Parameter estimation for MMPPs using the ME algorithm

Haifang Ge, Uli Harder & Peter Harrison[1]

Department of Computing
Imperial College London
Huxley Building
180 Queen's Gate
London SW7 2AZ
England

Abstract

In this paper we use an improved version of the maximum expectation (ME) algorithm to parametrise a modulated Markov Poisson process. We have found improvements to the k-means algorithm in previously published algorithms that combine the EM estimation method with the k-means algorithm. We are able to show that our modified method produces better results for the parameter estimation. In addition We have used our method to parametrise network traffic measured at the departmental (Imperial College) backbone and another standard traffic source. Using a 4 state MMPP the modelled traffic captures the inter arrival time distribution well.

[1] e-mail: hg99@doc.ic.ac.uk, uh@doc.ic.ac.uk, pgh@doc.ic.ac.uk

1 Introduction

It has been known for about a decade now that network traffic is not well described by Poisson models [1]. There are many different ways to improve the modelling of network traffic. One approach that has the advantage of being analytically tractable is the Markov modulated Poisson process, which is a special case of a doubly stochastic process.

There is a vast literature covering various aspects of the use of MMPPs. The seminal paper Neuts [2] introduced the idea in 1971. A good and comprehensive overview of MMPPs can be found in [3].

In an MMPP there is an underlying N-state Markov chain whose state (or phase) changes are described by the instantaneous transition matrix Q. Each state i, $1 \leq i \leq N$ of the Markov chain has got an associated Poisson process with an arrival rate λ_i. These are usually denoted as the diagonal matrix $\Lambda_{ij} = \lambda_i \delta_{ij}$.

Let the state of the MMPP at the time of the k-th arrival be $X_k = X(t_k)$ and $Y_k = t_k - t_{k-1}$ be the inter-arrival time between the $(k-1)$-st and k-th arrival, then an MMPP can be characterized as a Markov renewal process through the bivariate sequence $\{(X_k, Y_k), k \geq 0\}$ [4]. For this process we can derive the transition probability matrix and its density. Following [3] we first define the matrix $H(t)$ with elements $H_{ij}(t)$ as the conditional probabilities $P\{X(t) = j, Y_1 > t | X(0) = i\}$. From the Chapman-Kolmogorov equations we can derive

$$\frac{\mathrm{d}H(t)}{\mathrm{d}t} = (Q - \Lambda)H(t). \tag{1}$$

The solution to this above matrix differential equation is

$$H(t) = \exp[(Q - \Lambda)t]. \tag{2}$$

We further define $F(y)$ as the transition probability matrix for the Markov renewal process, with elements $F_{ij}(y) = P\{X_k = j, Y_k \leq y | X_{k-1} = i\}$, then its transition density matrix $f(y)$ is given by [4]

$$f(y) = \frac{\mathrm{d}F(y)}{\mathrm{d}y} = \exp[(Q - \Lambda)y]\Lambda. \tag{3}$$

In most real world applications the state of the Markov chain is hidden from the observer. This makes fitting parameters for a given a set of arrivals a very complicated problem. In this paper we present a combination of the Maximum Expectation (ME) method and the k-means method to estimate the parameters, i.e. Q and Λ, for a chosen number of states, similar to the work in [5–7].

Another method that can be used to estimate parameters is the for instance moment matching, which compares the moments of the given time series with that of an MMPP [8, 9].

In the next section we describe the EM and k-means method for MMPPs. We then present results comparing our method with those of of Rydén [7] and Deng and Mark [5]. In the last section we present results of our method when it is used to model real network traffic. We use the estimated parameters to run an MMPP simulation and compare the inter arrival time distribution and the power spectrum of the real and the simulated time series.

2 Maximum Likelihood Estimation for MMPPs

Maximum likelihood estimation (MLE) attempts to estimate the parameters Φ of a statistical model needed to describe some data set $\mathcal{Y} = \{y_1, y_2, ..., y_n\}$ with the model. If $p(\mathcal{Y}|\Phi)$ is the density function of the model that depends on parameter Φ, the likelihood of the parameters given the data is defined as

$$\mathcal{L}(\Phi|\mathcal{Y}) = p(\mathcal{Y}|\Phi). \tag{4}$$

The likelihood is thought of as a function of the parameters Φ where the data $\mathcal{Y}$ is fixed. Maximum likelihood estimation is to find the Φ that maximizes $\mathcal{L}$. That is, to find Φ^* where

$$\Phi^* = \text{argmax}_\Phi\, \mathcal{L}(\Phi|\mathcal{Y}). \tag{5}$$

Often $\log(\mathcal{L}(\Phi|\mathcal{Y}))$ is maximised instead because it is analytically simpler.

The Expectation Maximization (EM) algorithm implements maximum likelihood estimation when the data is incomplete or has missing values. The term EM was introduced in [10] where proof of general results about the behaviour of the algorithm was first given as well as a large number of applications.

As before, let us suppose that we have data $\mathcal{Y}$ that is observed. There are also some variables $\mathcal{X}$ that are unobservable. Then we call $\mathcal{Y}$ the *incomplete data* and $\mathcal{Z} = \mathcal{X} \cup \mathcal{Y}$ the *complete data* (or full data). Note that the unobserved $\mathcal{X}$ can be treated as a random variable whose probability distribution depends on the unknown parameters Φ and on the observed data $\mathcal{Y}$. Similarly,$\mathcal{Z}$ is a random variable because it is defined in terms of the random variable $\mathcal{X}$. That is

$$p(\mathbf{z}|\Phi) = p(\mathbf{x}, \mathbf{y}|\Phi) = p(\mathbf{x}|\mathbf{y}, \Phi)p(\mathbf{y}|\Phi).$$

We define

$$\mathcal{L}^c(\Phi|\mathcal{Z}) = \mathcal{L}^c(\Phi|\mathcal{X}, \mathcal{Y}) = p(\mathcal{X}, \mathcal{Y}|\Phi).$$

as the complete-data likelihood. The original likelihood $\mathcal{L}(\Phi|\mathcal{Y})$ is referred to as the incomplete-data likelihood function.

The EM algorithm uses an iterative method to maximise $\mathcal{L}^c(\Phi|\mathcal{Z})$

1. *Estimation (E) step:* Calculate $h(\Phi, \Phi^{(i-1)})$ using the current parameters estimates $\Phi^{(i-1)}$ and the observed data $\mathcal{Y}$ to estimate the probability distribution over the complete data $\mathcal{Z}$

$$h(\Phi, \Phi^{(i-1)}) = E\left[\log p(\mathcal{X}, \mathcal{Y}|\Phi)|\mathcal{Y}, \Phi^{(i-1)}\right]. \tag{6}$$

2. *Maximization (M) step:* Update parameters estimates $\Phi^{(i-1)}$ with the new estimates $\Phi^{(i)}$ that maximizes the h function

$$\Phi^{(i)} = \text{argmax}_\Phi\, h(\Phi, \Phi^{(i-1)}). \tag{7}$$

If the log-likelihood has a single maximum, EM will converge to this global maximum likelihood estimate for Φ. Otherwise, it is guaranteed only to converge to a local maximum. There are many papers on convergence; we refer readers to [10–14].

Expectation Maximisation for MMPPs We follow the EM procedure set out in Rydén [7]. For an MMPP, the set of parameters that governs the model is $\Phi = \{(Q, \Lambda)\}$, where $Q = \{q_{ij}\}$ is the generator of the underlying Markov chain and $\Lambda = diag(\lambda)$ the arrival rate matrix, as we have introduced in the introduction. The data set of the observed sequence of inter-arrival times is denoted by $\mathcal{Y} = \{y_1, y_2, ..., y_n\}$.

According to (4) to define the likelihood of the MMPP parameters given the data we only need to find out the density function of the model. Since Equation (3) has given us the transition density matrix of MMPP, i.e., $f(y|\Phi)$, let π be the initial state probability vector of the underlying CTMC, the MMPP density function can be computed by

$$p(\mathcal{Y}|\Phi) = \pi \prod_{k=1}^{n} f(y_k|\Phi)\mathbf{1}, \quad 1 \leq k \leq n.$$

where $\mathbf{1}$ represents a vector of appropriate dimension comprising 1s in each entry. Now we can have the likelihood of a MMPP with parameter set Φ

$$\mathcal{L}(\Phi|\mathcal{Y}) = p(\mathcal{Y}|\Phi) = \pi \cdot \prod_{k=1}^{n} f(y_k|\Phi) \cdot \mathbf{1}, \quad 1 \leq k \leq n. \tag{8}$$

A straightforward maximisation of the likelihood function $\mathcal{L}(\Phi|\mathcal{Y})$ to estimate parameters Q and Λ is extremely difficult due to the special product form of the Poisson rate and Markov transition rate parameters, as noted by Meier [4]. Furthermore, computation of the derivatives of Equation (3) is not practically feasible at all.

It is, however, possible to simplify the problem by assuming that $\mathcal{Y}$ is incomplete and introduce a set of unobserved data items $\mathcal{X}$. The latter set contains information about the state of the underlying Markov chain.

Suppose X_k is the state of the MMPP at the time of the k-th event, Rydén [15] took the embedded Markov chain $\{X_k\}$ as the missing data. This choice leads to a relatively simple E-step, while the non-explicit M-step has to be carried out numerically.

The serious limitation of Rydén's approach is the difficulty to extend it to MMPPs with more than two phases. Since an MMPP with more than two phases is likely to offer a better model for characterizing network traffic, we followed in our implementation the alternative method, proposed by Asmussen and Nerman [6]. This method can be extended to MMPPs of any order by taking the complete trajectory of the underlying continuous-time Markov chain $\{X(t)\}$ as the missing data. This method leads to a slightly more complicated E-step, but, since the resulting model is almost an exponential family, the M-step is explicit and very simple [7].

In our implementation we followed mostly the algorithm described in [7] and we refer the reader for details to that paper. One problem frequently encountered in estimating MMPP parameters is that the program has to deal with very large and very small numbers, hence a scaling procedure is needed to keep the numerical error small. We used methods described in [16] in our calculations.

2.1 *k*-means algorithm

When applying the EM algorithm, we need to input initial values. Since the EM algorithm is a local minimum algorithm [10], these values must be chosen carefully to minimize the risk of ending up at a local maximum. Here we adopt a method based on the concept of the k-means clustering algorithm [5] to compute the initial values of parameters. Another approach using the EM algorithm was discussed by Rydén and can be found in [7].

As introduced above, the MMPP has a state dependent arrival rate λ_i. For a Poisson process, the inter-arrival times of packets are exponentially distributed with mean $1/\lambda_i$. Therefore the inter-arrival times can be partitioned into separate groups, each for a distinct MMPP state, according to their lengths.

Suppose we have n observed inter-arrival times $\{y_1, y_2, \ldots, y_n\}$, and the number of states of the MMPP is given as r, partitioning the inter-arrival times into N groups can be carried out as follows. Instead of arbitrarily choosing r inter-arrival times as the initial cluster centers, we first compute the cluster boundaries. Define the range of the inter-arrival times as

$$\max(y_k) - \min(y_k), \quad \text{for } 1 \leq k \leq n. \tag{9}$$

The cluster boundaries $B_j, j = 0, 1, \ldots, r$, are obtained by uniformly dividing the range into r segments

$$B_j = \min(y_k) + j \times \frac{\max(y_k) - \min(y_k)}{r}. \tag{10}$$

Each inter arrival time is then assigned to the cluster with suitable boundaries, containing that arrival time. We then update the cluster boundaries, i.e., calculate the mean value of the inter-arrival times for adjacent clusters, and use this mean as new boundary between these two clusters. This process is iterated until no changes is made for each cluster boundary.

Through clustering we associated each of the inter-arrival times with an MMPP state. According to the definitions of the arrival rate λ_i and the transition rate q_{ij}, we have

$$\begin{aligned} \lambda_i &= \frac{\#\text{ of packets arrived during state } i}{\text{total sojourn time at state } i} \\ &\approx \frac{\#\text{ of inter-arrival times in cluster } i}{\text{sum of inter-arrival times in cluster } i} \end{aligned} \tag{11}$$

and

$$\begin{aligned} q_{ij} &= \frac{\text{\# of transitions from state } i \text{ to } j \text{ during state } i}{\text{total sojourn time at state } i} \\ &\approx \frac{\text{\# of occurrence that } y_k \text{ in cluster } i \text{ and } y_{k+1} \text{ in cluster } j}{\text{sum of inter-arrival times in cluster } i} \end{aligned} \quad (12)$$

A summary of execution steps of the algorithm is given below. Note, in practice, we use the *smoothed* inter-arrival data instead of the raw data. The smoothing can eliminate unwanted fluctuations and obtain long-term (stationary) inter-arrival time distribution.

Execution Steps Given the number of states of MMPP r and inter-arrival times sequence containing n objects,

1. Smooth the inter-arrival times $\{y_1, y_2, \ldots, y_n\}$ using a window function of width w

$$\overline{y}_k = \begin{cases} \frac{\sum_{i=1}^{k} y_i}{k} & 1 \le k < w+1 \\ \frac{\sum_{i=k-w}^{k+w} y_i}{2w+1} & w+1 \le k < n-w \\ \frac{\sum_{i=k-w}^{n} y_i}{n-k+w+1} & n-w \le k \le n \end{cases} \quad (13)$$

   The smoothing factor w was chosen empirically, $w = 2$ is suggested in [5].
2. Suppose M and m are the maximum and minimum of smoothed inter-arrival times respectively, compute the initial boundaries of r clusters

$$B_j = m + j \times \frac{M-m}{r}, j = 0, 1, \ldots, r. \quad (14)$$

3. repeat
   (a) (re)assign each smoothed inter-arrival time $\overline{y}_k$ to the cluster $\mathbb{C}_k$ if $B_{j-1} \le \overline{y}_k < B_j$, $j = 1, 2, \ldots, n$.
   (b) update the cluster boundaries

$$B_j = \frac{\sum_{\overline{y}_k \in \{\mathbb{C}_j \cup \mathbb{C}_{j+1}\}} \overline{y}_k}{\|\mathbb{C}_j \cup \mathbb{C}_{j+1}\|}, \quad j = 0, 1, \ldots, r-1. \quad (15)$$

   where $\|\cdot\|$ denotes the cardinality.
   until no change
4. Estimate the MMPP parameters. In [5] the authors suggest to use

$$q_{ij} = \frac{\|\{\overline{y}_k \in \mathbb{C}_i\} \cap \{\overline{y}_{k+1} \in \mathbb{C}_j\}\|}{\sum_{\{\overline{y}_k \in \mathbb{C}_i\} \cap \{\overline{y}_{k+1} \in \mathbb{C}_j\}} \overline{y}_k}$$

as estimator for the transition rates. However, keeping in mind the derivation of equation (12) earlier, a more suitable expression is

$$q_{ij} = \frac{\|\{\overline{y}_k \in \mathbb{C}_i\} \cap \{\overline{y}_{k+1} \in \mathbb{C}_j\}\|}{\sum_{\overline{y}_k \in \mathbb{C}_i} \overline{y}_k} \tag{16}$$

This can be confirmed by numerical experiments. The arrival rates are estimated as

$$\lambda_i = \frac{\|\mathbb{C}_i\|}{\sum_{\overline{y}_k \in \mathbb{C}_i} \overline{y}_k}. \tag{17}$$

3 Implementation and Verification of the algorithm

The estimation procedures were extensively tested on simulated MMPP data. The test is conducted as follows. First, a time series of an n-state MMPP with given parameters (true parameters) is generated by a simulator. This time series serves as the data for the estimation procedure. Then the estimated MMPP parameters (estimated parameter) are compared to the true parameters.

In all numerical results reported in this and the next section, the stopping criterion for the convergence of the EM algorithm was tested by a sufficiently small (0.1%)parameter ϵ, the EM algorithm terminated if

$$\begin{aligned} \frac{|\hat{q}_{ij} - q_{ij}|}{q_{ij}} &\leq \epsilon \qquad \text{for all } i \text{ and } j,\, i \neq j \\ \frac{|\hat{\lambda}_i - \lambda_i|}{\lambda_i} &\leq \epsilon \qquad \text{for all } i \end{aligned} \tag{18}$$

where $\hat{q}_{ij}$ and $\hat{\lambda}_i$ are updated estimates of q_{ij} and λ_i respectively.

To judge the quality of our implementation we compared it to models previously published in [5], [4] and [7].

3.1 Accuracy of the k-means algorithm

We first investigated how our modified k-means algorithm compared to the results published in [5]. The results are presented in table 1. They are all 2-state MMPPs.

The examples were arranged in the order of the degree of difficulty in parameter estimation as the arrival rates from the two states became closer to each other.

Our improved version generally provides better results. However, neither results are very good because of the simplicity of the k-means algorithm. However, we are going to use the results as initial parameters and the closer those are to the real values the better.

In addition we also chose the interrupted Poisson process (IPP). Again our method provides better results than that in [5], see table 2.

3.2 Accuracy of the EM algorithm

MMPPs of Order Two The following four examples are taken from [4, 5] Table 3 lists these different 2-state MMPPs with their true parameters and parameters estimated by different methods. As before, MMPPs were arranged in the order of the degree of difficulty in parameter estimation as the arrival rates from the two states became closer to each other. It is obvious that the pure k-means algorithm performs much worse than the EM algorithms. The combination of our version of the k-means algorithm gives the best results in the majority of the examples above. As the difficulty in parameter estimation increases, i.e. the MMPPs to be estimated have less distinguishable phases, our EM procedure performs much better. For the last and the most difficult example, the procedure of [4] fails to produce results and the procedure developed in [5] yields an relative accurate estimates of arrival rates, but poor estimation for transition rates. In general, the transition rates are more difficult to estimate than arrival intensities [7]. In summary, our EM procedure gives estimation with much higher accuracy.

MMPPs of Order Three We now compare 3-state MMPPs. The results are shown in table 4. Again our method outperforms others in previously examples. These examples are also studied in [5] to show effect of initial values on estimation accuracy. Appropriate initial values could prevent the algorithm ending up at a local minimum. We find from the table, the estimation procedure developed in [5] is more sensitive to the initial value estimation. In contrast, our EM procedure is robust and generates rather accurate estimation even when the difference between the initial and the real values is large.

An interesting point worth mentioning are the figures in the boxes in the middle of the table. The true parameters of arrival rates are 80, 50 and 20, while the estimates that our EM algorithm computed are approximately 50, 80 and 20. These estimates were obtained by setting the initial state probability vector $[0.1, 0.3, 0.6]$. If we change the initial state probability vector to $[0.6, 0.3, 0.1]$, the estimation results become approximately 80, 50 and 20, very close to the true values. For large sample sizes the estimated parameters should not depend on the initial state probabilities. The apparent discrepancy here is explained by looking at the inter arrival time histograms of the 2 MMPPs, see figure 1. The two graphs do look the same. Therefore the MLE approach, estimating the most likely values, cannot distinguish between the two, and the chosen initial becomes important.

MMPPs of Order Four Table 5 and 6 show two examples of parameter estimation for 4-state MMPP (MMPP-4). For the first example, the initial values were chosen arbitrarily. It can be seen that our EM procedure provides in general rather accurate estimates. Since there are no published results on MMPP-4 parameter estimation, we can not do a comparison study with other procedures. The second example in Table 6 is a harder one. The initial parameters were obtained by the k-means algorithm. As can be observed, those initial values are significantly different from the true model parameters. Our EM procedure converged after 310 iterations, and resulting estimates are quite accurate.

Effect of Trace Data Length on Estimation Accuracy In [5] an experiment was designed to examine how the quantity of data available can affect the quality of estimation. As part of our comparative studies, we repeat this experiment. Results are shown in Table 7. Just as [5] we observe that low rate parameters are more sensitive to the simulated trace data length than the high rate parameters. We also note that for a short simulation length, e.g., at 10 seconds, the performance of our estimation procedure deteriorates significantly, especially with regard to transition rate estimates. We presume that this is due to a non-negligible influence of the initial state on likelihood when the sample size is small.

4 Modelling network data

The foremost aim of this work was to model network traffic measured at the departmental core router with an MMPP. We chose to use a 4-state MMPP to do this. The departmental data was collected between between 15.45 and 16.45 on 11 April 2003. For a detailed analysis of similar data see [17]. For computational reasons we selected a trace of filtered data, corresponding to all outgoing non-zero packets from our webserever to estimate the MMPP parameters. We then ran a simulation of an MMPP producing a 1h trace. We had some problems with the convergence of the estimation period. The parameters used here were achieved by more than 300 interations, though the program still hadn't converged fully. Still the parameters hardly changed from step to step. First we compared the inter arrival times of the real and the simulated data by plotting the inter arrival time pdf.

There is fairly good agreement between the simulated and the real data. The difference for small inter arrival times are likely to be caused by measurement errors and the fact that they are governed by the minimum and maximum Ethernet frame size. The MMPP model does not take Ethernet frame size into account. The other difference is that extreme events, i.e. very long inter arrival times are missing. This is due to the restriction of using only a few minutes of traffic for the estimation procedure.

To investigate the level of autocorrelation of the simulated data we computed its power spectrum, c.f figure 2 (see [17] for detailed explanation). The earlier observation that extreme events are missing in the inter arrival times seems to suggest that they are actually important for the shape of the power spectrum. The original network data shows clear signs of a power law and hence self-similarity, whereas the simulated data appears to be white noise.

To investigate this further we used a timeseries of network data which was originally measured and described by Willinger et al. [1]. The data is available as BC-OctExt from the Internet traffic archive [18]. It began at 23:46 on October

3, 1989 and for our traffic modelling purposes we use 3.5 hour of traffic which includes 60,000 packet arrivals. Since this trace has got less events we could use 3.5h worth of events to estimate our parameters. This way we were more likely to observe long range dependence in the simulated data.

Plots show that the simulated shows more of a power law than the network data used before. So, we should expect to get better results for the other network we captured if use more events. However we need to improve the runtime behaviour of our code first to achieve that. For the inter arrival time we similar agreement for the BC-OctExt as we did for the Imperial College data. Again, it misses extreme values.Another observation is that the fitted model tends to generate inter arrivals too small to be realistic. In an Ethernet there is a fixed minimum time between frames, the interframe gap

5 Conclusions

We have shown how the k-means algorithm can be improved to estimate initial parameters for the EM algorithm used for MMPP parameter estimation. Our results are consistently better than those previously published. We have also used the EM algorithm to estimate parameters of a 4 state MMPP using network traffic data measured at the departmental backbone of the Computing Department of Imperial College London [19] and that published by Willinger et al. [20]. We managed to achieve close matches in the inter arrival time distribution. However, the power spectrum of the timeseries created with the estimated parameters show little signs of correlation. We suspect this to be due to the fact that we only used a 4-state model. This also explains the lack of extreme events in the inter arrival time density function. This shows that extreme events are vital for the correlation seen in network traffic. In future work we will investigate the possibility to use the entire trace to estimate the parameters of the MMPP.

Acknowledgements

The authors, in particular Haifang Ge, would like to thank David Thornley for help and suggestion. The research was partly funded by EPSRC (research grant QUAINT, GR/M80826).

References

1. W.E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, Vol. 2 No. 1:1–15, 1994.
2. Marcel F. Neuts. A queue subject to extraneous phase changes. *Advanced Applied Probability*, 3:78–119, 1971.
3. W. Fischer and K. Meier-Hellstern. The markov-modulated poisson process (mmpp) cookbook. *Performance Evaluation*, 18:149–171, 1992.

4. K. S. Meier-Hellstern. A fitting algorithm for markov-modulated poisson processes having two arrival rates. *European Journal of Operational Research*, 29:370–377, 1987.
5. L. Deng and J. W. Mark. Parameter estimation for markov modulated poisson processes via the em algorithm with time discretization. *Telecommunication Systems*, Vol. 1:321–338, 1993.
6. S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distribution via the em algorithm. *Scandinavia Journal of Statistics*, 23:419–441, 1996.
7. T. Rydén. An em algorithm for estimation in markov-modulated poisson processes. *Computational Statistics and Data Analysis*, 21:431–447, 1996.
8. A. T. Andersen and B. F. Nielsen. A markovian approach for modeling packet traffic with long-range dependance. *IEEE Journal on Selected Areas in Communications*, Vol.16, No.5:719–732, 1998.
9. T. Yoshihara, S. Kasahara, and Y. Takahashi. Practical time-scale fitting of self-similar traffic with markov-modulated poisson process. *Telecommunication Systems*, 17:1-2:185–211, 2001.
10. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B, Methodological*, 39:1–38, 1977.
11. M. Jordan and L. Xu. Convergence results for the em approach to mixtures of experts architectures. *Neural Networks*, 8:1409–1431, 1996.
12. R. Redner and H. Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2), 1984.
13. C. F. J. Wu. On convergence properties of the em algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.
14. L. Xu and M. I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8:129–151, 1996.
15. T. Rydén. Parameter estimation for markov modulated poisson processes. *Stochastic Models*, 10:795–829, 1994.
16. S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probability functions of a markov process to automatic speech recognition. *Bell Systems Technical Journal*, Vol. 62 No. 4:1035–1074, 1983.
17. T. Field, U. Harder, and P. G. Harrison. Network traffic behaviour in switched ethernet systems. *Proceedings MASCOTS 2002*, pages 33–42.
18. The Internet traffic archive. http://www.acm.org/sigcom/ita.
19. Trace data measured at Imperial College. http://www.doc.ic.ac.uk/uh/QUAINT/data/Black-Diamond-Departmental-Router/20020322/data.
20. W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: Statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Transactions on Networking*, Vol. 5, No. 1:71–86, 1997.

Appendix: Tables and Figures

	λ_1	λ_2	q_{12}	q_{21}
True parameters	100.0	10.0	10.0	1.00
Results in [5]	86.4	27.4	1.98	0.90
Our results	50.1	10.1	4.76	1.22
True parameters	100.0	20.0	1.00	1.00
Results in [5]	55.2	11.8	0.86	1.78
Our results	109.5	25.6	5.02	3.52
True parameters	100.0	30.0	3.00	2.00
Results in [5]	80.4	12.3	9.56	5.60
Our results	104.78	31.66	7.34	4.16
True parameters	100.0	70.0	1.00	1.00
Results in [5]	56.5	12.6	8.9	0.02
Our results	127.1	59.1	18.9	11.75

Table 1. Accuracy of k-means algorithm for MMPP-2 parameter estimation

	λ_1	λ_2	q_{12}	q_{21}
True parameters	100.0	0	3.0	3.0
Results in [5]	224.9	12.5	1.68	1.20
Our results	105.8	12.2	4.08	2.79

Table 2. Accuracy of k-means algorithm for IPP parameter estimation

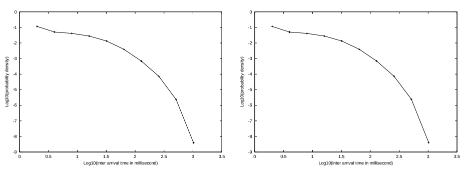

Fig. 1. The inter-arrival time histogram for MMPP-3 with arrival rates 80, 50 and 20 (left) and 50, 80 and 20 (right), both MMPPs have the same transition rates between states.

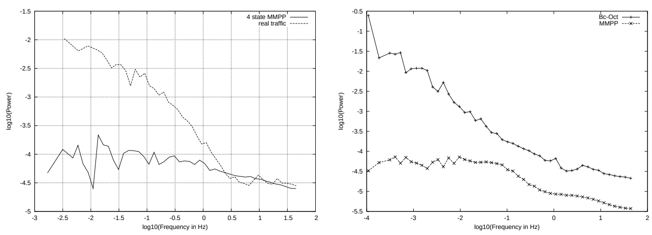

Fig. 2. Left: Power spectrum of packet arrivals for IC traffic and 4-state MMPP. Right: Power spectrum of packet arrivals for BC-OctExt traffic and 4-state MMPP.

	λ_1	λ_2	q_{12}	q_{21}
True parameters	100.0	10.0	10.0	1.00
k-means algorithm	50.1	10.1	4.76	1.22
Estimates by method of [4]	101.4	10.9	7.44	0.67
EM estimates by method of [5]	97.7	10.1	10.2	0.93
Our EM algorithm	99.96	10.0	9.53	0.98
True parameters	100.0	20.0	1.00	1.00
k-means algorithm	109.5	25.6	5.02	3.52
Estimates by method of [4]	96.3	17.9	1.23	0.42
EM estimates by method of [5]	99.4	21.1	0.64	1.00
Our EM algorithm	99.8	19.6	0.97	0.98
True parameters	100.0	30.0	3.00	2.00
k-means algorithm	104.7	31.6	7.34	4.16
Estimates by method of [4]	109.0	36.0	1.36	0.40
EM estimates by method of [5]	99.7	29.7	2.54	2.00
Our EM algorithm	99.7	30.1	2.82	1.95
True parameters	100.0	70.0	1.00	1.00
k-means algorithm	127.1	59.1	18.9	11.75
Estimates by method of [4]	-	-	-	-
EM estimates by method of [5]	94.6	79.4	37.7	0.29
Our EM algorithm	110.5	68.2	1.84	1.17

Table 3. Accuracy of MMPP-2 parameter estimation with different methods

	λ_1	λ_2	λ_3	q_{12}	q_{13}	q_{21}	q_{23}	q_{31}	q_{32}
True parameters	80.0	50.0	20.0	1.00	1.00	1.00	1.00	1.00	1.00
Initial estimates	80.0	50.0	20.0	1.00	1.00	1.00	1.00	1.00	1.00
Estimates by method of [5]	78.0	51.5	19.7	0.93	0.90	0.97	1.15	1.13	1.00
Our EM algorithm	81.2	49.4	20.3	0.91	0.99	0.79	1.00	1.17	0.65
True parameters	80.0	50.0	20.0	1.00	1.00	1.00	1.00	1.00	1.00
Initial estimates	50.0	50.0	10.0	1.00	1.00	1.00	1.00	1.00	1.00
Estimates by method of [5]	59.5	59.5	14.5	1.26	1.26	0.81	1.00	0.81	1.00
Our EM algorithm	49.8	81.1	20.3	0.70	1.03	0.84	0.99	0.65	1.19
	81.1	49.8	20.3	0.84	0.99	0.70	1.03	1.19	0.65
True parameters	80.0	50.0	20.0	1.00	1.00	1.00	1.00	1.00	1.00
Initial estimates	150.0	60.0	30.0	1.00	1.00	1.00	1.00	1.00	1.00
Estimates by method of [5]	125.9	64.4	24.5	1.05	0.30	0.90	0.30	4.22	5.77
Our EM algorithm	81.3	49.6	20.2	0.94	0.95	0.78	1.04	1.18	0.66

Table 4. Accuracy of MMPP-3 parameter estimation with different methods

	λ_1	λ_2	λ_3	λ_4	q_{12}	q_{13}	q_{14}
True parameters	1.0	10.0	50.0	100.0	1.00	1.00	1.00
Initial estimates	2.0	15.0	40.0	150.0	1.00	1.00	1.00
Our EM algorithm	1.2	10.0	51.1	100.4	0.82	0.73	0.94

	q_{21}	q_{23}	q_{24}	q_{31}	q_{32}	q_{34}	q_{41}	q_{42}	q_{43}
True parameters	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Initial estimates	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Our EM algorithm	0.87	1.30	0.99	1.00	0.85	1.22	1.12	0.92	1.06

Table 5. Accuracy of MMPP-4 parameter estimation, Example 1

	λ_1	λ_2	λ_3	λ_4	q_{12}	q_{13}	q_{14}
True parameters	200.0	100.0	50.0	10.0	1.00	1.00	1.00
k-means algorithm	269.1	131.2	63.7	20.1	42.56	2.20	0.49
Our EM algorithm	199.6	99.6	48.9	10.2	0.83	0.65	1.11

	q_{21}	q_{23}	q_{24}	q_{31}	q_{32}	q_{34}	q_{41}	q_{42}	q_{43}
True parameters	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
k-means algorithm	24.49	14.42	0.60	0.97	11.32	4.11	0.18	0.38	3.44
Our EM algorithm	1.07	1.17	0.96	0.65	1.03	1.10	1.01	1.26	0.64

Table 6. Accuracy of MMPP-4 parameter estimation, Example 2

Simulation length	Estimation procedure	λ_1	λ_2	q_{12}	q_{21}
	True parameters	100.0	20.0	1.00	1.00
	Initial estimates	40.0	8.0	0.6	2.4
80 seconds	EM estimates by method of [5]	96.6	19.8	0.77	0.89
	Our EM algorithm	100.9	19.4	1.08	1.00
50 seconds	EM estimates by method of [5]	96.4	19.6	0.80	0.88
	Our EM algorithm	100.6	19.6	0.86	0.81
25 seconds	EM estimates by method of [5]	96.1	21.6	0.83	1.13
	Our EM algorithm	98.1	18.6	0.64	0.94
10 seconds	EM estimates by method of [5]	93.6	21.4	0.43	0.49
	Our EM algorithm	95.9	18.2	0.17	0.42

Table 7. Effect of Trace Data Length on Estimation Accuracy