Inference of Link Delay in Communication Networks

Ye Xia and David Tse

Abstract—**This paper studies the feasibility and algorithms for inferring the delay at each link in a communication network based on a large number of end-to-end measurements. The restriction is that we are not allowed to measure directly on each link and can only observe the route delays. It is assumed that we have considerable flexibility in choosing which routes to measure. We investigate two different cases: 1) each link delay is a constant and 2) each link delay is modeled as a random variable from a family of distributions with unknown parameters. We will answer whether such indirect inference is possible at all, and when possible, how it can be carried out. The emphasis is on developing the maximum-likelihood estimators for scenario 2) when the link delays are modeled by exponential random variables or mixtures of exponentials. We have derived solutions based on the EM algorithm and demonstrated that, even though they do not necessarily reflect the true model parameters, they do seem to maximize the likelihood in most cases and that the resulting probability density functions match the true functions on regions where the probability mass concentrates.**

Index Terms—**Expectation maximization (EM) algorithm, maximum-likelihood estimator, network delay measurement, network tomography.**

I. Introduction

A. Motivation

As THE Internet grows, it becomes increasingly important to monitor the network performance and identify failures. These must be based on the knowledge of the complete or partial topology of the network and on the measurement of some quantities of the network. In this paper, we study the issue of network measurement, given that the network topology is completely known. The useful information that one might want to know about the network includes packet delay, packet loss ratio, link capacity and throughput, etc. One might be interested in the end-to-end information on some route or localized information at some routers or links. For information with randomness, it is also possible to distinguish its longtime average and instantaneous value. In building a network measurement infrastructure, there is a choice about the measurement locations, where the measurement software or hardware are placed. We can think of placing them: 1) in every network router and end-hosts; 2) only in the end-hosts at the edge of the network; or 3) in selected routers and end-hosts. The constraints are, first, it might never be possible to have a consistent measurement infrastructure throughout the entire Internet, which is partitioned into a disparate array of administrative domains, and second, it is costly and practically difficult to install, maintain, and upgrade the measurement related software and hardware in a large number of routers and computers.

This paper addresses the choice of measurement locations, in particular, the feasibility of measuring every link of the entire network, perhaps indirectly, from a subset of the network nodes. The object of measurement, which we call the link attribute, is additive in the sense that the combined attribute on multiple links is the sum of the individual link attributes. The instantaneous delay, the average delay, and the number of packet loss are all additive. Loss ratio and throughput are examples of nonadditive link attributes. We will consider delay as the prototype of an additive link attribute.

We will specifically investigate a special case of choice 2) above. The measurement agents are at selected end-hosts at the edge of the network. The following is a typical question we would like to answer. Suppose we do not have a direct measurement of the average delay on a particular link, but we do have the end-to-end delay measurement on many routes that pass through that link. Can we, then, calculate the link delay based on the end-to-end measurement?

At the abstract level, the problem addressed in this paper is to obtain the vector x in $Rx = y$ given that the vector y and the matrix R are known. Here, R is the routing matrix whose (i, j)th entry is an integer representing the number of times link j appears in route i, y is the vector of route delays, and x is the vector of link delays. There are two versions of the problem: x is deterministic or x is a random vector. In the latter version, we are particularly interested in the case where the distribution of x is from a known parametric family, such as the exponential family or Gaussian family. This is known as the parametric model. In this version, we assume the components of x are statistically independent.

B. Previous Research and Our Contribution

Our work is inspired by [1], which studies how to infer packet loss ratios on individual links in a multicast tree based on the observed loss statistics on end-to-end routes. Our work is independently done from a few other works ([2] and [3]) on the same subject of inferring link delays based on end-to-end delays. As far as we know, the deterministic analysis of Section II has no counterpart in other works.

Nonparametric models are considered in [2] and [3], where each link delay is a random variable but its distribution is not from any parametric family. In [2], the variances of the link delays are estimated using a moment method. Similar methods are also discussed briefly in Section III-B2 of this paper. It is interesting to note that [2] does not give a moment method for estimating the mean of the link delays. As a corollary of our results on the deterministic case, we show that, in the nonparametric case, it is generally not possible to estimate the mean

Manuscript received October 1, 2005; revised April 22, 2006.

Y. Xia is with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611-6120 USA (e-mail: yxl@cise.ufl.edu).

D. Tse is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, Berkeley, CA 94720-1770 USA (e-mail: dtse@eecs.berkeley.edu).

Digital Object Identifier 10.1109/JSAC.2006.884022

of the link delay based on end-to-end measurement using moment-type methods. In [3], the objective is to construct the link delay distributions (histograms) based on end-to-end measurement, under the assumption that link delay has a positive probability of taking the value 0. The problem and techniques involved are fairly different from this paper.

In the study of parametric delay models in Section III, our main effort is on applying the expectation maximization (EM) algorithm to infer the parameters of the link delay distributions. The authoritative reference on the general EM algorithm is [4]. In [5], Vardi applies the EM algorithm in a similar network setting but his objective is to estimate the traffic matrix. Such a problem is to estimate the traffic rates of all end-to-end connections based on the knowledge of routing and the observed traffic rates on the links at the routers. At the abstract level, the problem is to estimate y in the equation $R'y = x$ given R and x, where R' is the transpose of R. Notice that the observed and unobserved data are switched, compared with the link delay inference problem.

From the statistical inference perspective, link delay inference and traffic matrix estimation are similar problems. In [5], Vardi uses a Poisson model for the unobserved variables and we consider exponential and mixture of exponential models. In [6], the unobserved variables are Gaussians. This does not contradict one of our results in the paper that states the Gaussian model is not identifiable, because our result, in fact, says the Gaussian mean is not identifiable. In [6], the mean of each Gaussian variable depends on its variance, which can be identified using, for instance, the method of moment (see Section III-B2). In [7], Castro *et al.* develop a pseudo-likelihood approach together with a pseudo-EM algorithm, with the objective of reducing computational complexity. Such an approach is also applied to the link delay inference problem under the same nonparametric model used in [3]. Some other related studies in traffic matrix estimation can be found in [8]–[11].

Our main contributions are as follows. In the deterministic analysis in Section II, we show that, under fairly general condition, the internal deterministic link attributes cannot be identified based on end-to-end measurement from the edge of the network, regardless how much power the observer has in setting up redundant measurement routes. In Section III, we consider parametric probabilistic models. Based on the result in the deterministic case, we show that, if the link attributes are independently and identically distributed (iid) Gaussian random variables, the Gaussian mean cannot be identified, and hence, the Gaussian model is not identifiable.

We next consider two other parametric models, the exponential model (Section III-B) and the mixture of exponential model (Section III-C), which constitute the main part of the paper. Maximum-likelihood estimators for these models pose both analytical and computational challenges. In both models, we give detailed solutions based on the EM algorithm. Our implementation of the EM algorithm avoids both numerical integration and exponential dependence on the number of model parameters. In the case of the exponential model, we also give several methods of moment, and suggest some other techniques, such as *Successive EM Algorithm.*

We stress that, with the parametric model, the intention is not necessary to discover the true model parameters. We show with numerical experiments (Section IV) that the strength of the EM algorithm is not in finding the true model parameters, but in excellent curve-fitting to the probability density functions (pdf) of both the observed and unobserved random variables. This curve-fitting ability could, in practice, be more relevant to our problem, where the link attributes may not come from a well-specified probability family or cannot be represented parametrically. Under this circumstance, we can use the mixture model and/or the sum of exponential model, both of which have been suggested for modeling the link delays, to approximate the underlying true distribution function, and use the EM algorithm to find good fit.

II. Deterministic Case

In this section, we consider link attributes as constant quantities, and hence, are subject to deterministic analysis. We will study the possibility of determining the link attributes through end-to-end measurement. The main results are Proposition 2.2 and Corollary 2.3.

The deterministic analysis is motivated by the consideration of nonparametric probability model of link delays. We give two motivating examples here. In the first example, we would like to determine the empirical distribution of each link delay through end-to-end measurement. For each measured end-to-end delay sample, we normally need to determine the delay due to each link on that end-to-end route. In a slightly different example, suppose we have the average delay on an end-to-end route and would like to find the average delay of each link on the route. In the following, we will state the problem more formally.

A. Some Standard Graph Theory Definitions

A directed graph $G = [A, U]$ is defined to be a set A whose elements are called *nodes* (or *vertices*) and a set U whose elements are called *edges*. Each edge is directed and is represented by an ordered pair of vertices.

Suppose G is a directed graph. A *path* of length q is a sequence of q edges, $P = (u_1, \ldots, u_q)$, where $u_1, \ldots, u_q \in U$, and $u_1 = (a_0, a_1)$, $u_2 = (a_1, a_2), \ldots, u_q = (a_{q-1}, a_q)$. A *simple path* is a path in which no node appears more than once. A *circuit* (also known as simple cycle) is a closed path whose initial and final nodes coincide, and in which no other nodes are visited more than once. The *distance* from node a_1 to node a_2 is the length of the shortest path from a_1 to a_2.

For each $a \in A$, define $d_G^+(a)$ to be the number of edges starting from$a$, and$d_G^-(a)$ to be the number of edges ending at$a$.

An undirected graph, or simply a *graph* G is a pair $[A, U]$, where A is the set of nodes and U is a set of *edges*. Each edge is undirected and is represented by an unordered pair of nodes.

B. Statement of the Problem and Key Results

We represent a network as a directed graph $G = [A, U]$, where A is the set of routers and end-hosts, and U is the set of communication links between elements of A. Note that a link in this paper is considered directed, represented by a directed edge in the directed graph. Hence, a physical link in the network that allows data transmission in both directions is considered two links in this paper, and is represented by two edges going in opposite directions in the directed graph representation. The reason is that we allow the attributes, such as delays, on the two directions of the physical link to be different. Let $A_s \subseteq A$ be the set of end-hosts that are designated to send packets, called *senders*, and let $A_r \subseteq A$ be the set of end-hosts that are designated to receive packets, called *receivers*. We call the elements in A_s *sender nodes*, the elements in A_r *receiver nodes*, and, together, they are called *measurement nodes*. Any node a with $d_G^+(a) \geq 1$ and $d_G^-(a) \geq 1$ is called a *router node*. Because a router node has at least one incoming link and at least one outgoing link, it is capable of forwarding packets from an incoming link to an outgoing link. Let us associate with each link, $u \in U$, a constant x_u, which represents a fixed attribute of the link u, such as the expected delay. Assuming the attributes are additive when multiple links are involved, our objective is to study the feasibility of determining each x_u, for $u \in U$, when the accumulated attributes from the senders to the receivers on all possible paths are given. An example is that we want to know whether the expected delay on each link can be recovered simply by sending packets at the senders and observing the received packets at the receivers, taking the difference between the packet sending and receiving time instances.

Definition 2.1:: A *route* from $s \in A_s$ to $r \in A_r$ is a path in the directed graph $G = [A, U]$ in which any circuit appears at most once. The *route attribute* is the accumulated attributes of the links on the route. (If a link appears on a route n times, its link attribute is accumulated n times.)

Lemma 2.1: Given a finite directed graph $G = [A, U]$, the sets A_s and A_r, the total number of routes is finite.

Proof: Let N be the number of nodes in G. A circuit can traverse at most N distinct nodes. The number of circuits that each traverse $K \leq N$ distinct nodes is bounded by $N!/(N-K)!$. Therefore, the total number of distinct circuits is finite. The total number of simple paths of length $K < N$ is bounded by $N!/(N-K-1)!$. So, the total number of distinct simple paths is also finite. Notice each route consists a finite number of nodes. At each node, the possible number of distinct circuits is also finite. Hence, the total number of routes must be finite. ■

Although a path in which some circuits appear more than once is not considered as a valid route, the definition of the route is not restrictive for the problem we are considering. This is because the accumulation of the link attributes around a circuit, called the circuit attribute, can be observed. The accumulated attributes on two paths that differ only in the number of times a circuit is traversed differ by a known constant. Note that allowing circuits in the definition of the route potentially leads to much greater number of possible routes on which end-to-end measurement can be made. The main results in this section show that we are not made more powerful in our ability to infer internal link attributes based on end-to-end measurement.

Suppose the links in the directed graph G are labeled as $1, 2, \ldots, L$, where L is the total number of links. Suppose the routes are also indexed from 1 to M, where M is the number of routes. Let the vector $x = (x_1, x_2, \ldots, x_L)'$ in $\mathbb{R}^L$ represent all link attributes, and let $y = (y_1, y_2, \ldots, y_M)'$ in $\mathbb{R}^M$ represent all route attributes, and let $r(i,j)$ be the number of times link j appears on route i. Let $U_i \subseteq U$ be the set of links on route i. Then

$$y_i = \sum_{j \in U_i} r(i,j) x_j. \tag{1}$$

Let R be the $M \times L$ matrix whose (i,j)th entry is $r(i,j)$, called the *route matrix* for G. We can write (1) in vector format

$$Rx = y. \tag{2}$$

Definition 2.2: The directed graph G is said to be *identifiable* if (2) has an unique solution; otherwise, we say G is *unidentifiable*. G is said to be *strongly unidentifiable* if no component of x is uniquely determined by (2).

We now state the main results of the section but postpone the proof of Proposition 2.2 to Section II-D.

Proposition 2.2: Suppose, for any $a_1 \in A_s \cup A_r$ and $a_2 \in A_s \cup A_r$, where $a_1 \neq a_2$, the distance from a_1 to a_2 is at least 2. Then, G is strongly unidentifiable.

The condition for Proposition 2.2 says that no two measurement nodes are adjacent, i.e., are connected by a link. The focus of the deterministic analysis is to prove Proposition 2.2 and to point out its consequences. One necessary condition for solving the set of linear equations in (2) uniquely is to have $M \geq L$, i.e., the number of routes is no less than the number of links. We therefore assume this to be true. We point out here in advance that, in the special case of a symmetric network, the conclusion is different: A symmetric network is identifiable. This will be elaborated in Section II-C.

Corollary 2.3: If at least one link in G does not directly connect two measurement nodes, then G is unidentifiable.

Proof: Remove all links which directly connect any pair of measurement nodes. Then, delete nodes which have no links attached. Links that do not directly connect two measurement nodes, together with the nodes to which they are attached, survive the deletion process. The resulting directed graph has at least one connected component, and in every connected component, all measurement nodes are separated by a minimum distance of 2. The resulting graph is, therefore, strongly unidentifiable. ■

The proof of Corollary 2.3 also shows that, because every link that does not directly connect two measurement nodes will survive the deletion process, we cannot identify its attribute based on end-to-end measurement. Suppose each measurement node is a measuring site, where the measurement software or hardware is placed. By the above comment, the only way to uniquely determine the attribute of a particular link is to measure it at the two nodes directly attached to it. All nodes are required to become measurement sites in order to monitor the entire network. It is not possible to monitor the network from the edge relying on the redundancy of routes.

The unidentifiability result should not be confused with the so-called ill-posed inverse problem in traffic matrix estimation, that is, to find the solution for the vector y in $x = R'y$ (see [11]). In that case, the vector y normally has much higher dimension than x, and the solution is not unique. In the current problem of finding x from $y = Rx$, the matrix R can have much more rows than columns, giving hope that the columns are linearly independent, and hence, the solution is unique. Our result shows that this isn't so except for trivial situations.

We next make the some assumptions about the directed graph G to be considered, which do not reduce the generality of our results but simplify the exposition.

Definition 2.3: A node or a link in G is *reachable* if it is on at least one route.

Assumption 2.1: 1) The directed graph G is connected when viewed as an undirected graph. 2) Every link is reachable.

In the case where G is not connected as a graph, we can look at each connected component separately. When a link is not reachable, it certainly cannot be identified, and it will not affect the identifiability of other links. Hence, we can remove it from G. A consequence of the assumptions is that every node is reachable. For, if a node is not reachable, none of the links connected to it is reachable. Also, every node which is not a measurement node must be a router node. Otherwise, all the links connected to it are not on any routes.

C. Symmetric Networks

Certain special networks are easily identifiable. We say a network is symmetric if its directed graph representation, $G = [A, U]$, has the property that, for any two nodes $a \in A$ and $b \in A$, if (a, b) is an edge in U, then (b, a) is also an edge in U, and moreover, the attribute of (a, b) is identical to that of (b, a). In other words, between adjacent nodes a and b, there is exactly one link from a to b and one from b to a, and they have the same attributes. Under Assumption 2.1, we have the following.

Proposition 2.4: If two nodes a and b are connected by a link from a to b and also by a link from b to a, and if the two links have identical attributes, then either link attribute can be uniquely determined.

Proof: The attribute of any circuit can be uniquely identified because of the following. Given a route with the circuit, remove the circuit and we get another route. The attribute of the circuit is the difference of the attributes of the two routes. The attribute of the link from a to b is half of the attribute of the circuit (a, b, a). ■

Corollary 2.5: Suppose the directed graph G represents a symmetric network. Then, G is identifiable.

Proof: The attribute of every link can be uniquely determined according to the previous proposition. ■

In reality, communication links almost always come in pairs between adjacent network nodes. However, the attributes of the forward and backward links are usually not identical. For instance, the link delay depends on the traffic load, which is rarely symmetric. Consequently, the results about the symmetric network do not generally apply. On the other hand, by the proof of Proposition 2.4, the combined attributes of the forward and backward links can be uniquely determined between any two adjacent nodes. For purposes such as fault discovery, this can be adequate.

D. Strong Unidentifiability of Asymmetric Networks

In this section, we will prove Proposition 2.2. Again, we assume that G satisfies Assumption 2.1.

The dimension of matrix R is $M \times L$, where M is equal to the number of routes, and L is the number of links. Let the rows of R be vectors $v'_1, v'_2, \ldots, v'_M$, and let the columns of R be vectors $w_1, w_2, \ldots, w_L$. For each $i \in \{1, 2, \ldots, L\}$, define vectors $e_i \in \mathbb{R}^L$ to be of the form $(0, \ldots, 0, 1, 0, \ldots, 0)'$ with 1 in the ith location.

Lemma 2.6: Every column vector of R can be expressed as a linear combination of other column vectors.

Proof: Let us fix i, for $i \in \{1, 2, \ldots, L\}$. The ith column vector of R corresponds to link i in the directed graph $G = [A, U]$. Let $a \in A$ and $b \in A$ be the two nodes to which link i is connected. Since all measurement nodes are separated by a distance of at least 2, at least one of these nodes, say a, is not a measurement node. Then, a must be a router node by the assumption that every link including link i must be on at least one route. Let $U_a^+ \subseteq U$ be the set of outgoing links starting from node a, and $U_a^- \subseteq U$ be the set of incoming links to node a. Then, we have the following identity:

$$\sum_{j \in U_a^-} w_j = \sum_{j \in U_a^+} w_j. \tag{3}$$

To show this, let us fix a k for $k \in \{1, 2, \ldots, M\}$. $\sum_{j \in U_a^-} w_j(k)$ is the number of times the kth route, numbered by the rows of matrix R, enters node a. Here, $w_j(k)$ denotes the kth component of vector w_j. Similarly, $\sum_{j \in U_a^+} w_j(k)$ is the number of times the kth route leaves node a. Since a is a router node and is not a measurement node, any route which enters it must leave it. Hence, we get the equality of (3). Since link i is either in U_a^+ or in U_a^-, by rearranging (3), we get

$$w_i = \begin{cases} \sum_{j \in U_a^+} w_j - \sum_{j \in U_a^-, j \neq i} w_j, & \text{if } i \in U_a^- \\ \sum_{j \in U_a^-} w_j - \sum_{j \in U_a^+, j \neq i} w_j, & \text{if } i \in U_a^+ \end{cases}. \tag{4}$$

■

Corollary 2.7: G is unidentifiable.

Proof: By Lemma 2.6, matrix R does not have full rank. ■

Lemma 2.8: For every $i \in \{1, 2, \ldots, L\}$, $e_i \notin span\{v_1, v_2, \ldots, v_M\}$.

Proof: We need to show that there is not a nonzero row vector $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_M)$ such that $e_i = \sum_{j=1}^{M} \lambda_j v_j$. Note that, if we write $e_i = \sum_{j=1}^{M} \lambda_j v_j$, then $e'_i = \lambda R = (\lambda w_1, \lambda w_2, \ldots, \lambda w_L)$. Since all components of e_i are 0's except that the ith component is a 1, it is sufficient to show that, for any λ, $\lambda w_j = 0$ for every $j \neq i$ necessarily implies $\lambda w_i = 0$. By Lemma 2.6, we can write $w_i = \sum_{j=1, j \neq i}^{L} \alpha_j w_j$, for some $\alpha_j \in \mathbb{R}$. Then

$$\lambda w_i = \lambda \sum_{j=1, j \neq i}^{L} \alpha_j w_j = \sum_{j=1, j \neq i}^{L} \alpha_j \lambda w_j = 0.$$

■

Finally, we are in the position to show strong unidentifiability.

Proof: (Proposition 2.2) Let us view R as a linear transformation from $\mathbb{R}^L$ to $\mathbb{R}^M$. It is sufficient to show that, for each $i = 1, 2, \ldots, L$, there exists a vector $z_i \in kernel\ R$ such that $z'_i e_i \neq 0$. This is enough because if $z'_i e_i \neq 0$ for a fixed $i \in \{1, 2, \ldots, L\}$, then the ith entry of z_i must be nonzero. Then, if the vector $\hat{x}$ is a solution to (2), $\hat{x} + c z_i$ is also a solution, for any real constant c. In particular, the ith entries for all these solutions are different. That is, the attribute of link i cannot be determined uniquely. Letting i vary, we can conclude that the directed graph G is strongly unidentifiable.

Let us call the span of $\{v_1, v_2, \ldots, v_M\}$ the row space of the linear transformation R. It is a well-known fact that $kernel\ R$ and $row\ space\ R$ are orthogonal complement of each other. For every v such that $v \perp kernel\ R$, it must be true that $v \in row\ space\ R$ [12, p. 138].

By Lemma 2.8, $e_i \notin \mathit{row\ space}\ R$, for every $i \in \{1, 2, \dots, L\}$. Therefore, for each i, there exist a vector $z_i \in \mathit{kernel}\ R$, such that $z_i' e_i \neq 0$. Otherwise, e_i would be in $\mathit{row\ space}\ R$. ■

E. Conclusion of Deterministic Analysis and Its Relevance to Probabilistic Models

In summary, the results of this section show that the deterministic, additive attribute of a link in an asymmetric network cannot be observed unless we measure it directly from the two nodes it is attached to. We loosely say the network cannot be completely observed from a proper subset of the nodes.

Knowing in advance (through other means) the attributes of a subset of the links is generally insufficient for identifying the attributes of all other links. However, it is likely that the attributes of some links can be identified by the following successive graph reduction algorithm. In each directed graph, we remove those links whose attributes are known. For each removed link, we merge the two nodes attached to the link into one node. The attribute of the removed link is subtracted from every end-to-end measurement whose corresponding route contains the removed link. If, in the resulting reduced graph, some link connects two measurement nodes directly, then its attribute becomes known by measurement. Such links can be removed in the same fashion, resulting in another reduced graph. The procedure is repeated until we cannot proceed further. The final reduced graph is strongly unidentifiable.

Link attributes such as delay are often non-negative. In Proposition 2.2, we did not impose the constraint of non-negativity. If such a constraint is imposed, and if all link attributes are strictly positive, Proposition 2.2 is still valid, since the vector of link attributes has an open neighborhood in $\mathbb{R}^L_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ is the set of non-negative real number. If some link attributes are zero, then it is possible that all link attributes can be uniquely determined. This last point makes determining a nonparametric probabilistic model possible, as also noted in [3]. We elaborate this in the following.

Suppose all link delays are random and independent from each other, and suppose we wish determine the delay distribution, given route delay samples obtained on end-to-end routes. According to the deterministic analysis, this can not be done, in general, if we do not know any probabilistic structure of the delays. One of the simplest such structures is that, for every link, there is nonzero probability that the link delay is zero. This turns out to be sufficient for determining the link delay distributions. The intuition is that, for every link i, one can always find an end-to-end measurement sample in which the following is true: There is one route containing link i such that all links on the route except i have zero delay. This idea has been fully developed in [3], where the objective is to recover the (nonparametric) distributions of the link delays.

Of course, we can assume much richer probabilistic structure. For instance, the probability distribution is from a known family, such as the exponential random variables. This so-called parametric model is the subject of the next section. It has the advantage over the nonparametric model in that only a small number of parameters need to be identified.

III. Probabilistic Case—Parametric Models

Link attributes, such as delay, are often random, which, on one hand, alters the objectives we can pursue, and on the other hand, adds a different set of information that we can utilize. It is entirely possible that the network can be observed from a subset of the nodes. In this section, we will study the case where the link attributes can be specified by some parametric probability models. The objective is to recover the unknown parameters of the model through statistical inference.

A parametric probability model is one whose distribution function has known functional form with some parameters. The distribution is completely specified if the parameters are known. Parametric models pose certain difficulties to our problem. First, often times, the model only becomes useful if we determine all its parameters. This can be analytically and/or computationally difficult for additive link attributes, since it is often difficult to handle the sum of random variables. Second, by stipulating that the link attribute follows a parametric model, we have made a very strong assumption. Even if we can estimate the parameters correctly and efficiently, the contribution may not be great if the model is invalid when compared with the reality. However, there is also a more relaxed view about parametric models. Suppose the true probability distribution of the link attribute is not from the family of models that we specify. We may even suppose that it can only be described nonparametrically, which is most likely the case in reality. We can still apply the parametric model and choose suitable parameters to "curve-fit" the true distribution. What we achieve is a parsimonious and approximate description of the true distribution. We will see in Section IV that it is in this view that our exploration of the parametric models is more useful.

Throughout the discussion on the probabilistic models, including both the parametric and nonparametric models, we assume all link attributes are independent of each other at all time instances, and attributes of the same link at different time instances are independent.

A. Gaussian Model

Let $G = [A, U]$ be the directed graph representing the network. Suppose Assumption 2.1 is satisfied. Suppose there are total L links in G numbered as $1, 2, \dots, L$, and there are total M routes, numbered as $1, 2, \dots, M$. For each link i, let its link attribute X_i be a Gaussian random variable with mean μ_i and variance σ_i^2. Since $X_1, X_2, \dots, X_L$ are independent, they are multivariate Gaussian with the mean vector $\mu = (\mu_1, \mu_2, \dots, \mu_L)'$ and covariance matrix $\Sigma = \mathit{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_L^2)$. Let $X = (X_1, X_2, \dots, X_L)'$. Let R be the route matrix associated with G, and let $Y = (Y_1, Y_2, \dots, Y_M)'$ be the route attributes on routes $1, 2, \dots, M$. Since $Y = RX$, Y is also multivariate Gaussian with the mean vector $R\mu$ and covariance matrix $\Sigma_Y = R\Sigma R'$.

Denote by P_θ the multivariate Gaussian distribution for the random vector Y with parameter $\theta \in \Theta$, where Θ is the parameter space. In this case, $\theta = (R\mu, \Sigma_Y)$.

Definition 3.1: A parametric model is said to be *identifiable* if $\theta_1 \neq \theta_2$ implies $P_{\theta_1} \neq P_{\theta_2}$, for all $\theta_1, \theta_2 \in \Theta$. Otherwise, we say it is *unidentifiable*.

Then, we have the following theorem.

Theorem 3.1: If no measurement nodes are adjacent in the directed graph G, the Gaussian model P_θ specified above is unidentifiable.

Proof: Let $K = rank\, R$. By Proposition 2.2, $K < L$. This implies we can select K routes so that any route vector (i.e., a row in R) is a linear combination of the selected K routes. Therefore, without the loss of generality, we can assume R is a $K \times L$ route matrix with full row rank. From $Y = RX$, Y is a multivariate Gaussian with mean $R\mu$ and covariance matrix $\Sigma_Y = R\Sigma R'$. Σ_Y is invertible. The distribution of Y is completely determined by its mean vector and the covariance matrix. It depends on μ through $R\mu$. Since $kernel\, R \neq \{0\}$, there exist vectors $\mu^1 \neq \mu^2$ such that $R\mu^1 = R\mu^2$. ■

Theorem 3.1 is a rather disappointing result. Multivariate Gaussian random variables have some nice properties that make many calculations simpler. For instance, the joint distribution is completely specified by the mean and the covariance matrix. A linear transformation of the multivariate Gaussian random variables is also Gaussian. The mean and covariance of the transformed Gaussian can be obtained by linear transformations on the original mean and covariance. These nice properties works against us in this case, where the linear transformation by R is not injective.

It might be worth pointing out that the variances of link attributes can be identified in certain networks, as will be shown in Section III-B2. In studying the traffic matrix estimation problem, [6] considers a Gaussian model where the mean of each Gaussian random variable depends on its variance. Such a model can be identifiable.

B. Exponential Model

The exponential model is a often assumed model for the distributions of link delays. Suppose that, for each $i \in \{1, 2, \ldots, L\}$, the attribute of link i, X_i, is characterized by an exponential distribution with parameter λ_i, denoted by $X_i \sim exp(\lambda_i)$, and suppose $X_1, X_2, \ldots, X_L$ are independent. Since the parameters λ_i's are from the space of positive real numbers, and there are a finite number of links in question, we will assume that all λ_i's are distinct. We will first investigate the identifiability issue of a single route.

Lemma 3.2: The distribution of any route attribute is identifiable up to an ordering.

Proof: Without loss of generality, let us assume the route in question, denoted by r, has I links, $1, 2, \ldots, I$. The link attributes $X_1, X_2, \ldots, X_I$ are independent exponential random variables with parameter $\lambda_1, \lambda_2, \ldots, \lambda_I$. Let Y_r be the route attribute, i.e., $Y_r = \sum_{i=1}^{I} X_i$. Since the moment generating function for an exponential random variable with parameter λ_i is $\lambda_i/(\lambda_i - s)$, the moment generating function for Y_r is

$$G(s) = \prod_{i=1}^{I} \frac{\lambda_i}{\lambda_i - s}. \tag{5}$$

If the set $\lambda^1 = \{\lambda_1^1, \lambda_2^1, \ldots, \lambda_I^1\}$ is not the same as the set $\lambda^2 = \{\lambda_1^2, \lambda_2^2, \ldots, \lambda_I^2\}$, the resulting moment generating functions will be different, also. Since we can find the unique distribution function corresponding to the moment generating function of the form in (5), by the definition of identifiability, the distribution of any route attribute is identifiable if we ignore the order of the I links in the route. ■

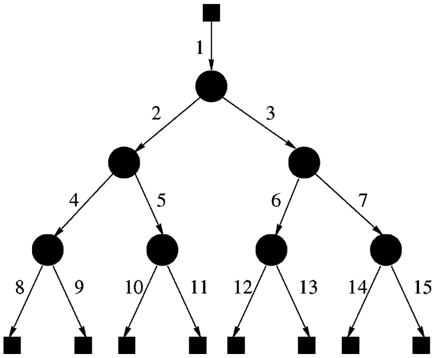


Fig. 1. A binary tree.

Suppose it is possible to estimate the (unordered) set of λ_i's correctly on all routes. It is not difficult to determine the correspondence between the links and the parameters λ_i's on some directed graphs. Consider the graph of a binary tree in Fig. 1 with one sender node at the root, eight receiver nodes at the leaves, seven intermediate router nodes, and 15 links. There are eight routes, each associated with one of the receivers. Number the routes $1, 2, \ldots, 8$ from left to the right. Suppose the estimates of $\lambda_1, \lambda_2, \ldots, \lambda_{15}$ based on the observations on each route are exact so that we can consider the estimates and the true parameter values are equivalent. Since the estimation based on route 1 yields $\{\lambda_1, \lambda_2, \lambda_4, \lambda_8\}$ and the estimation based on route 2 yields $\{\lambda_1, \lambda_2, \lambda_4, \lambda_9\}$, we can conclude that λ_8 must be associated with X_8 and that λ_9 must be associated with X_9. In this fashion, we can find the parameters λ_i's for all links at the bottom level. Recursively, we can move one level up in the graph and identify the parameters for X_4, X_5, X_6, and X_7. For example, consider route 1, which is associated with the parameter set $\{\lambda_1, \lambda_2, \lambda_4, \lambda_8\}$, and route 3, which is associated with the parameter set $\{\lambda_1, \lambda_2, \lambda_5, \lambda_{10}\}$. Since we have identified X_8 with λ_8 and X_{10} with λ_{10} when we consider the bottom level, we can conclude that λ_4 must be associated with X_4 and that λ_5 must be associated with X_5.

The above reasoning leads to the following lemma.

Lemma 3.3: For an exponential model in which all parameters λ_i's are distinct, it is possible to construct an identifiable directed graph in which the measurement nodes are not necessarily adjacent.

This is in contrast with both the deterministic case and the Gaussian model.

In order to estimate the parameters of link attributes based on the observations of the route attribute, it is natural to consider a maximum-likelihood estimator. However, maximizing the likelihood of the sum of independent exponentials is difficult. Consider the following example. Let the random variable Y be the sum of three independent exponential random variables with parameters λ_1, λ_2, and λ_3. Suppose we make N independent observations of Y, and let $Y^{(n)}$ denote the nth observation. Let the random vector $\mathbf{Y} = (Y^{(1)}, Y^{(2)}, \ldots, Y^{(N)})$. Let $P_{\mathbf{Y}}(y^{(1)}, y^{(2)}, \ldots, y^{(N)})$ denote the density of $\mathbf{Y}$. $P_{\mathbf{Y}}$ is also known as the likelihood function. After some algebra, we get

$$\begin{aligned}P_{\mathbf{Y}}&\left(y^{(1)},y^{(2)},\ldots,y^{(N)}\right)\\&=\prod_{n=1}^{N}\Bigg(\frac{\lambda_1\lambda_2\lambda_3}{(\lambda_2-\lambda_1)(\lambda_2-\lambda_1)}e^{-\lambda_1 y^{(n)}}\\&\qquad+\frac{\lambda_1\lambda_2\lambda_3}{(\lambda_1-\lambda_2)(\lambda_3-\lambda_2)}e^{-\lambda_2 y^{(n)}}\\&\qquad+\frac{\lambda_1\lambda_2\lambda_3}{(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)}e^{-\lambda_3 y^{(n)}}\Bigg).\end{aligned}$$

We want to obtain the parameters λ that maximize the likelihood function $P_{\mathbf{Y}}$ or the log-likelihood function. However, we know no easy way to solve this analytically because the derivatives with respect to the λ_i's appear to be quite complex. The difficulty is in part due to the sum in the expression of the likelihood function, which becomes log of the sum in the log-likelihood function.

There are two solutions to our dilemma. The first one is an iterative technique, called the EM algorithm, which is widely used in maximum-likelihood-based parameter estimation with incomplete data. The second approach is to deviate from maximum-likelihood-based estimation and use the method of moments. We will discussion each of these two techniques in the following.

1) EM Algorithm: Let us consider a single route with I links, numbered as $1,2,\ldots,I$. Let the link delay be $X_i \sim exp(\lambda_i)$ for link i, $i=1,2,\ldots,I$, and the X_i's are independent. We assume the parameters λ_i's are different. Let the route delay be Y. Then, $Y=\sum_{i=1}^{I}X_i$. Suppose N independent samples of Y are observed, denoted by $\mathbf{Y}=(Y^{(1)},Y^{(2)},\ldots,Y^{(N)})$. At each time n, the delay of link i is denoted by $X_i^{(n)}$. Let $X^{(n)}=(X_1^{(n)},X_2^{(n)},\ldots,X_I^{(n)})'$, and let $\mathbf{X}=(X^{(1)},X^{(2)},\ldots,X^{(N)})$. Let $\lambda=(\lambda_1,\lambda_2,\ldots,\lambda_I)$.

The EM algorithm is an iterative algorithm for finding the parameters λ that maximize the log-likelihood, denoted $l_\lambda(\mathbf{Y})=\log P_{\mathbf{Y}}(\mathbf{Y}\mid\lambda)$ [4]. Notice that the random variables $\mathbf{Y}$ are observed. The random variables $\mathbf{X}$ are not observed, and are called hidden random variables. In our case, if $\mathbf{X}$ were observed, then $\mathbf{Y}$ would be completely determined. At each step of the iteration, we have a current estimate of the parameters, denoted by $\lambda^{(t)}$, where t stands for the tth iteration. At the tth iteration, let the conditional density of the hidden variables $\mathbf{X}$ conditional on the observed random variables $\mathbf{Y}$ be $P_{\mathbf{X}|\mathbf{Y}}(\mathbf{X}\mid\mathbf{Y};\lambda^{(t)})$.

The E-step of the EM algorithm is to write the *complete log-likelihood* (of all random variables) with generic parameters λ, under the assumption that the hidden variables are also observed. Then, the conditional expectation of the complete log-likelihood is evaluated with respect to the conditional density $P_{\mathbf{X}|\mathbf{Y}}(\mathbf{X}\mid\mathbf{Y};\lambda^{(t)})$. The M-step of the algorithm is to find a new set of parameters, denoted by $\lambda^{(t+1)}$, that maximize the expected complete log-likelihood computed in the E-step.

We now apply the EM algorithm to our case. The complete likelihood is denoted by $L_\lambda^c(\mathbf{X})=P_{\mathbf{X}}(\mathbf{X}\mid\lambda)$. $L_\lambda^c(\mathbf{X})$ is given by

$$L_\lambda^c(\mathbf{X})=\prod_{n=1}^{N}\prod_{i=1}^{I}\lambda_i e^{-\lambda_i X_i^{(n)}}=\prod_{n=1}^{N}\left(\prod_{i=1}^{I}\lambda_i\right)e^{-\sum_{i=1}^{I}\lambda_i X_i^{(n)}}.$$

Then, the complete log-likelihood, denoted by $l_\lambda^c(\mathbf{X})$ is

$$l_\lambda^c(\mathbf{X})=\sum_{n=1}^{N}\left(\sum_{i=1}^{I}\log\lambda_i-\sum_{i=1}^{I}\lambda_i X_i^{(n)}\right).\qquad(6)$$

In the tth iteration step and in the E-step of the EM algorithm, the expected value of the complete log-likelihood is

$$\begin{aligned}&\mathbb{E}\left[l_\lambda^c(\mathbf{X})|\mathbf{Y},\boldsymbol{\lambda}^{(t)}\right]\\&\quad=\sum_{n=1}^{N}\left(\sum_{i=1}^{I}\log\lambda_i-\sum_{i=1}^{I}\lambda_i\mathbb{E}\left[X_i^{(n)}|\mathbf{Y},\lambda^{(t)}\right]\right).\qquad(7)\end{aligned}$$

The M-step is to choose $\lambda^{(t+1)}$ that maximizes the expected log-likelihood $\mathbb{E}[l_\lambda^c(\mathbf{X})\mid\mathbf{Y},\lambda^{(t)}]$. That is

$$\lambda^{(t+1)}=\arg\max_{\lambda\in\Theta}\mathbb{E}\left[l_\lambda^c(\mathbf{X})|\mathbf{Y},\lambda^{(t)}\right]\qquad(8)$$

where the parameter space Θ is, $\Theta=\{(\lambda_1,\lambda_2,\ldots,\lambda_I):\lambda_i>0,i=1,2,\ldots,I\}$. The M-step in our case is fairly simple. By taking derivative with respect to the λ_i's, the optimization yields

$$\lambda_i^{(t+1)}=\frac{N}{\sum_{n=1}^{N}\mathbb{E}\left[X_i^{(n)}|\mathbf{Y},\lambda^{(t)}\right]}\quad\text{for i}=1,2,\ldots,\text{I}.\qquad(9)$$

Hence, the key for applying the EM algorithm is to compute $\mathbb{E}[X_i^{(n)}\mid\mathbf{Y},\lambda^{(t)}]$, for $i=1,2,\ldots,I$ at each iteration step. By independence, $\mathbb{E}[X_i^{(n)}\mid\mathbf{Y},\lambda^{(t)}]=\mathbb{E}[X_i^{(n)}\mid Y^{(n)},\lambda^{(t)}]$, where $Y^{(n)}=\sum_{i=1}^{I}X_i^{(n)}$. We will next compute this value. For convenience, we drop the time index n and iteration index t. Let $P_Y(\cdot\mid\lambda)$ denote the density for the random variable Y. It can be shown that

$$P_Y(y|\lambda)=\sum_{i=1}^{I}\left(\prod_{j=1,j\neq i}^{I}\frac{\lambda_j}{\lambda_j-\lambda_i}\right)\lambda_i e^{-\lambda_i y}.\qquad(10)$$

Then, for any real vector $x=(x_1,x_2,\ldots,x_I)$ such that $x_i\geq 0$ for all i, and $\sum_{i=1}^{I}x_i=y$

$$P_{X|Y}(x|y;\lambda)=\frac{\prod_{i=1}^{I}\lambda_i e^{-\lambda_i x_i}}{P_Y(y|\lambda)}.\qquad(11)$$

Because $y=\sum_{i=1}^{I}x_i$, there are only $I-1$ independent x_i's in (11). In principle, we can find the expression for the conditional density $P_{X_i|Y}(x_i\mid y;\lambda)$ by integrating the right-hand side of (11) with respect to the other $I-2$ variables. But, we can do this more simply by noting that $Y=X_i+W_i$, where $W_i=\sum_{k=1,k\neq i}^{I}X_k$ has the density $P_{W_i}(w_i\mid\lambda)$. By the general expression in (10) for the density of the sum of independent exponentials

$$P_{W_i}(w_i|\lambda)=\sum_{k=1,k\neq i}^{I}\left(\prod_{j=1,j\neq k,j\neq i}^{I}\frac{\lambda_j}{\lambda_j-\lambda_k}\right)\lambda_k e^{-\lambda_k w_i}.$$

Therefore, for $0 \leq x_i \leq y$

$$P_{X_i|Y}(x_i|y;\lambda) = \frac{\lambda_i e^{-\lambda_i x_i} P_{W_i}(y - x_i|\lambda)}{P_Y(y|\lambda)}. \tag{12}$$

The conditional expectation of X_i can then be computed. We get the closed-form expression

$$\mathbb{E}[X_i|Y,\lambda] = \frac{1}{P_Y(Y|\lambda)} \sum_{k=1,k\neq i}^{I} \left(\prod_{j=1,j\neq k}^{I} \frac{\lambda_j}{\lambda_j - \lambda_k} \right) \cdot \left(\frac{\lambda_k e^{-\lambda_k Y}}{\lambda_i - \lambda_k} - \frac{\lambda_k e^{-\lambda_i Y}}{\lambda_i - \lambda_k} - \lambda_k Y e^{-\lambda_i Y} \right). \tag{13}$$

Substituting the result from (13) into (9) with the correct time and iteration indices, we then have an iterative procedure to compute the parameters λ.

2) Method of Moments: The following heuristic algorithm based on the method of moments can be an alternative for estimating the parameters in the exponential model. Similar approach was taken in [2]. Again, consider the example shown in Fig. 1. From $Y_1 = X_1 + X_2 + X_4 + X_8$ and $Y_2 = X_1 + X_2 + X_4 + X_9$, the covariance of Y_1 and Y_2 is

$$\mathrm{Cov}(Y_1, Y_2) = \mathrm{Var}(X_1 + X_2 + X_4).$$

Here, $\mathrm{Cov}(\cdot)$ and $\mathrm{Var}(\cdot)$ stand for covariance and variance. From

$$\mathrm{Var}(Y_1) = \mathrm{Var}(X_1 + X_2 + X_4) + \mathrm{Var}(X_8),$$

we get

$$\mathrm{Var}(X_8) = \mathrm{Var}(Y_1) - \mathrm{Cov}(Y_1, Y_2).$$

Both quantities on the right-hand side above can be estimated based on the measurement samples collected for the routes. Since the variance of an exponential random variable with parameter λ_i is $1/\lambda_i^2$, the parameter λ_8 can be estimated. In a similar fashion, the parameters λ_9 to λ_{15} can all be estimated from the variances for the bottom-level links in the tree. We can then move up one level from the bottom and estimate the parameters for link 4, 5, 6, and 7. For example, taking $Y_1 = X_1 + X_2 + X_4 + X_8$ and $Y_3 = X_1 + X_2 + X_5 + X_{10}$, we get

$$\mathrm{Var}(X_4) = \mathrm{Var}(Y_1) - \mathrm{Cov}(Y_1, Y_3) - \mathrm{Var}(X_8).$$

By now, we have estimates for all quantities on the right-hand side. It is obvious that this algorithm can be continued upward until all link parameters are determined.

We stress that this method requires multicast on the binary tree for collecting the samples for the Y_i's. In other words, each time a packet is sent at the root node, 2^{n-1} samples are collected at the bottom level, one for each route, where n is the depth of the tree. In the following, we will discuss some issues related to this heuristic approach.

1) The estimator based on the method of moments may not be asymptotically efficient, while a maximum-likelihood estimator usually is. For an exponential random variable, the sample mean estimator and the maximum-likelihood estimator are the same. We might lose efficiency when the sample variance is used in estimating the parameter. Nevertheless, we expect the estimator proposed here to be reasonably efficient, particularly when the depths of the binary tree is small. The major advantage of this estimation scheme is its computational simplicity.
2) In the Gaussian model, the variances can be estimated using the method of moments. However, since the means and variances for the Gaussian model are unrelated, the method gives no information about the means. Our deterministic analysis shows that applying the method of moments only to the first moment will not yield unique estimates of the model parameters. That is why, in the exponential model, we need to rely on the second moment.
3) The method here can be used also for the one-parameter Gamma model, in which each link attribute is considered to be a Gamma random variable with known order n_i and unknown parameter λ_i. The density is of the form $\lambda_i e^{-\lambda_i x}(\lambda_i x)^{n_i - 1}/(n_i - 1)!$, where $x \geq 0$, and the variance is n_i/λ_i^2.

The method of moments is in fact a family of estimation methods that rely on the moments of the random variables. Here is another one of these methods. Again, consider the example shown in Fig. 1. From $Y_1 = X_1 + X_2 + X_4 + X_8$ and $Y_2 = X_1 + X_2 + X_4 + X_9$, we get $Y_1 - Y_2 = X_8 - X_9$. Notice that $Y_1 - Y_2$ can be observed. Since the mean of an exponential random variable with parameter λ_i is $1/\lambda_i$ and the variance is $1/\lambda_i^2$, we can write

$$\begin{aligned} \mathbb{E}[Y_1 - Y_2] &= 1/\lambda_8 - 1/\lambda_9 \\ \mathrm{Var}(Y_1 - Y_2) &= 1/\lambda_8^2 + 1/\lambda_9^2. \end{aligned}$$

Given a random variable Z, let us define the following notations. Let $\hat{m}(Z)$ be the sample mean and $\hat{\sigma}^2(Z)$ be the sample variance. Then, assuming certainty equivalence, i.e., the estimates of the mean and the variance are in fact the mean and the variance of the random variable, it is reasonable to set up the following equations, and solve for λ_8 and λ_9

$$\begin{aligned} \hat{m}(Y_1 - Y_2) &= 1/\lambda_8 - 1/\lambda_9 \\ \hat{\sigma}^2(Y_1 - Y_2) &= 1/\lambda_8^2 + 1/\lambda_9^2. \end{aligned}$$

The positive solutions to the above equations are unique

$$\begin{aligned} \frac{1}{\lambda_8} &= \frac{\hat{m}(Y_1 - Y_2) + \sqrt{2\hat{\sigma}^2(Y_1 - Y_2) - (\hat{m}(Y_1 - Y_2))^2}}{2}, \\ \frac{1}{\lambda_9} &= \frac{-\hat{m}(Y_1 - Y_2) + \sqrt{2\hat{\sigma}^2(Y_1 - Y_2) - (\hat{m}(Y_1 - Y_2))^2}}{2}. \end{aligned}$$

In the similar fashion, all parameters associated with the links at the bottom level can be solved. Then, we can move one level up in the tree and try to solve the parameters associated with links in that level. For example, $Y_1 - Y_3 = X_4 + X_8 - X_5 -$

X_{10}. Since the parameters for X_8 and X_{10} have already been estimated, we consider them known constants. By independence of the X_i's and by certainty equivalence, we set up the following equations and want to solve for λ_4 and λ_5:

$$\hat{m}(Y_1 - Y_3) = 1/\lambda_4 + 1/\lambda_8 - 1/\lambda_5 - 1/\lambda_{10}$$
$$\hat{\sigma}^2(Y_1 - Y_3) = 1/\lambda_4^2 + 1/\lambda_8^2 + 1/\lambda_5^2 + 1/\lambda_{10}^2.$$

As a final note on the exponential model, if each link delay is a sum of J independent exponentials with J parameters (assume J is a known constant), the techniques used for the exponential model can be applied.

C. Mixture of Exponential Model

Some may argue that the exponential model is too simple to model link delays. In this section, we present a different model, the mixture of exponentials. This model is motivated by the observation that the link delay distribution may have different decaying tails at different time, possibly due to the difference in the traffic load. If we consider each decaying tail as a mode, the link delay switches among these modes. Another motivation is that the power-decaying tail can be approximated quite nicely with this model, as shown in [13]. More generally, the mixture of exponential model may have enough degree-of-freedom for approximating a wide range of distributions.

Specifically, let us model the attribute of link i, X_i, as a mixture of J exponentials, each with a parameter $\lambda_{i,j}$, for $j = 1, 2, \ldots, J$, where J is a known constant. Let $\pi_{i,j}$ be the probability that X_i takes one of these J exponentials, with $\sum_{j=1}^{J} \pi_{i,j} = 1$. Let $\lambda_i = (\lambda_{i,j}, \lambda_{i,2}, \ldots, \lambda_{i,J})$. We can write the density for X_i

$$P_{X_i}(x_i|\lambda_i) = \sum_{j=1}^{J} \pi_{i,j}\lambda_{i,j}e^{-\lambda_{i,j}x_i}. \quad (14)$$

Again, assume $Y = \sum_{i=1}^{I} X_i$, where Y is the route attribute. We would like to estimate the parameters λ and π based on the observations of Y. We formulate a maximum-likelihood estimator and apply the EM algorithm, because the estimation problem can be considered in a setting with unobserved hidden variables. First, the link attributes, X_i's, are unobserved. Second, we can regard the particular mode a link attribute chooses as a hidden variable. Let us define a vector-valued random variable Q_i associated with each X_i taking values in the set $\{e_1, e_2, \ldots, e_J\}$, where $e_j \in \mathbb{R}^J$ is a vector $(0, \ldots, 0, 1, 0, \ldots, 0)'$, with the 1 in the jth position. Denote $\mathbf{X} = (X_i^{(n)})$ and $\mathbf{Q} = (Q_i^{(n)})$, where $i = 1, 2, \ldots, I$ and $n = 1, 2, \ldots, N$. Then, the complete likelihood is

$$L_{\lambda,\pi}^c(\mathbf{X}, \mathbf{Q}) = \prod_{n=1}^{N}\prod_{i=1}^{I}\prod_{j=1}^{J}\left(\pi_{i,j}\lambda_{i,j}e^{-\lambda_{i,j}X_i^{(n)}}\right)^{Q_{i,j}^{(n)}}. \quad (15)$$

Here, we make the notation more compact by letting the discrete variables $\mathbf{Q}$ vary. Note that $Q_{i,j}^{(n)}$ stands for the jth entry of the vector $Q_i^{(n)}$, which is either 1 or 0. For each index i, only one of the J terms indexed by j survives. The complete log-likelihood is

$$l_{\lambda,\pi}^c(\mathbf{X}, \mathbf{Q}) = \log L_{\lambda,\pi}^c(\mathbf{X}, \mathbf{Q})$$
$$= \sum_{n=1}^{N}\sum_{i=1}^{I}\sum_{j=1}^{J} Q_{i,j}^{(n)}\left(\log \pi_{i,j} + \log \lambda_{i,j} - \lambda_{i,j}X_i^{(n)}\right). \quad (16)$$

To simplify the notation, let $\langle\cdot\rangle$ represent the conditional expectation operator $\mathbb{E}[\,\cdot \mid \mathbf{Y}, \lambda^{(t)}, \pi^{(t)}]$. The expected value of the complete log-likelihood is

$$\left\langle l_{\lambda,\pi}^c(\mathbf{X}, \mathbf{Q})\right\rangle = \sum_{n=1}^{N}\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\left\langle Q_{i,j}^{(n)}\right\rangle \log \pi_{i,j}\right.$$
$$\left. + \left\langle Q_{i,j}^{(n)}\right\rangle \log \lambda_{i,j} - \lambda_{i,j}\left\langle Q_{i,j}^{(n)}X_i^{(n)}\right\rangle\right). \quad (17)$$

In the M-step of the EM algorithm, we would like to find $\lambda^{(t+1)}$ and $\pi^{(t+1)}$ so that $\langle l_{\lambda,\pi}^c(\mathbf{X}, \mathbf{Q})\rangle$ is maximized. This is a constraint optimization problem, subject to the constraint that $\sum_{j=1}^{J} \pi_{i,j} = 1$ for $i = 1, 2, \ldots, I$. It is easy to show that, for $i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, J$

$$\pi_{i,j}^{(t+1)} = \sum_{n=1}^{N}\left\langle Q_{i,j}^{(n)}\right\rangle/N \quad (18)$$
$$\lambda_{i,j}^{(t+1)} = \sum_{n=1}^{N}\left\langle Q_{i,j}^{(n)}\right\rangle \Big/ \sum_{n=1}^{N}\left\langle X_i^{(n)}Q_{i,j}^{(n)}\right\rangle. \quad (19)$$

Therefore, the key challenge is to compute $\langle Q_{i,j}^{(n)}\rangle$ and $\langle X_i^{(n)}Q_{i,j}^{(n)}\rangle$. Since samples at different time instances are assumed to be independent, the conditioning in the above expressions of conditional expectations is on $Y^{(n)}$ alone. In order to compute the above expectations, we need to evaluate the posterior probabilities conditional on the observation of $Y^{(n)}$. We will drop the time index n and iteration index t in the following analysis.

1) Some General but Difficult Approaches: Let the random vectors $X = (X_i)$ and $Q = (Q_i)$, for $i = 1, 2, \ldots, I$, and x and q be the corresponding nonrandom versions of them. One strategy is to start with the joint conditional density, $P_{X,Q|Y}(x, q \mid y; \lambda, \pi)$. More explicitly, for any x such that $x_i \geq 0$ for all i, and $\sum_{i=1}^{I} x_i = y$, we need to compute

$$P_{X,Q|Y}(x, Q_{1,j_1} = 1, Q_{2,j_1} = 1, \ldots, Q_{I,j_I} = 1|y; \lambda, \pi)$$
$$= \frac{\prod_{i=1}^{I} \pi_{i,j_i}\lambda_{i,j_i}e^{-\lambda_{i,j_i}x_i}}{P_Y(y|\lambda, \pi)} \quad (20)$$

for all I-tuples $(j_1, j_2, \ldots, j_I) \in \{1, 2, \ldots, J\}^I$, where

$$P_Y(y|\lambda, \pi) = \sum_{j_1, j_2, \ldots, j_I \in \{1, 2, \ldots, J\}^I}$$
$$\times\left(\sum_{i=1}^{I} \lambda_{i,j_i}e^{-\lambda_{i,j_i}y}\prod_{k=1, k\neq i}^{I}\frac{\lambda_{k,j_k}}{\lambda_{k,j_k} - \lambda_{i,j_i}}\right)\prod_{i=1}^{I}\pi_{i,j_i}. \quad (21)$$

We can then compute $P_{X_i,Q_{i,j}|Y}$ and $P_{Q_{i,j}|Y}$ by marginalizing the joint conditional density. The procedure developed here is a general one for mixture models, including the mixture of exponentials and the mixture of exponentials and Gaussians. However, the computation complexity is exponential in the number of links, I, due to the discrete nature of the random variables, Q_i's. For example, if we choose $k = 2$ and $L = 10$, then we need to compute (20) 2^{10} times for each observation of Y, and the outer sum in (21) has also 2^{10} terms.

A simpler approach is to first compute the conditional density $P_{X|Y}(x \mid y; \lambda, \pi)$

$$P_{X|Y}(x|y;\lambda,\pi) = \frac{\prod_{i=1}^{I}\left(\sum_{j=1}^{J}\pi_{i,j}\lambda_{i,j}e^{-\lambda_{i,j}x_i}\right)}{P_Y(y|\lambda,\pi)} \tag{22}$$

where

$$P_Y(y|\lambda,\pi) = \int\limits_{\Omega}\prod_{i=1}^{I}\left(\sum_{j=1}^{J}\pi_{i,j}\lambda_{i,j}e^{-\lambda_{i,j}x_i}\right)dx_1\dots dx_{I-1} \tag{23}$$

where the set

$$\Omega = \{(x_1,\dots,x_{I-1}) : x_i \geq 0, i = 1,\dots,I-1, \text{ and } \sum_{i=1}^{I-1}x_i \leq y\Bigg\}.$$

Then, the conditional density $P_{X_i,Q_i|Y}$ can be computed by

$$\begin{aligned}P_{X_i,Q_i|Y}&(x_i, Q_{i,j} = 1|y;\lambda,\pi)\\ &= P_{Q_i|X_i,Y}(Q_{i,j} = 1|x_i,y;\lambda,\pi)P_{X_i|Y}(x_i|y;\lambda,\pi).\end{aligned} \tag{24}$$

Given X_i, Q_i is independent of Y. Hence

$$\begin{aligned}P_{Q_i|X_i,Y}&(Q_{i,j} = 1|x_i,y;\lambda,\pi)\\ &= P_{Q_i|X_i}(Q_{i,j} = 1|x_i;\lambda,\pi)\\ &= \frac{\pi_{i,j}\lambda_{i,j}e^{-\lambda_{i,j}x_i}}{\sum_{r=1}^{J}\pi_{i,r}\lambda_{i,r}e^{-\lambda_{i,r}x_i}}.\end{aligned} \tag{25}$$

Given the conditional density $P_{X_i,Q_i|Y}$, in principle, $\langle X_iQ_{i,j}\rangle$ can be computed. To compute $\langle Q_{i,j}\rangle$, we also need $P_{Q|Y}(Q_{i,j} = 1 \mid y; \lambda, \pi)$. In the following, notations are simplified when there is no confusion:

$$\begin{aligned}P(Q_{i,j} = 1|y) &= \int\limits_{x_i \leq y} P(Q_{i,j} = 1, x_i|y)dx_i\\ &= \int\limits_{x_i \leq y} P(Q_{i,j} = 1|x_i, y)P(x_i|y)dx_i\\ &= \int\limits_{x_i \leq y} P(Q_{i,j} = 1|x_i)P(x_i|y)dx_i.\end{aligned}$$

Both approaches require complicated numerical integration.

2) Transform-Based Approach: The transform method can further help us to reduce the computation complexity, because of the simple form of the moment generating function for the exponential distribution. In particular, we need not calculate the integration in (23). For simplicity of discussion, suppose all the $\lambda_{i,j}$'s are distinct. The moment generating function of Y can be expressed as

$$G_Y(s) = \prod_{i=1}^{I}\sum_{j=1}^{J}\frac{\pi_{i,j}\lambda_{i,j}}{\lambda_{i,j} - s}. \tag{26}$$

We can perform partial fraction expansion on $G_Y(s)$ and get

$$G_Y(s) = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{\alpha_{i,j}\lambda_{i,j}}{\lambda_{i,j} - s} \tag{27}$$

where

$$\alpha_{i,j} = \pi_{i,j}\prod_{k\neq i,k=1}^{I}\sum_{l=1}^{J}\frac{\pi_{k,l}\lambda_{k,l}}{\lambda_{k,l} - \lambda_{i,j}}. \tag{28}$$

Hence

$$P_Y(y) = \sum_{i=1}^{I}\sum_{j=1}^{J}\alpha_{i,j}\lambda_{i,j}e^{-\lambda_{i,j}y}. \tag{29}$$

We now discuss how to compute $\mathbb{E}[Q_{i,j} \mid Y]$

$$\begin{aligned}\mathbb{E}[Q_{i,j}|Y = y] &= P(Q_{i,j} = 1|Y = y)\\ &= \frac{P_{Y,Q_{i,j}}(y, Q_{i,j} = 1)}{P_Y(y)}\\ &= \frac{P(Q_{i,j} = 1)P_{Y|Q_{i,j}}(y|Q_{i,j} = 1)}{P_Y(y)}\\ &= \frac{\pi_{i,j}P_{Y|Q_{i,j}}(y|Q_{i,j} = 1)}{P_Y(y)}.\end{aligned} \tag{30}$$

Given $Q_{i,j} = 1$

$$Y = X_1 + \cdots + X_{i-1} + X_{i,j} + X_{i+1} + \cdots + X_I$$

where $X_{i,j}$ denotes the jth "mode" of the random variable X_i. That is, $X_{i,j}$ is an independent exponential variable with parameter $\lambda_{i,j}$. We see that, conditional on $\{Q_{i,j} = 1\}$, Y is again a sum of mixtures of exponential random variables, different from the unconditional Y only in that $\pi_{i,l} = 1$ for $l = j$ and 0 otherwise. Its pdf is given by (29) with suitably modified $\alpha_{k,l}$'s, denoted by $\alpha'_{k,l}$, where

$$\alpha'_{k,l} = \begin{cases}\alpha_{k,l}, & k \neq i\\ \alpha_{i,j}/\pi_{i,j}, & k = i, l = j\\ 0, & k = i, l \neq j\end{cases}.$$

Next, we shall discuss how to compute $\mathbb{E}[X_iQ_{i,j} \mid Y]$. Note that, conditional on $\{Y = y\}$ or not

$$X_iQ_{i,j} = \begin{cases}X_i, & \text{if } Q_{i,j} = 1\\ 0, & \text{otherwise}\end{cases}.$$

We have

$$\begin{aligned}\mathbb{E}&[X_iQ_{i,j}|Y = y]\\ &= \mathbb{E}[X_i|Y = y, Q_{i,j} = 1]P(Q_{i,j} = 1|Y = y).\end{aligned} \tag{31}$$

We have just discussed how to compute $P(Q_{i,j} = 1 \mid Y = y)$ in (30). However, we will show that it is not necessary to compute it here. In order to compute $\mathbb{E}[X_i \mid Y = y, Q_{i,j} = 1]$, we need to compute $P(X_i \mid Y = y, Q_{i,j} = 1)$. For $0 \leq x_i \leq y$

$$\begin{aligned} &P_{X_i|Y,Q_{i,j}}(x_i|Y=y, Q_{i,j}=1) \\ &\quad = \frac{P_{X_i,Y|Q_{i,j}}(x_i, y|Q_{i,j}=1)}{P_{Y|Q_{i,j}}(y|Q_{i,j}=1)} \\ &\quad = \frac{P_{X_i|Q_{i,j}}(x_i|Q_{i,j}=1)P_{Y|X_i,Q_{i,j}}(y|X_i=x_i, Q_{i,j}=1)}{P_{Y|Q_{i,j}}(y|Q_{i,j}=1)}. \end{aligned}$$

Conditional on $Q_{i,j} = 1$, X_i is an exponential random variable with parameter $\lambda_{i,j}$. Let us write $Y = X_i + W_i$, where $W_i = \sum_{k=1,k\neq i}^{I} X_k$. We then get

$$\begin{aligned} &P_{X_i|Y,Q_{i,j}}(x_i|Y = y, Q_{i,j} = 1) \\ &\qquad = \frac{\lambda_{i,j} e^{-\lambda_{i,j}x_i} P_{W_i}(y - x_i)}{P_{Y|Q_{i,j}}(y|Q_{i,j} = 1)}. \end{aligned} \tag{32}$$

Since W_i is a sum of mixtures of exponential random variables, by the general expression for the pdf in (27), we have

$$P_{W_i}(y - x_i) = \sum_{k=1,k\neq i}^{I} \sum_{l=1}^{J} \alpha''_{k,l} \lambda_{k,l} e^{-\lambda_{k,l}(y-x_i)} \tag{33}$$

where

$$\alpha''_{k,l} = \pi_{k,l} \prod_{p=1,p\neq i,p\neq k}^{I} \sum_{q=1}^{J} \frac{\pi_{p,q}\lambda_{p,q}}{\lambda_{p,q} - \lambda_{k,l}}. \tag{34}$$

Putting (32) and (33) together, we have (35) shown at the bottom of page. Combining this with (31), we have

$$\begin{aligned} &\mathbb{E}[X_i Q_{i,j}|Y = y] \\ &\quad = \frac{\pi_{i,j}\lambda_{i,j}}{P_Y(y)} \sum_{k=1,k\neq i}^{I} \sum_{l=1}^{J} \frac{\alpha''_{k,l}\lambda_{k,l}}{(\lambda_{i,j} - \lambda_{k,l})^2} \\ &\quad\quad \cdot (e^{-\lambda_{k,l}y} - e^{-\lambda_{i,j}y} - (\lambda_{i,j} - \lambda_{k,l})ye^{-\lambda_{i,j}y}). \end{aligned} \tag{36}$$

In summary, the key advantage of the transform-based approach is that there is no need for numerical integration and that the number of computation steps is a polynomial with small degree, rather than exponential, in I and J.

IV. Computational Experience With the Parametric Models

In this section, we report the numerical experiments with the EM algorithms for the exponential model and the mixture of exponential model. We first summarize our general experience.

The EM algorithm almost always reaches the maximum log-likelihood value very quickly,[1] after less than 100 steps for realistic scenarios, provided that the initial parameters are not very large. We typically draw the initial values for $\pi_{i,j}$ uniformly at random on [0,1], and then rescale them so that, for each i, $\sum_{j=1}^{J} \pi_{i,j} = 1$. The initial $\lambda_{i,j}$'s are drawn uniformly on [0, d], where d is about 1/6 to 1/2 of the maximum value of the $\lambda_{i,j}$'s. For instance, if the true $\lambda_{i,j}$'s are between 0.5 to 6, we can choose $d = 1$. We avoid choosing large initial values for the $\lambda_{i,j}$'s mainly for reasons of numerical precision and rounding errors, since the algorithm repeatedly calculates quantities such as $e^{-\lambda_{i,j}x}$.

However, reaching the maximum log-likelihood value does not imply that the true parameters have been identified. In the exponential case, we typically can identify most λ_i's in each run of the algorithm, each starting from a different initial point in the parameter space. We can always identify the smallest λ_i. By combining experiments from different initial points, we can increase the number of different λ_i's identified. Of course, a technical difficulty exists here: Out of the set of λ_i's after the EM algorithm converges, how do we decide which ones are true model parameters and which ones are not? We have three tentative solutions here. The first is to choose those λ_i's that consistently show up in different runs with different initial conditions. The second is to rely on multiple overlapping routes, for instance, the paths in a tree, as shown in Fig. 1. A λ_i showing in multiple routes not only suggests it is likely a true model parameter, but also indicates where it might reside in the network. A third approach is what we call *Successive EM Algorithm*, which is motivated by the observation that the smallest λ_i is always present in the final result. Let us suppose $\lambda_1 < \cdots < \lambda_I$. Once we have identified λ_1, which we always do, we can keep it fixed in the subsequent runs of the EM algorithm. The hope is that λ_2 can always be identified next. We can repeat the procedure and identify the λ_i's in increasing order. Our preliminary results with this approach are very encouraging.

In the case of the mixture of exponential model, it seems difficult to identify the true parameters. However, the parameters discovered by the EM algorithm allows very accurate fit of the model to the true distribution function for the link attribute, except at the tail. In other words, what EM seems to do is to fit the curve to where most of the mass of the function is, and it does so very quickly. It appears that the EM algorithm mainly tries to curve-fit the observed variable, Y. In the process, it also

$$\begin{aligned} \mathbb{E}[X_i|Y = y, Q_{i,j} = 1] &= \frac{\lambda_{i,j} \sum_{k=1,k\neq i}^{I} \sum_{l=1}^{J} \alpha''_{k,l}\lambda_{k,l} e^{-\lambda_{k,l}y} \int\limits_{0}^{y} x_i e^{-(\lambda_{i,j}-\lambda_{k,l})x_i} dx_i}{P_{Y|Q_{i,j}}(y|Q_{i,j} = 1)} \\ &= \frac{\lambda_{i,j}}{P_{Y|Q_{i,j}}(y|Q_{i,j} = 1)} \sum_{k=1,k\neq i}^{I} \sum_{l=1}^{J} \frac{\alpha''_{k,l}\lambda_{k,l}}{(\lambda_{i,j} - \lambda_{k,l})^2} \\ &\quad \cdot \left(e^{-\lambda_{k,l}y} - e^{-\lambda_{i,j}y} - (\lambda_{i,j} - \lambda_{k,l})ye^{-\lambda_{i,j}y}\right) \end{aligned} \tag{35}$$

[1] Rather, the algorithm almost always reaches the vicinity of the true maximum log-likelihood value.

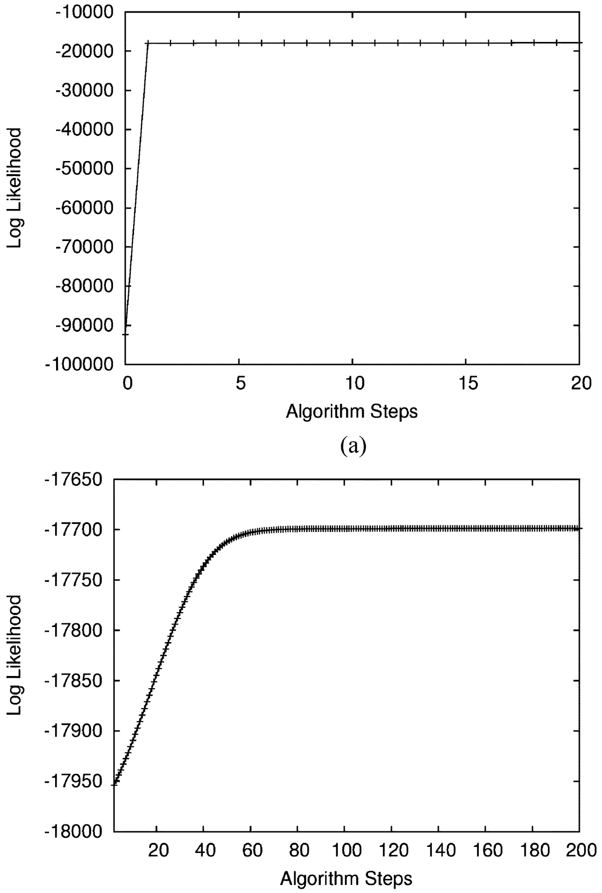


Fig. 2. Log-likelihood values as function of EM algorithm steps for the exponential model. (a) Step 1 to 20. (b) Step 2 to 200.

TABLE I
Exponential Model Parameters and Estimation Results

Links	True λ_i's	λ_i's After 76 Steps	
1	1.00	0.970	1.05
2	1.50	1.59	1.61
3	2.00	2.23	2.01
4	2.50	2.24	2.26
5	3.00	3.63	2.46
6	3.50	4.13	3.63
7	4.00	4.22	4.30
8	4.50	4.46	4.37
9	5.00	4.51	5.65
10	5.50	4.63	5.71
11	6.00	5.27	5.84
12	6.50	5.48	5.88

curve-fits very well to individual link attributes, which is a mysterious process at this point.

A. Exponential Model

In Fig. 2, we show the typical convergence speed of the EM algorithm. In this example, we have 12 links, and their values are shown in Table I. We choose 10 000 samples and the true log-likelihood is -17699.11. Fig. 2(a) shows a giant increase in the log-likelihood after only one step in the EM algorithm. Fig. 2(b) shows that the log-likelihood continues to increase linearly. After 76 steps, its value reaches the vicinity of the true log-likelihood and stays in that region. In short, the EM algorithm has good convergence speed and behavior. Similar observations have been made in the case of the mixture of exponential model.

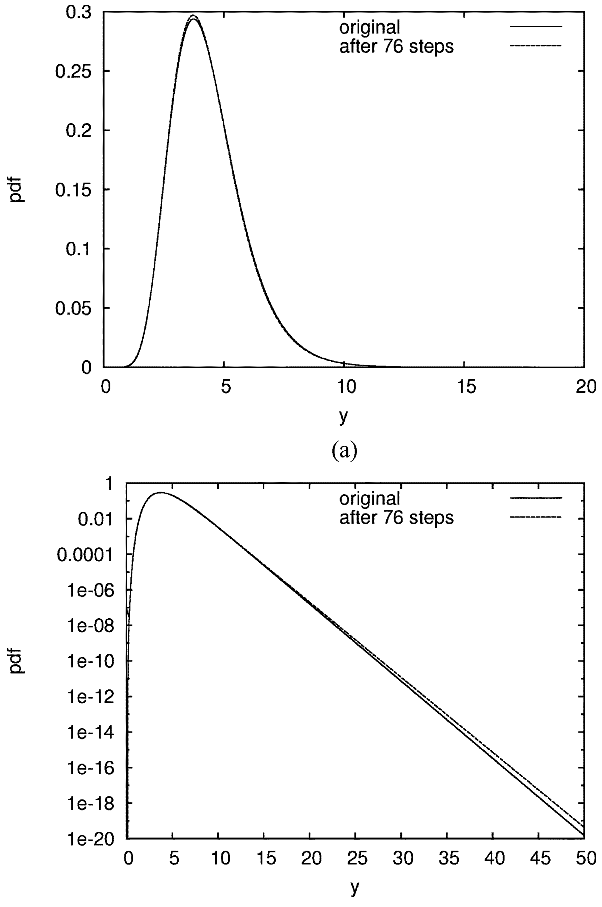


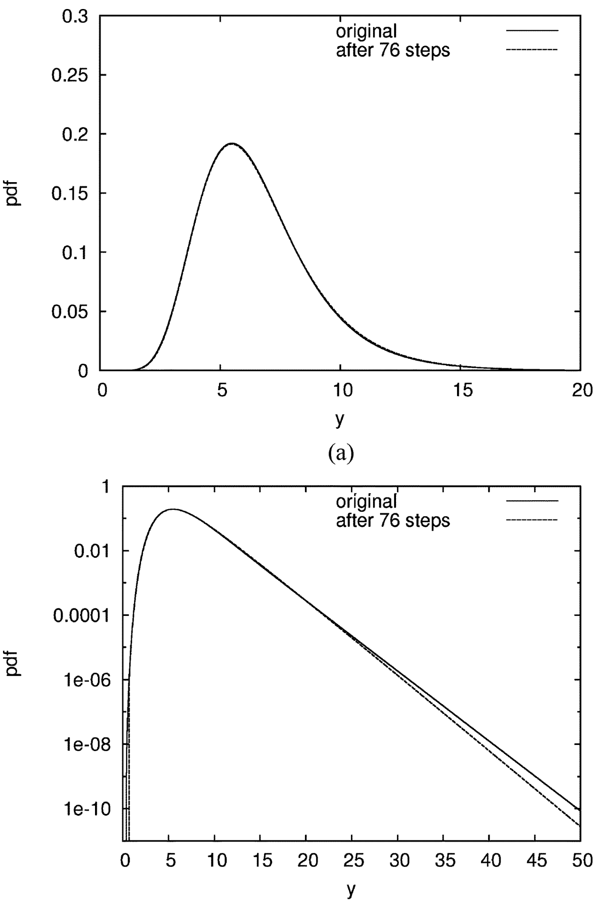
Fig. 3. Original pdf and curve fit by the EM result after 76 EM steps, for the exponential model. (a) Linear-linear scale. (b) Log-linear scale.

Fig. 3 shows the true pdf of the end-to-end observation, Y, and the identified pdf after 76 steps, on linear–linear scale and log-linear scale. EM algorithm works remarkably well as a curve-fitting algorithm. As can be seen from Fig. 3(b), the fit is accurate well into the tail of the curve. The lack of degree-of-freedom in the single-parameter model for the link attribute actually prevents the EM algorithm from creating better fit. We will see that the mixture of exponential model has advantage in curve fitting due to its larger degree-of-freedom, in terms of the number of parameters.

Table I also shows two sets of inferred λ_i's after 76 EM steps, coming from two EM runs with different starting points.

B. Mixture of Exponential Model

In the experiments with the mixture of exponential model, we choose twelve links and each link attribute is a mixture of three exponential random variables. Fig. 4 compares the identified pdf for the observable Y after 76 EM steps and the original pdf. Again, the fit is remarkable. Fig. 5 goes a step further and compares the pdf of the inferred link attribute and that of the true link attribute for several links. Out of the 12 links, we choose a



(a)

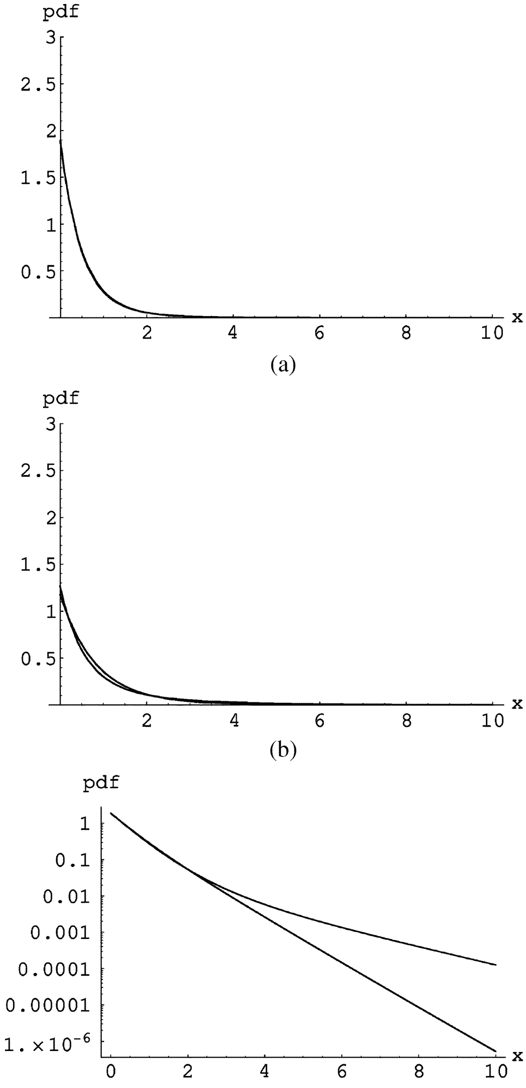
(b)

Fig. 4. Original pdf and curve fit by the EM result after 76 EM steps, for the mixture of exponential model. (a) Linear-linear scale. (b) Log-linear scale.

case of good fit (which is typical), shown in Fig. 5(a), and the worst case, shown in Fig. 5(b). The worst case in fact also has excellent fit. It appears that, in the process of fitting the pdf for Y, the EM algorithm is constrained to fit the pdfs of the individual link attributes as well. However, as indicated by Fig. 5(c), which is in log-linear scale, the fit to the true link attribute is restricted to where the probability concentrates but not the tail. In other words, we have very good fit on a restricted domain of the pdf where the total probability is nearly 1. Even though the inferred parameters do not reflect the true ones, we have excellent statistical information about the original pdf on this restricted domain.

Since the fit between the true and the inferred pdfs is very good and since the probability tail decays exponentially, we expect the fit between the lower order (first, second, third, etc.) moments should be very good as well. Consider an example of a route with three links, where each link attribute is a mixture of three exponential random variables. Table II shows the comparison results for the moments of orders 1, 2, 3, 4, and 10. The true and the inferred values are very close up to the tenth moment.

In the numerical computation aspect, the step of partial fraction expansion, shown in (28) and (29), is a source of serious rounding errors due to the finite precision of the computer. The reason is that the products $\alpha_{i,j}\lambda_{i,j}$ can vary over a large range, say -10^8 to 10^8, and at y near 0, $\sum_i \sum_j \alpha_{i,j}\lambda_{i,j}$ is a number on [0,1]. This creates the most severe type of rounding error called *subtractive cancellation* (see [14]).



(a)

(b)

(c)

Fig. 5. Model and actual link attributes for the mixture of exponential model. (a) Good and typical fit. (b) Not so-good fit. (c) Plot (a) on log-linear scale.

TABLE II
COMPARISON OF MOMENTS

Moments	True Model	Inferred
1	1.75	1.74
2	4.32	4.29
3	14.03	13.92
4	57.32	56.80
10	7.77×10^6	7.50×10^6

V. CONCLUSION

This paper addresses the issue of determining or estimating link attributes based on the observation of end-to-end route attributes. We have investigated the theoretical, algorithmic and computational problems that arise in such an application. We have considered two situations: The link attributes are constants (the deterministic case), or they are random and have distributions from a model family (the parametric model). We have argued that, even if the true link attributes are not from the assumed model family or any other model family, i.e., they must be specified nonparametrically, we can still use a sufficiently expressive family to approximate the true distributions. The mixture of exponential model and the sum of exponential model are particularly suitable for random link delay. The focus of the paper is to develop solutions for the maximum-likelihood estimator based on the EM algorithm. It has been found that, even though the solutions from the EM algorithm do not necessarily reflect the true underlying parameters, they do seem to, in almost all cases, maximize the likelihood, and they generate good fit to the true distribution on the region where the probability mass concentrates.

Our study on the application of the EM algorithm to maximum-likelihood estimation of network internal delay leaves open many interesting issues. For instance, we have yet fully exploited the performance of the Successive EM Algorithm in identifying the true model parameters. A related endeavor is to identify the true model parameters and assign them to their respective links by correlating the measurement results on diverse and overlapping routes, for instance, the tree paths. A multicast-tree based measurement strategy may be helpful. However, difficulties exist in the computation of the pdf and the conditional expectation, given the end-to-end observation. A related idea is to run the EM-based algorithm on the difference of the route attributes and identify a small number of links at a time. Another item on the agenda is to study how well the generated curves fit the link delay distributions that do not belong to the family of models assumed and used in the EM algorithm. From the computation point of view, there is the crucial issue of how to better handle the rounding errors that seem to plague the numerical EM algorithm.

REFERENCES

[1] R. Cáceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Trans. Inf. Theory*, vol. 45, pp. 2462–2480, 1999.

[2] N. Duffield and F. L. Presti, "Multicast inference of packet delay variance at interior network links," in *Proc. INFOCOM*, Tel-Aviv, Israel, Mar. 2000, pp. 1351–1360.

[3] F. L. Presti, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," *IEEE/ACM Trans. Networking*, vol. 10, no. 6, pp. 761–775, Dec. 2002.

[4] A. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc.*, vol. 39, pp. 1–38, 1977.

[5] Y. Vardi, "Network tomography: estimating source-destination traffic intensities from link data," *J. Amer. Stat. Assoc.*, vol. 91, pp. 365–377, Mar. 1996.

[6] J. Cao, D. Davis, S. V. Wiel, and B. Yu, "Time-varying network tomography: router link data," *J. Amer. Stat. Assoc.*, vol. 95, pp. 1063–1075, 2000.

[7] G. Liang and B. Yu, "Maximum pseudo likelihood estimation in network tomography," *IEEE Trans. Signal Process. (Special Issue on Data Networks)*, vol. 51, no. 8, pp. 2043–2053, Aug. 2003.

[8] C. Tebaldi and M. West, "Bayesian inference on network traffic using link count data," *J. Amer. Stat. Assoc.*, vol. 93, pp. 557–576, Jun. 1998.

[9] A. Medina, N. Taft, S. Battacharya, C. Diot, and K. Salamatian, "Traffic matrix estimation: existing techniques compared and new directions," in *Proc. SIGCOMM*, Pittsburgh, PA, Aug. 2002.

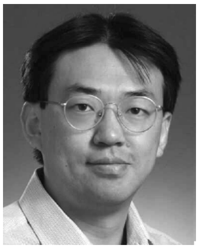
[10] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *Proc. ACM SIGMETRICS*, San Diego, CA, Jun. 2003.

[11] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information theoretic approach to traffic matrix estimation," in *Proc. SIGCOMM*, Karlsruhe, Germany, Aug. 2003.

[12] G. Strang, *Linear Algebra and Its Applications*, 3rd ed. Philadelphia, PA: Saunders, 1998.

[13] D. Starobinski and M. Sidi, "Modeling and analysis of power-tail distributions via classical teletraffic methods," *Queueing Systems*, vol. 36, no. 1–3, pp. 243–267, 2000.

[14] M. J. Maron and R. J. Lopez, *Numerical Analysis—A practical Approach*, 3rd ed. Belmont, CA: Wadsworth, 1991.



Ye Xia received the B.A. degree from Harvard University, Cambridge, MA, in 1993, the M.S. degree from Columbia University, New York, in 1995, and the Ph.D. degree from University of California, Berkeley, in 2003, all in electrical engineering.

From June 1994 to August 1996, he was a Member of Technical Staff at Bell Laboratories, Lucent Technologies, Holmdel, NJ, where he worked in performance evaluation of a shared-memory ATM switch and studied traffic control for ATM networks. Since August 2003 to the present, he has been an Assistant Professor in the Department of Computer and Information Science and Engineering, University of Florida, Gainesville. His primary research interest is in communication networks.

David Tse received the B.A.Sc. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 1989, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1991 and 1994, respectively.

From 1994 to 1995, he was a Postdoctoral Member of Technical Staff at AT&T, Bell Laboratories. Since 1995, he has been with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, where he is currently a Professor. He is a coauthor of *Fundamentals of Wireless Communication*, with P. Viswanath. His research interests are in information theory, wireless communications, and networking.

Dr. Tse received a 1967 NSERC Four-Year Graduate Fellowship from the Government of Canada in 1989, a NSF CAREER Award in 1998, the Best Paper Awards at the INFOCOM 1998 and INFOCOM 2001 Conferences, the Erlang Prize in 2000 from the INFORMS Applied Probability Society, the IEEE Communications and Information Theory Society Joint Paper Award in 2001, and the Information Theory Society Paper Award in 2003. He was the Technical Program Co-Chair of the International Symposium on Information Theory in 2004, and was an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY from 2001 to 2003.