

4th KDD Workshop on Temporal Data Mining: Network Reconstruction from Dynamic Data

Loews Philadelphia Hotel
Philadelphia, PA
Aug 20, 2006, 8:30am-12:30pm

Much of data contained in large databases has explicit or implicit temporal information. Over the past decade, many powerful data mining techniques have been developed to analyze temporal and sequential data. TDM'06 continues in the tradition of previous temporal data mining workshops at KDD but will focus on a specific topic: "What can we infer about the structure of a complex dynamical system from observed temporal data?" Properties that may be inferred include hierarchy, topology, sign (+/-), order, lag, lead, and strength of influences. The aim of this workshop is to critically evaluate the need in this area and identify promising technologies and methodologies for doing the same.

Highlights:

- Highly interactive format focused toward defining the field of network reconstruction.
- Four invited speakers covering various aspects of network reconstruction in diverse domains from systems biology to signal processing: Bud Mishra (NYU), C. Lee Giles (Penn State University), Vijay Nair (University of Michigan), and Vinod Sharma (Indian Institute of Science, Bangalore).
- Release of a challenge dataset for network reconstruction problems from multi-electrode neuronal data, with the goal of discovering the "neural code". Participants are encouraged to study/analyze the dataset from the workshop website and contribute to the ongoing discussion.
- Two contributed papers: on reconstructing (partial) order constraints from sequential data and on generalized temporal model reconstruction from discrete symbol sequences.
- Discussions toward establishing a collaborative industry-university effort in network reconstruction, especially aimed at adapting algorithms developed in diverse contexts toward network reconstruction goals.

Organizers:

K.P. Unnikrishnan (General Motors), Naren Ramakrishnan (Virginia Tech), P.S. Sastry (Indian Institute of Science), and R. Uthurusamy (General Motors).

For further details including detailed workshop agenda and public-domain dataset, visit the workshop webpage <http://people.cs.vt.edu/~ramakris/kddtdm06/>

Workshop Agenda

The workshop includes a collection of invited talks, short contributions by the organizers as background work, and a discussion session aimed at defining the field of network reconstruction, release of a challenge dataset that embodies multiple facets of network reconstruction, and proposals for collaborative efforts aimed at adapting algorithms developed in diverse contexts toward network reconstruction goals.

8:30 Welcome and introduction

Network reconstruction from dynamic data
Workshop organizers

8:40 Invited talks

Computer and communications networks: assessing and monitoring quality of service
[Vijay Nair](#) (University of Michigan)

Estimating traffic intensities in a communication network via active network tomography
[Vinod Sharma](#) (Indian Institute of Science)

Remembrance of experiments past: analyzing time course datasets to discover complex temporal invariants
[Bhubaneshwar Mishra](#) (NYU)

Probabilistic models for discovering temporal semantic social networks
[C. Lee Giles](#) (Penn State)

10:00 Break

10:30 Short papers by organizers

[Reconstructing partial orders from linear extensions](#)

P.L. Fernandez, L.S. Heath, N. Ramakrishnan, and J.P. Vergara

[Discovering network patterns in micro-electrode array data](#)

D. Patnaik, P.S. Sastry, and K.P. Unnikrishnan

10:50 Panel Discussion by Invited Speakers

"Network" Neuroscience
Grand challenge datasets

11:30 Discussion by all participants

Establishing the field
Proposals for collaborative efforts

12:30 Adjourn

Abstracts of talks

1. Computer and Communications Networks: Assessing and Monitoring Quality of Service

Vijay Nair
Department of Statistics
Department of Industrial & Operations Engineering
University of Michigan, Ann Arbor

There are interesting challenges in collecting and analyzing data from computer and communication networks for the purpose of assessing and monitoring quality of service characteristics. We will briefly describe two classes of network tomography problems and some research on network monitoring. This is joint work with George Michailidis, Earl Lawrence, Bowei Xi, and Xiaodong Yang.

2. Estimating traffic intensities in a communication network via active network tomography.

Vinod Sharma
Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore

We consider a product-form queuing network. A probing traffic stream is sent through the various nodes of the network to measure the congestion in the network due to other traffic streams. By measuring the total number of packets (customers) of the probing stream in the network at different times we estimate the arrival intensity of the other streams and the service rates of the nodes the probing stream enters. This work has applications in providing QoS routing in the communication networks.

3. Remembrance of Experiments Past: Analyzing Time Course Datasets to Discover Complex Temporal Invariants

Bud Mishra
Courant Institute of Mathematical Sciences
New York University, New York, NY

Current microarray data analysis techniques draw the biologist's attention to targeted sets of genes but do not otherwise present global and dynamic perspectives (e.g., invariants) inferred collectively over a dataset. Such perspectives are important in order to obtain a process-level understanding of the underlying cellular machinery; especially how cells react, respond, and recover from environmental changes.

We have devised, GOALIE (Gene-Ontology for Algorithmic Logic and Invariant Extractor), a novel computational approach and software system that uncovers formal temporal logic models of biological processes from time course microarray datasets. GOALIE `re-describes' data into the vocabulary of biological processes and then pieces together these re-descriptions into a Kripke-structure model, where possible worlds encode transcriptional states and are connected to future possible worlds. An HKM (Hidden Kripke Model) constructed in this manner then supports various query, inference, and comparative assessment tasks, besides providing descriptive process-level summaries.

4. Probabilistic Models for Discovering Temporal Semantic Social Networks

C. Lee Giles

School of Information Sciences and Technology³
Pennsylvania State University, University Park, PA

The increasing amount of communication between individuals in e-formats (e.g. email, Instant messaging and the Web) has motivated computational research in social network analysis (SNA). Previous work in SNA has emphasized the social network (SN) topology measured by communication frequencies while ignoring the semantic information in SNs. In this paper, we propose two generative Bayesian models for semantic community discovery in SNs, combining probabilistic modeling with community detection in SNs. To simulate the generative models, an EnFGibbs sampling algorithm is proposed to address the efficiency and performance problems of traditional methods. Experimental studies on Enron email corpus show that our approach successfully detects the communities of individuals and in addition provides semantic topic descriptions of these communities.

5. Reconstructing Partial Orders from Linear Extensions

Proceso L. Fernandez[†], Lenwood S. Heath^{*}, Naren Ramakrishnan^{*},
and John Paul C. Vergara[†]

[†] Dept. of Information Systems and Computer Science,
Ateneo de Manila University, Quezon City, Philippines

^{*} Department of Computer Science, Virginia Tech, Blacksburg VA

Reconstructing system dynamics from sequential data traces is an important algorithmic challenge with applications in computational neuroscience, systems biology, paleontology, and physical plant engineering. Here, we formalize a key computational task in network reconstruction, namely recovering complex order-theoretic constraints among the system variables underlying a given dataset. Specifically, we focus on the problem of reconstructing partial orders (posets) from their linear extensions. We discuss the theoretical complexity of this

problem, a general framework to pose and study various inference tasks, and sketch algorithmic results for mining restricted classes of posets.

6. Discovering Network Patterns in Microelectrode Array Data

D. Patnaik†, P.S. Sastry†, and K.P. Unnikrishnan*

†Dept. of Electrical Engineering, Indian Institute of Science Bangalore India

*General Motors R&D Center, Warren MI

Microelectrode array(MEA) recording is a relatively new experimental technique in neurobiology for studying simultaneous activity of groups of neurons. The objective of analyzing the MEA recordings is to discover different types of temporal correlations between the neurons in an ensemble and hence infer the functional connectivity of the neural tissue. To discover such relationships from multi-neuronal data, there is a need for analysis techniques which are efficient and which can unearth interesting regularities that involve more than pairs of neurons. In this article, a novel application of frequent episode discovery framework to microelectrode array data analysis is presented. In this article it is shown, through simulations, that by combining discovery of different types of episodes with suitable temporal constraints, one can discover the network structures and connectivity patterns of the neurons constituting the network.

Explanation of the Data files

The synthetic data is intended to resemble spike sorted, simultaneously recorded, multi-neuronal data. Analysis of these spike trains can throw light on the functioning of the neural tissue. If we label each neuron and then organize these spikes into a single time-ordered sequence, then the data would consist of a sequence of tuples of the form (neuron-label, time of spike). Our files contain data in this form. The data is generated by simulating a network of 26 neurons, labeled A - Z. The spiking of a neuron depends on the total input received by the neuron and hence if there are some strong connections among a set of neurons, then, the spike trains would contain some temporal patterns of spiking by these neurons which repeat many times along the data sequence. We use a simulator where we can embed different patterns by appropriately choosing values for interconnection weights. This is how the different data files are generated.

The data files provided are for five different patterns. There are four files for each pattern, thus making up a total of twenty files. The file name shows the pattern number and the noise probability. We have used two different noise levels: 0.1 and 0.25. For each pattern and noise level, we have two files. These are two random realizations of the firing pattern of that network. Each data file contains 100000 firings with each line indicating name of the neuron and time of firing.

There are twenty six neurons in the network, each labeled A – Z. The neurons are first randomly interconnected with the weight of the connection being uniformly distributed in the interval $[-1, 1]$. After that, one or more patterns are introduced into the network by modifying the appropriate interconnection weights. Each pattern is a directed acyclic graph with nodes labeled by alphabets. Thus a pattern is introduced by changing all weights of connections into each node of the graph. For example, if a node 'C' in the graph has two incoming edges, say, from 'A' and 'B', then the weights are set so that when both 'A' and 'B' fire then there is sufficient input to make 'C' fire. The model is run in a simple discrete event simulation mode to generate the output spike train. Whenever a neuron fires, its name and the current time is written onto the data file. When a neuron fires, it stays ON for 40 units of time and then can not fire again for the next 20 units of time (which captures the refractory period). In addition, all the neurons to which this

neuron feeds its output are scheduled for consideration at a random time uniformly distributed in the interval [20, 40]. This captures single synaptic propagation delay. All neurons scheduled for consideration are maintained on a queue sorted according to time. As the simulation time advances, we pick appropriate neurons from this queue, find their total input (based on the current state of all neurons) and if it exceeds a threshold we fire that neuron. Even if the input does not exceed the threshold, the neuron is fired with some probability, which we call the noise probability. (As stated earlier, firing a neuron involves writing a spike in the output file and scheduling all the down-stream neurons onto the queue). In addition, for each instant in the simulation time we put a randomly selected neuron on the queue. Since a neuron can fire with some probability even in the absence of any input, this will ensure some general background firing by all neurons. In selecting neurons like this, all neurons corresponding to the 'first' nodes of the patterns have a higher probability of being selected so that the output is biased in favor of repetitions of the embedded pattern.

Essentially, the data is in a form where the 'frequent-patterns' idea of data-mining is useful. For example, if there is a path 'C--D--E in the graph, we can expect many repetitions of 'D firing within 20 to 40 time units of firing of C and D is followed by firing of E within 20 to 40 time units'. Of course, not every occurrence of C in the data file would be a part of such a firing sequence because there is noise. Similarly, there may be other neurons firing in between neurons of such a sequence since all neurons are also firing randomly.

All five patterns are directed acyclic graphs. Pattern-5 is a single long sequence of nodes (sometimes called a synfire chain). This is probably the simplest pattern to discover. Pattern-1 is a simple graph of six nodes. Pattern-2 consists of two such graphs. Pattern-3 is a graph of 11 nodes in which there are many paths that are seven nodes long. In all these graphs, the maximum degree of a node is three. Pattern-4 is a binary tree having four levels. (In each data file, the node names in the embedded patterns are easily obtained by looking at the histogram of the number of firings by each neuron). In our opinion, Pattern-5 and Pattern-1 are the simplest and Pattern-2 and Pattern-3 are the toughest.

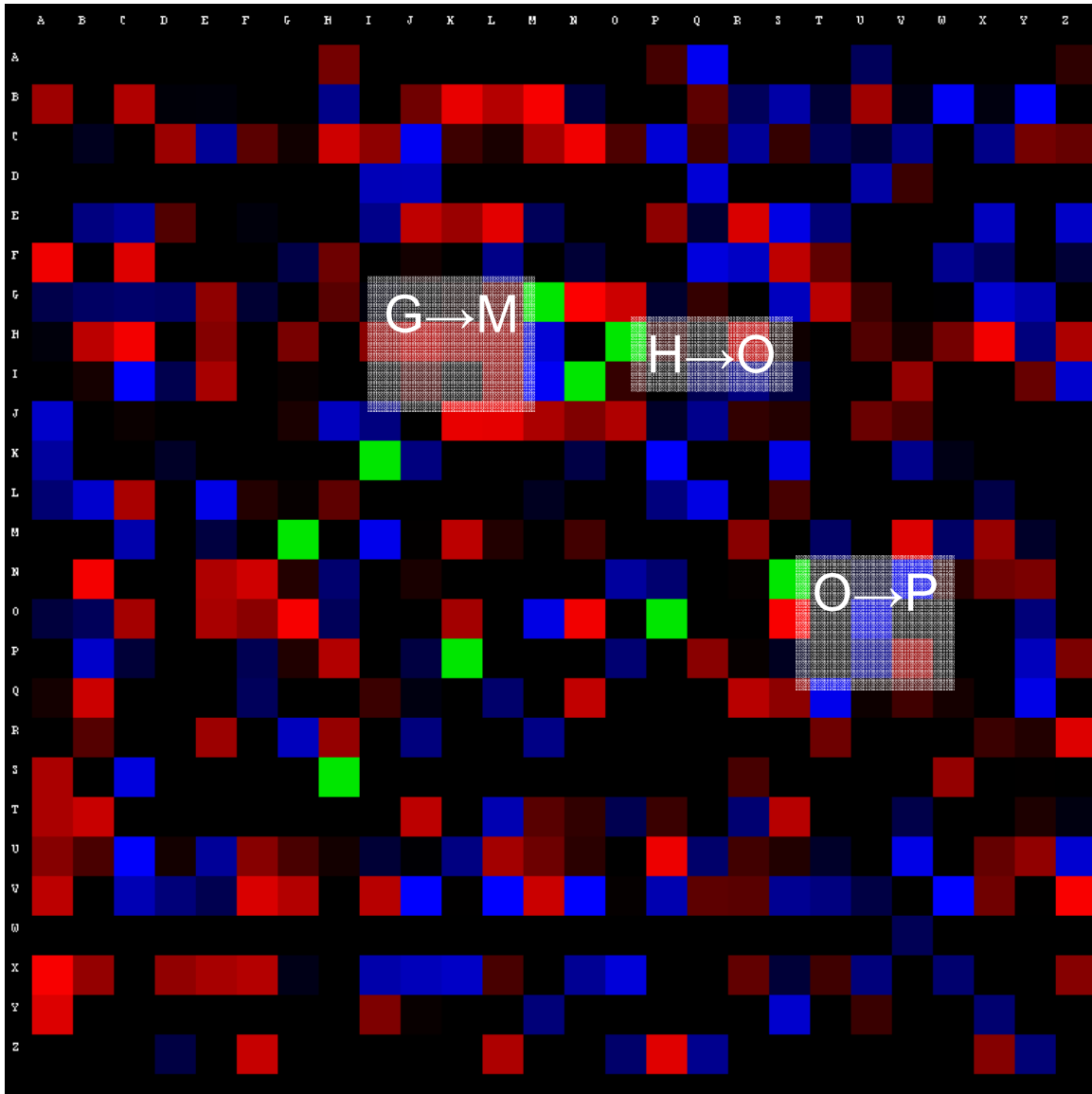


Figure 1. Connection matrix for generating the sequences G-M and H-O-P-K-I-N-S. Row neurons are connected to column neurons. We illustrate some of the connections. Red connections are -ve, Blue are +ve, and Green are forced +ve. Intensity shows value.

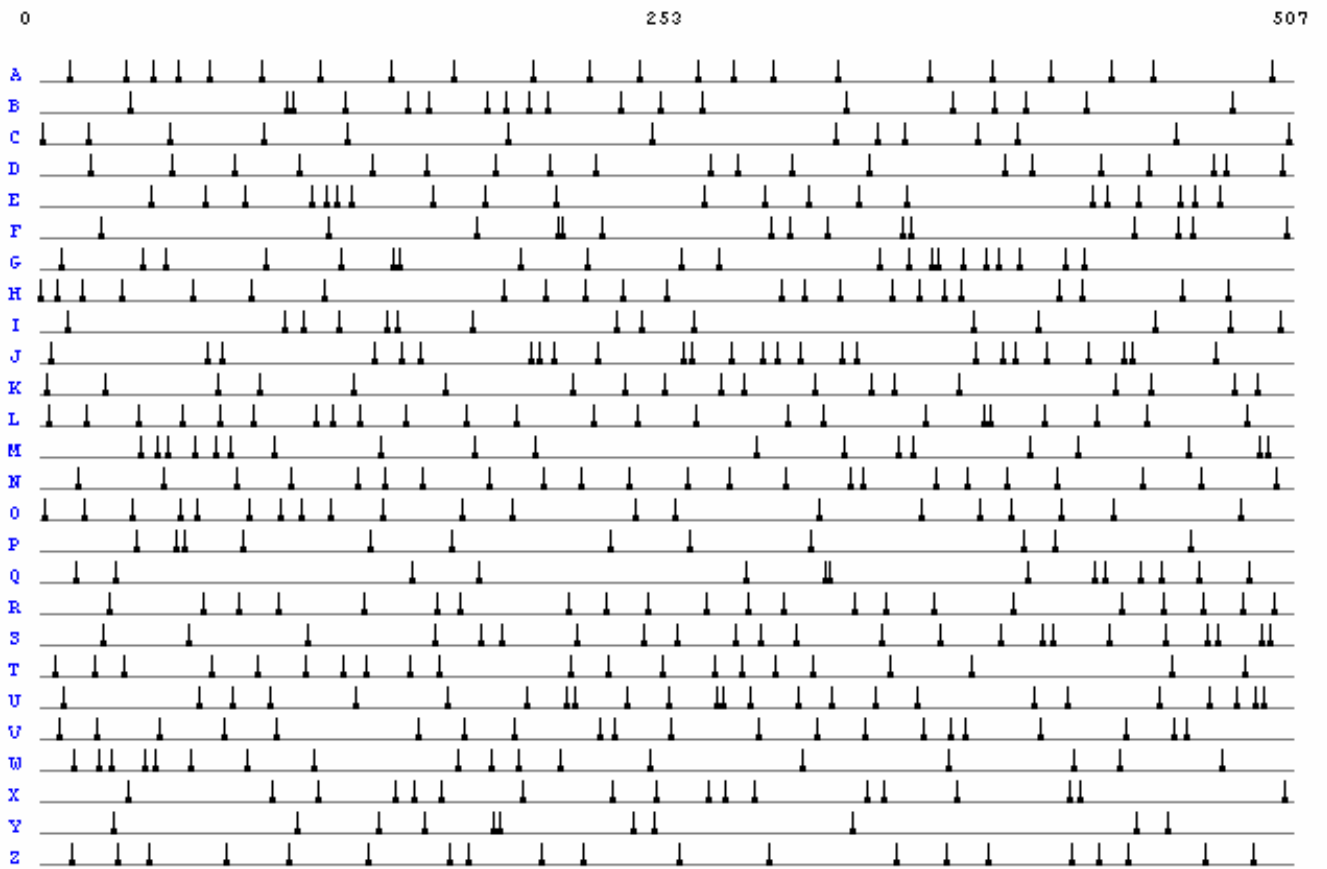


Figure 2. Example spike trains from the 26 neurons

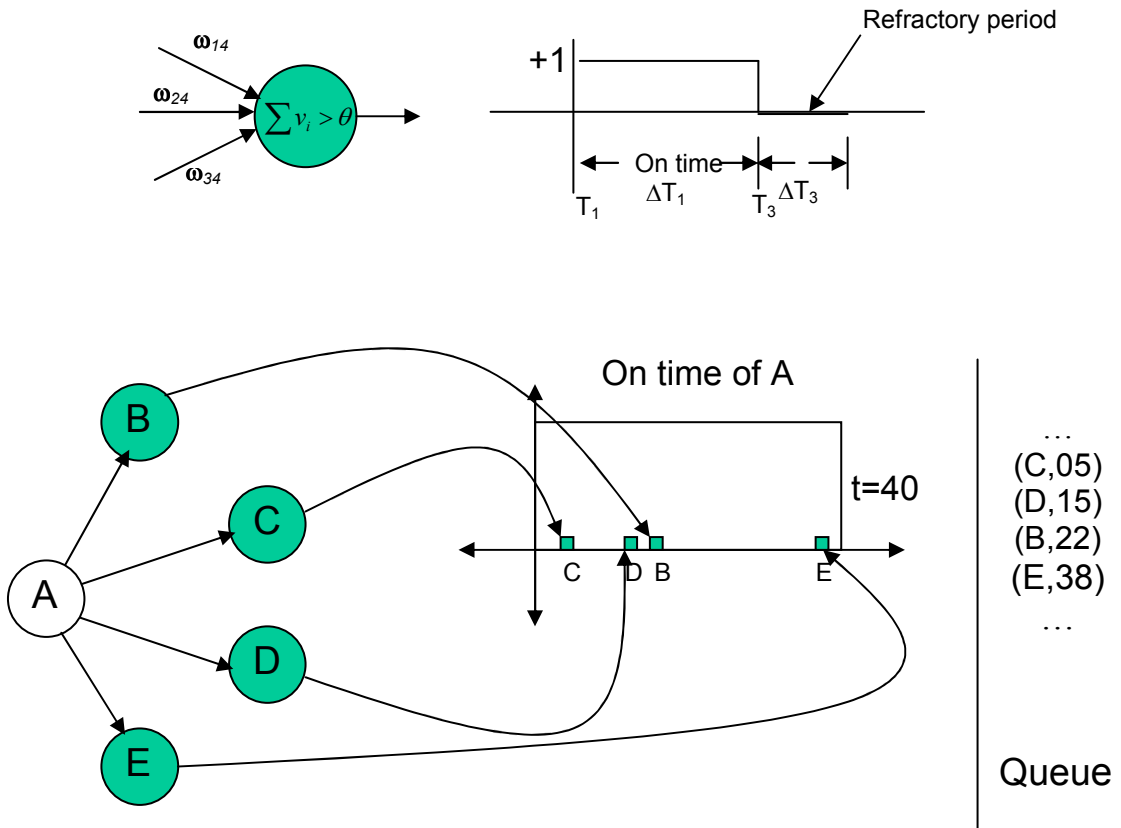


Figure 3. Schematic of the data generation scheme. See text for details. This corresponds to Model-2 in the simulator.