/measurement/skitter/router_topology/.

[19] G. H. Golub and C. F. van Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, Oct. 1996.

[20] G. Sagnol, S. Gaubert, and M. Bouhtou, "Optimal monitoring in large networks by successive c-optimal designs," in *Teletraffic Congress (ITC), 2010 22nd International*, sept. 2010, pp. 1 –8.

[21] G. Sagnol, "Computing optimal designs of multiresponse experiments reduces to second-order cone programming," *Journal of Statistical Planning and Inference*, vol. 141, no. 5, pp. 1684 – 1708, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378375810005318

[22] H. Singhal and G. Michailidis, "Optimal sampling in state space models with applications to network monitoring," in *Proceedings of the 2008 ACM SIGMETRICS*, 2008, pp. 145–156.

[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Mathematical Programming*, vol. 14, pp. 265–294, 1978.

[24] D. Bajovic, B. Sinopoli, and J. Xavier, "Sensor selection for event detection in wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 59, no. 10, pp. 4938 –4953, oct. 2011.

[25] E. Tsakonas, J. Jalden, and B. Ottersten, "Semidefinite relaxations of robust binary least squares under ellipsoidal uncertainty sets," *Signal Processing, IEEE Transactions on*, vol. 59, no. 11, pp. 5169 –5180, nov. 2011.

[26] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation: part i: Greedy pursuit," *Signal Process.*, vol. 86, pp. 572–588, March 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1140723.1140735

[27] S. Stoev, G. Michailidis, and J. Vaughan, "On global modeling of backbone network traffic," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –5.

[28] K. Park and W. Willinger, Eds., *Self-Similar Network Traffic and Performance Evaluation*. J. Wiley & Sons, Inc., 2000.

[29] I. Jolliffe, *Principal Component Analysis (2nd ed.)*. New York: Springer-Verlag, 2002.

[30] A. Barabasi and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[31] H. Singhal and G. Michailidis, "Identifiability of flow distributions from link measurements with applications to computer networks," *Inverse Problems*, vol. 23, no. 5, p. 1821, 2007.

[32] A. Marshall, I. Olkin, and B. Arnold, *Inequalities: Theory of Majorization and Its Applications (2nd ed)*. Springer, 2010.

[33] W. Feller, *An Introduction to Probability Theory and its Applications (2nd ed.)*. New York: John Wiley and Sons, 1971.

PLACE PHOTO HERE

Michael G. Kallitsis received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 2005 and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh in 2010. He was a Postdoctoral Fellow in the Department of Statistics at University of Michigan, Ann Arbor from 2010 to 2012.

Currently, Dr Kallitsis is a member of the research group within Merit Network Inc., Ann Arbor. His research interests include algorithms for monitoring and analysis of large-scale data, queueing theory and performance evaluation of communication networks, stochastic optimization and control of communication systems, stochastic modeling of systems, distributed algorithms, mathematical economics and game theory.

PLACE PHOTO HERE

Stilian A. Stoev received his masters degree in mathematics from the Sofia University St Kliment Ohridski, Bulgaria in 1998 under the supervision of Prof Dimitar Vandev. He received his PhD degree in mathematics and statistics from Boston University in 2005 under the supervision of Prof Murad S. Taqqu. In the same year, Dr Stoev joined the statistics department at the University of Michigan, Ann Arbor where he has been working since and currently holds the position of an Associate Professor.

Dr Stoev is working in applied probability and statistics with emphasis on stochastic processes with dependence and heavy tails. His current interests include multivariate extreme value theory, sum and max-stable processes, prediction of extremes, as well as global modeling, analysis, and prediction for computer network traffic.

PLACE PHOTO HERE

George Michailidis received the Ph.D. degree in mathematics from the University of California, Los Angeles, in 1996.

He was a Postdoctoral Fellow in the Department of Operations Research at Stanford University from 1996 to 1998. He joined the University of Michigan, Ann Arbor, in 1998, where he is currently a Professor of Statistics, Electrical Engineering, and Computer Science. His research interests are in the areas of stochastic network modeling and performance evaluation, queuing analysis and congestion control, statistical modeling and analysis of Internet traffic, network tomography, and analysis of high dimensional data with network structure.