DISSERTATION
submitted to
Combined Faculties for the Natural Sciences and for Mathematics
of the
Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Put forward by

MSc: Navid Bazzazzadeh

Born in: Esfahan, Iran

Oral examination:

IDENTIFICATION OF TEMPORAL DYNAMICS IN BIOLOGICAL PROCESSES

Advisor: Prof. Dr. Roland Eils

ABSTRACT

The behavior and dynamics of complex systems are the focus of many research fields. The complexity of such systems comes not only from the number of their elements, but also from the unavoidable emergence of new properties of the system, which are not just a simple summation of the properties of its elements. The behavior of dynamic complex systems relates to a number of well developed models, the majority of which do not incorporate the modularity and the evolutionary dynamics of a system simultaneously. In this work, we deploy a Bayesian model that addresses this issue. Our model has been developed within the Random Finite Set Theory's framework. We introduced the stochastic evolution diagram as a novel mathematical tool to describe the evolutionary dynamics of complex modular systems. It has been shown how it could be used in real world applications. We have extended the idea of Bayesian network for non-stationary dynamic systems by defining a new concept "labeled-edge Bayesian network" and providing a Bayesian Dirichlet (BD) metric as its score function.

ZUSAMMENFASSUNG

Das Verhalten und Dynamik von komplexen Systemen steht im Fokus von vielen Forschungsbereichen. Die Komplexitt solcher Systeme hngt nicht nur mit der Anzahl von deren Elementen zusammen, sondern auch mit der unvermeidbaren Entstehung neuer Eigenschaften des Systems, welche nicht nur einfache Summation der Eigenschaften seiner Elemente sind. Das Verhalten von dynamisch komplexen Systemen steht mit einer Anzahl von hoch entwickelten Modellen in Verbindung, welche zum grten Teil, die Modularitt und die evolutionre dynamik eines Systems nicht gleichzeitig beinhalten knnen. In dieser Arbeit, wir setzen ein Bayesian Model ein, welches sich mit diesem Problem befasst. Unser Model wurde innerhalb des Random Finite Set Theorie's Rahmenwerk entwickelt. Wir stellen die stochastische Evolution Diagram als eine neue mathematisches Werkzeug vor, um die evolutionre Dynamik von komplex modularen Systemen zu beschreiben. Es wurde auch gezeigt wie es in realen Applikationen eingesetzt werden koennte. Wir haben die Idee von Bayesian Netzwerk fr nicht stationre dynamische Systeme durch Definition eines neuen Konzepts ”labeled-edge Bayesian Network” und ein Bayesian Dirichlet (BD) Metric als dessen Auswertungsfunktion erweitert.

ACKNOWLEDGMENTS

First of all, I would like to thank God for all the blessings in my life, most important of which are my family. Their encouragements have provided me with the inspiration and confidence to pursue all my goals.

Thank you to my PhD advisor, Professor Roland Eils, and my research advisor, Dr. Benedikt Brors, for all their kind guidance and support during the course of my research and sharing their time and knowledge with me.

I would like to acknowledge German Cancer Research Center (DKFZ) for the financial support that gave me this opportunity to study and work on my PhD project and a great environment to learn so much and for this I owe them my sincerest gratitude.

I would also like to sincerely thank Dr. Daniel E. Clark, my masters thesis advisor, for all his great comments and help.

I must thank Dr. Sergey Mastitsky, my research mentor and friend, for all his advice that helped me to manage my project and publish my work and also I should thank Mr. Reza Esmaeili Soumeeh for his help for editing this dissertation.

TABLE OF CONTENTS

Appendices:

LIST OF FIGURES

Figure **Page**

List of Algorithms

CHAPTER 1

INTRODUCTION

The structure and dynamics of complex systems are the focus of many research areas, e.g. biology, physics, social science, economics, and engineering [7, 8, 16, 27, 60]. The complexity of such systems originates not only in a large number of their building elements but also in the intricate interplay between their elements that lead to the emergence of novel system properties. The underlying structure of virtually any complex system can be represented as a network.

Interacting with each other, elements (nodes) of large networks form smaller, discrete sub-networks or modules, with dense internal connections among nodes within modules and sparse connections among modules [51]. Importantly, a module can be assigned an identifiable functional role that separates it from other modules [47, 51]. Examples of functional modules in complex systems are many and varied. They include protein complexes and signaling/metabolic pathways [66], sets of genes with common regulatory programs [2, 61], and communities in social networks [22, 38].

Topologies of the large networks representing real complex systems can be considered stationary only during short periods of time. As time goes, systematic rewiring in these networks occurs due to either exogenous stimuli or internal developmental programs [7, 36]. This evolutionary dynamics can manifest itself in transformations such as birth of novel modules or the death of the previously existing one. It can also merge or split the modules.

Understanding a complex system is unfeasible without knowing its structure and behavior over time under different conditions [36]. However, the reverse engineering of the evolving non-stationary networks/systems is a major challenge which has recently been addressed by deploying a host of machine learning techniques. For example, traditional clustering algorithms have been extended to evolutionary clustering [9] based on its smooth cost function [9, 80, 84] or Dirichlet process [4, 6, 19, 70, 79]. Nonlinear non-stationary time series typically generated from observations on complex systems have been tackled by employing hidden Markov models [21] and change point algorithms [18, 56, 82]. Time series analysis has often been coupled with graph partitioning techniques employed to reveal either a sequence of dependent networks [40, 56, 64, 82] or a common network reflecting the structure of shared information through time [26].

To our knowledge, none of the existing methods explicitly account for network modularity and historical dependencies among modules when inferring the evolutionary dynamics of complex systems. In this thesis, we introduce a novel framework that addresses this issue by integrating concepts of the hidden set Markov models and statistical formalism of the theory of random finite sets. Our key assumption within this framework is that modules of a large network can be considered as random sets of nodes. Furthermore, at each time point a network with dynamic topology can itself be considered as a random finite set of such random sets. The dependencies among nodes and their parents in this network are modeled as conditional probabilities, implying that in the course of time old dependencies may disappear and new ones may emerge.

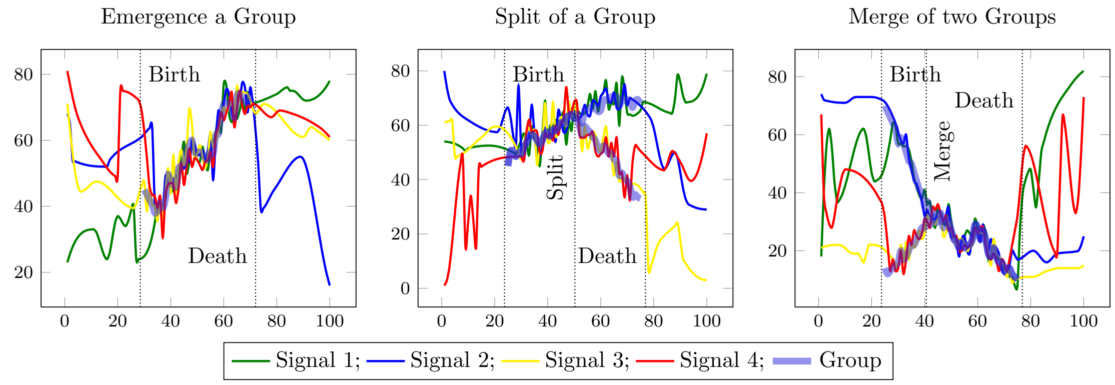


Figure 1.1: Examples of emergence, split and merge of modules over time

Taking multivariate time course data as its input, our method allows one to infer evolutionary dynamics by detecting changes in points of time and the corresponding network modules which have been affected. In other words, our method addresses the following questions that can be of great interest for experimenters (Figure 1.1):

- Is there a network module that emerges at some, a priory unknown, time and exists for a certain period? (Figure 1.1 at left)

- Is there a network module that splits into smaller modules and, if so, how many smaller modules and when? (Figure 1.1 at middle)

- Are there network modules that merge into a bigger module and, if so, when does it happen? (Figure 1.1at right)

Answering these questions is facilitated by analyzing the ”stochastic evolution diagram” a tool that we introduce to encapsulate information about evolving modular systems. Hence, we call our method BASED, which stands for ”BAyesian Stochastic Evolution Diagrams”. In addition to formulating the mathematical essentials of BASED, we validate its plausibility on simulated datasets and demonstrate its performance in real application by using a publicly available microarray timeseries dataset that models TGF-beta-induced epithelial-mesenchymal transition in human lung adenocarcinoma cells [58].

The focus of this research is declared mathematically in chapter 2. Some essential principles required for this thesis are presented in chapter 3. Chapter 4 describes how the characteristics of the declared problem can be formulated in the random finite set framework. Chapter 5 reviews the Bayesian filter as an optimal solution to infer the state of a dynamic system at a point in time. In chapter 6, two approximation methods for the Bayesian filter are reviewed in details. In the chapter 7, a new concept "Stochastic Evolution Diagram (SED)" is defined and have shown how it encapsulates all the information about an evolutionary dynamic system, also it is shown how one can generalizes the idea of HMM to SED. Chapter 8 presents an EM algorithm to identify the SED. Chapter 9 proves how to extend the idea of Bayesian network for non-stationary dynamics system theoretically by using SED. In chapter 10, the accuracy of proposed methods and their applicability in a real world application are tested. Chapter 11 summarizes and discusses about approaches that have addresses problems about evolutionary dynamics. Chapter 11.4 summarizes the work presented in this thesis, and potential future works.

CHAPTER 2

PROBLEM DECLARATION

This chapter is devoted to define precisely the terms complex system, modular network, evolutionary dynamic system and evolutionary modular system. In this research, a complex system is understood as a system with a large number of measurable features (e.g., thousands of gene expression profiles, stock market rates, moving objects, etc.), and whose underlying network contains associations between nodes. Associations between nodes may include a similarity function and any mathematically defined correlation. We are interested in the conditional probability dependencies between features (Bayesian network). The terms ”complex system” and ”underlying network” are exchangeable, similarly applied to any ”random variable” and ”node”. A modular network is a network which is well divided into modules in which there are dense internal connections between nodes within modules but only sparse connections between different modules (Figure 2.1). In many real dynamic complex systems the topology of the underlying network is not static and can evolve over time. In the case of the probability dependency network, a node does not depend on the state of its parents permanently, and it is possible that current dependencies disappear and new dependencies emerge. Figure 2.2 illustrates the evolution of the underlying network's topology within a complex system, where each network temporarily governs

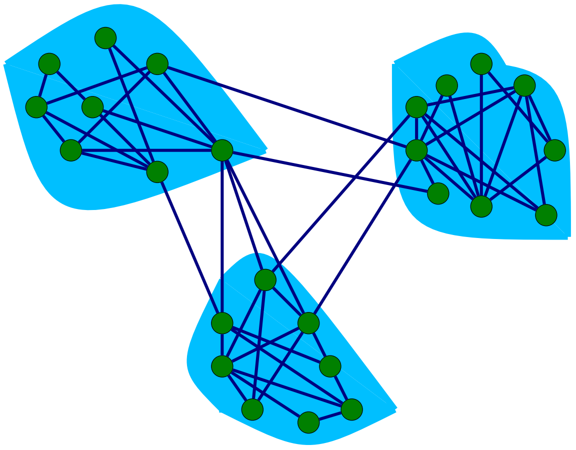


Figure 2.1: An example of a modular network separated into 3 modules.

the dynamic of the system (a network phase). The bottom part of Figure 2.2 illustrates another graphical representation of a time-varying network. In it, the label of each edge indicates its respective lifetime. A system under this scenario undergoes *evolutionary dynamics*.

Assume that a module in such a network is a subset of the set of nodes. Composition of such a module can vary over time and thus at each point in time this module can be viewed as a random set of nodes. Then the network itself can be viewed at each point in time as a random set of random sets. Mathematically, this can be presented as follows. (It should be emphasized that here we consider only nodes, not edges until chapter 9)

Consider a network with n nodes, and let $X = \{x_1, \dots, x_n\}$ be a set of random variables (nodes). The power set 2^X is the set of all subsets of X, including the empty set $\emptyset$. Loosely speaking, we will consider a *random set* as a set variable, whose value is one of the elements of the 2^X. The elements of the power set can be taken in such

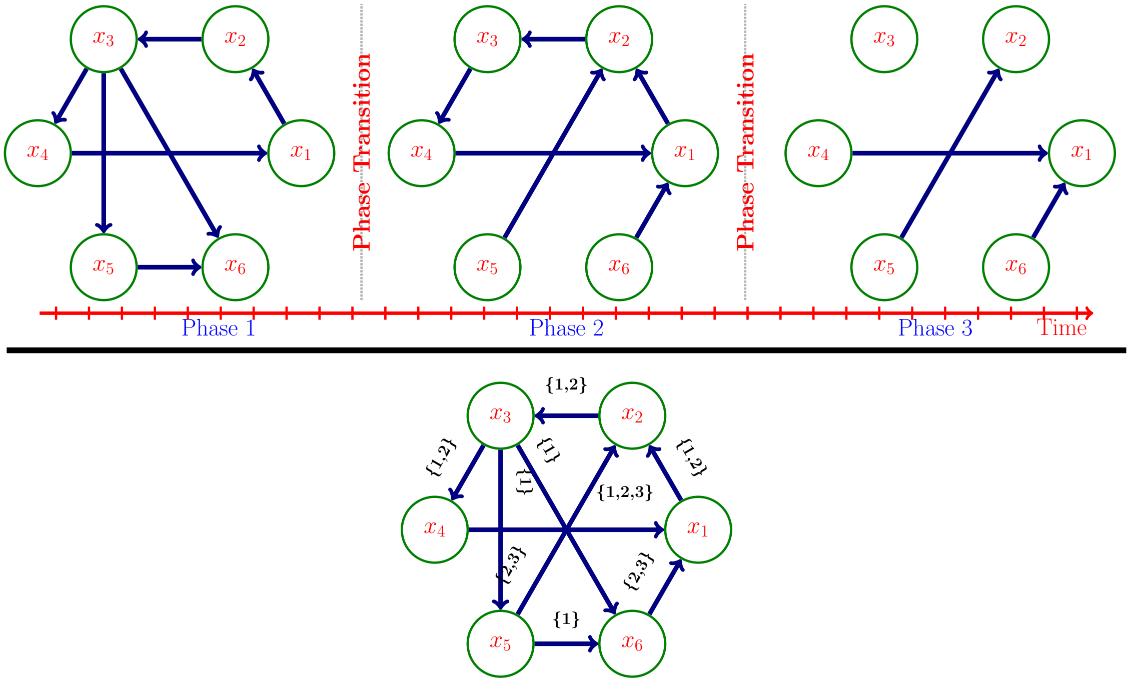


Figure 2.2: An example of two graphical representations of evolution in topology of networks.

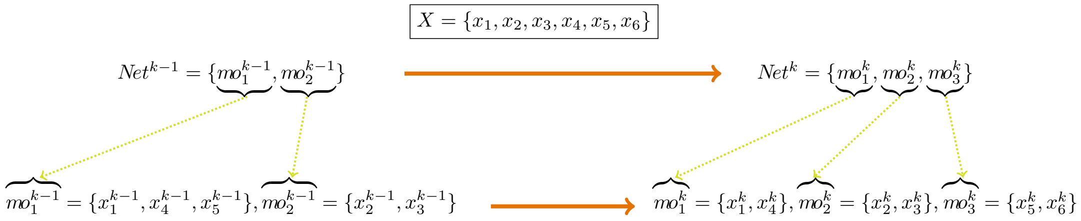

$X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$
$Net^{k-1} = \{mo_1^{k-1}, mo_2^{k-1}\}$
$Net^k = \{mo_1^k, mo_2^k, mo_3^k\}$
$mo_1^{k-1} = \{x_1^{k-1}, x_4^{k-1}, x_5^{k-1}\}, mo_2^{k-1} = \{x_2^{k-1}, x_3^{k-1}\}$
$mo_1^k = \{x_1^k, x_4^k\}, mo_2^k = \{x_2^k, x_3^k\}, mo_3^k = \{x_5^k, x_6^k\}$

Figure 2.3: Changes in topology of a network with 6 nodes represented by random sets.

a way that it will present propositions concerning the actual state of a module in a network at a particular time, by containing *all and only* the nodes in which the proposition is true. Thus, we can represent a module of the network at time k as a finite random set. For example, if at time k, N_m^k nodes belong to the mth module $mo_m^k = \{x_{i_1}, \ldots, x_{i_{N_m^k}}\}$, where $x_{i_j} \in X$, and if there are M^k modules at time k, then the network can be represented as follows:

$$Net^k = \{mo_1^k, \ldots, mo_{M^k}^k\} \tag{2.1}$$

As the topology of the network underlying a complex system evolves in time, the Net^k changes. A change in a random set variable includes changes in size of the set, and the states of its elements. Figure 2.3 illustrates changes in the topology of a network with 6 nodes from time $k-1$ to time k. Thus, if we were able to find the most probable sequence of states of the random finite sets $Net^1, ..., Net^k$, we could understand the dynamics of the complex systems. Briefly, the state of a network at time k shows how many modules exist, and to which module a node belongs. Theoretically an optimal approach toward module detection, tracking, and identification is the following generalization of the recursive Bayesian filter:

$$
\begin{aligned}
p^{k|k-1}(Net^k|Z^{1:k-1}) &= \int f^{k|k-1}(Net^k|net)p^{k-1|k-1}(net|Z^{1:k-1})\delta net && (2.2)\\
p^{k|k}(Net^k|Z^{1:k}) &= \frac{L^k(Z^k|Net^k)p^{k|k-1}(Net^k|Z^{1:k-1})}{\int L^k(Z^k|net)p^{k|k-1}(net|Z^{1:k-1})\delta net} && (2.3)\\
\widehat{Net}^{MAP} &= \arg\max_{net} p^{k|k}(net|Z^{1:k}) && (2.4)\\
\widehat{Net}^{EAP} &= \int net\ p^{k|k}(net|Z^{1:k})\delta net && (2.5)
\end{aligned}
$$

Where Net^k is the network hidden state set. Z^k is the observation set at time k, and $Z^{1:k}$ are all the observation sets from time 1 to k. $p^{k|k}(Net^k|Z^{1:k})$ is the network posterior density function conditioned by the accumulated observation-sets till time k. $p^{k|k-1}(Net^k|Z^{1:k-1})$ is the prediction of the network posterior. $L^k(Z^k|Net^k)$ is the network likelihood function that describes the likelihood of observing Z^k given that the network is in state Net^k. $f^{k|k-1}(Net^k|net)$ is the Markov transition density function that reflects the probability of the network's transition to state Net^k given that it was at state net at time $k-1$.

The network filter Equations 2.2 and 2.3 are applicable if one is able to define effectively the random set value functions $f^{k|k-1}$, $L^k()$ (the Markov transition density function and the likelihood function) as well as the differential and integral calculus for these functions to be able to estimate the network state by means of the expected a posteriori (EAP) or maximum a posteriori (MAP) estimators recursively (Equations 2.4 and 2.5).

The presented network filter cannot be applied like the classical Bayesian filter for a single vector variable in a blind fashion. We require tools of the finite set statistics (FISST) to accommodate set-valued functions, which provide a mathematically consistent and rigorous generalization of the likelihood function and Markov transition function. In the chapters 3, 4, 5, it is reviewed how random set theory provides us FISST to construct these two functions from random set variables. We use a number of analogies between random set statistics and classical statistics (random vector variable) to show the similarities in school of Bayesian thinking in both worlds.

2.1 Motivation and Goals

A systematic understanding of a biological system can be done by study of either the structure of system including interactions and biochemical pathways, or of dynamics of the system including the system's behavior over time under different conditions [36]. In reality, dynamic and structure of a system are dependent on each other. A change in dynamics may yield a new structure or vice versa. Our motivation is to show how to analyze a complex system by considering both dynamics and structure. The hope is that investigation will reveal possible change points of the structure as well as a time table for functionality.

Modularity is considered to be one of the main structural properties of biological systems. A biological network module consists of a set of elements (e.g., genes, proteins) and has distinct function [39]. A biological function can rarely be assigned to an individual element. In contrast, biological functions are carried out by modules made up from interaction among many components, and these functions can not easily be predicted by studying the properties of the isolated components [27, 39, 51, 71]. Indeed, most genes and proteins do not have a function on their own; rather; their role is realized through a complex network of interactions with other proteins, genes and molecules [72]. Over the course of a biological process (e.g., cell cycle), the functionalities of each module are dynamic and context dependent at each time. As a result, it can undergo systematic rewiring, rather than being invariant over time [1]. "Modules can be insulated from or connected to each other. Functional modules need not to be rigid, fixed structures; a given component may belong to different modules at different times" [27].

Evolutionary dynamics (time-evolving topology of underlying network) is the main common dynamical property of most temporal biological processes. The evolutionary behavior of complex networks can be fitted with a number of well developed models [30, 37, 40, 53, 56, 63, 64, 65, 79], but an important challenge (which is still an open question) is how to model temporal large-scale complex networks. As the number of nodes grows in a network, the number of possible network topologies and dimension of network parameters will grow exponentially [79], and computational complexity will be the main challenge.

To our knowledge, none of methods have incorporated explicitly the modularity property to model evolutionary behavior of complex networks. As a result, the investigator cannot fully understand a system. This is particulary true when they are more interested in the global behavior of the nodes in large networks than in the characteristics of an individual node [79] which is frequent case in biological research [1, 17, 27, 36, 39, 51, 61, 71, 72]. We have fused modularity property as a prior knowledge into the introduced evolutionary dynamics model to reduce the computational complexity.

As mentioned before, biological processes can be described as ordered and parallel occurring events. A Gannt chart can be used to illustrate the start and finish time of events and also shows responsible elements of each event. Figure 2.4 depicted an example Gantt chart that it is sorted based on the order of occurring events, and also

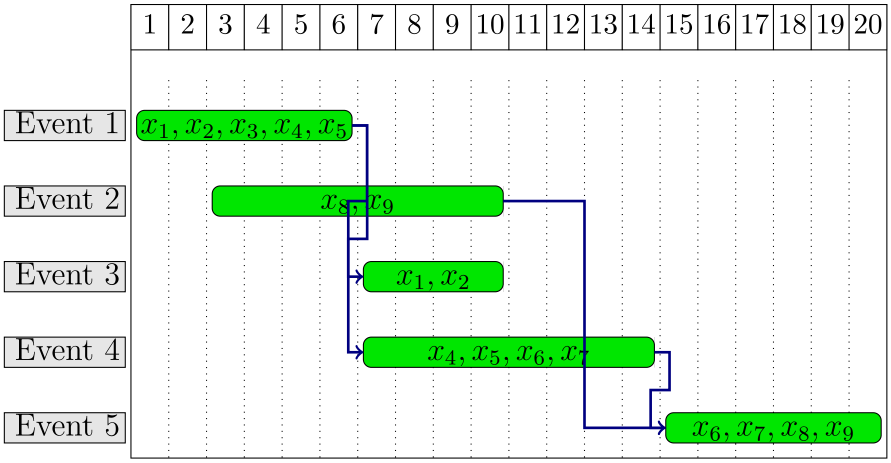


Figure 2.4: A Gantt char example. Events are sorted based on time of occuring

shows the dependencies between events, for example, Event 3 and 4 will be triggered when Event 1 is done, and Event 5 will be triggered when Event 2 and 4 are done. Figure 2.5 shows the same example, but sorted base on dependencies between events.

The biological motivation of this research was to introduce a mathematical framework to detect and identify the occurred events and also reconstruction of interaction of elements responsible for each event. Briefly, we introduced a tool to reconstruct the Gantt chart underlying a dynamical complex system by having a time series observation from the system, and also reconstructing temporal interaction between elements.

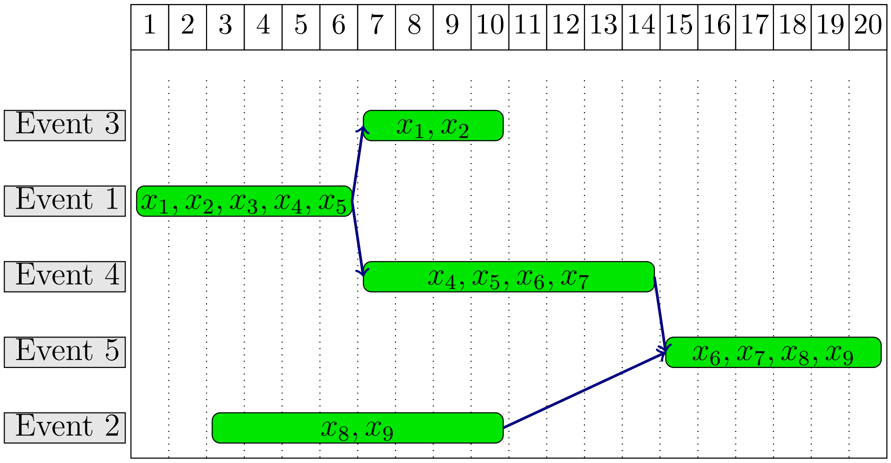


Figure 2.5: A Gantt chart example. Events are sorted based on dependencies

CHAPTER 3

RANDOM FINITE SETS (RFS)

This section briefly describes some of the fundamental statistical concepts of the Random Finite Set (RFS) theory used to formulate our model.

3.1 Almost Parallel Worlds Principles

A rigidly accurate stochastic mathematical foundation for random set-related problems, the point process theory, was formulated several decades ago. However, in 1997 Goodman et al. [24] introduced an "engineering friendly" point process theory called the finite set statistics (**FISST**), which has attracted a great interest thereafter. A detailed description of FISST can be found in [45]. The basic idea of the finite set statistics is to redefine conventional statistical concepts (e.g., derivative, integral, probability mass function, likelihood, etc.) for a random finite set of variables (e.g., set derivatives, set integral, etc.), or, in other words, to define a framework that mathematically transforms the structure of a set of random variables into a bundled single composite variable (random finite set) representing all properties of the characteristics of the original random variables. This transformation results in the creation of a new "almost-parallel world" [45]. The *Almost-Parallel Worlds Principle (APWOP)* states that *almost* any concept or algorithm in conventional statistics can, in principle, be directly translated into a corresponding concept or algorithm in the random set world [45].

3.2 Random Finite Set and Belief Mass Function

Let $U = \{x_1, \ldots, x_n\}$ and $n < \infty$. Then a random variable which takes its value from the universal sample space 2^U is called a random finite set [45]. A measure ($\mathbf{X} : 2^U \rightarrow [0 \quad 1]$) is defined by definition ($\triangleq$) by assigning probabilities $m_x(A) \triangleq P(X = A) = P_X(\{A\})$ directly to each $A \in 2^U$, and the belief mass function for a random set A is defined as $\beta_X(A) \triangleq P(X \subset A) = \sum_{B \subset A} m_x(B)$. The belief mass function plays the same role in random finite set statistics as the cumulative distribution plays in random vector statistics [45].

3.3 Spatio-Temporal Point Processes and Hidden-Set Markov Model

By definition, *module phase* is a period of time when the state of a module (a RFS) does not change, but each elements of the module are allowed to evolve over this period. Then a *module phase transition* is defined as an event when new elements appear or old elements disappear in the module.

Loosely speaking, analogous to a Markov process, a spatio-temporal point process is a memoryless time-varying RFS process, or $p(mo_i^k | mo_i^{k-1}, \ldots, mo_i^1) = p(mo_i^k | mo_i^{k-1})$.

Also analogous to the hidden Markov model (HMM), a hidden-set Markov model (HSMM) is a model in which the system is assumed to undergo a point process with unobserved (hidden) states. The rest of chapter is organized as follows. Section 3.4 describes the formulation of a module's motion model and its observation model. In section 3.5, basic needed principles of finite set statistics (FISST) is reviewed.

3.4 Motion Model and Observation Model of a Module

As mentioned above, both the state of a module and the observation from a module are considered as random finite sets (RFS). In this section, we formulate the RFS motion and observation models of an evolving module by considering the following two examples:

$$\begin{aligned} mo_i^{k-1} &= \{x_1^{k-1}, x_2^{k-1}, x_3^{k-1}\} && \\ mo_i^{k} &= \{x_1^{k}, x_2^{k}, x_4^{k}, x_5^{k}\} && \text{Example.1} \\ \dot{Z}^{k} &= \{\dot{z}_1^{k}, \dot{z}_2^{k}\} && \text{Example.2} \end{aligned}$$

In Example 1, for a given module state mo_i^{k-1} each $x_j \in mo_i^{k-1}$ either continues to survive at time k with a probability $\dot{p}_S^k(x_j^{k-1})$ (e.g., x_1^{k-1} and x_2^{k-1}), or dies with a probability $1 - \dot{p}_S^k(x_j^{k-1})$ (e.g., x_3^{k-1}). In addition, spontaneous birth of new nodes (e.g., x_4^k and x_5^k) can occur at time k with the corresponding probability.

Thus, the cardinality of the module as a RFS, as well as the state of its nodes, are allowed to evolve. Similarly, in Example 2, which considers the observation state model for a given module mo_i^k, each node (e.g. x_1^k and x_2^k) can get detected by the measurement tools (sensors) that produce observations (e.g., $\dot{z}_1^k$ and $\dot{z}_2^k$). The sensor also can fail to detect the nodes (e.g., x_4^k and x_5^k), producing no measurements.

Figure 3.1 illustrates the state space of a network module and observation space, whereas Figure 3.2 illustrates the motion model of a module from time $k-1$ to time k. The module observation model is illustrated in Figure 3.3.

Let a module at time k have a state mo_i^k. Temporal evolution of the modules state, which involves motion of each individual nodes, as well as birth and death of

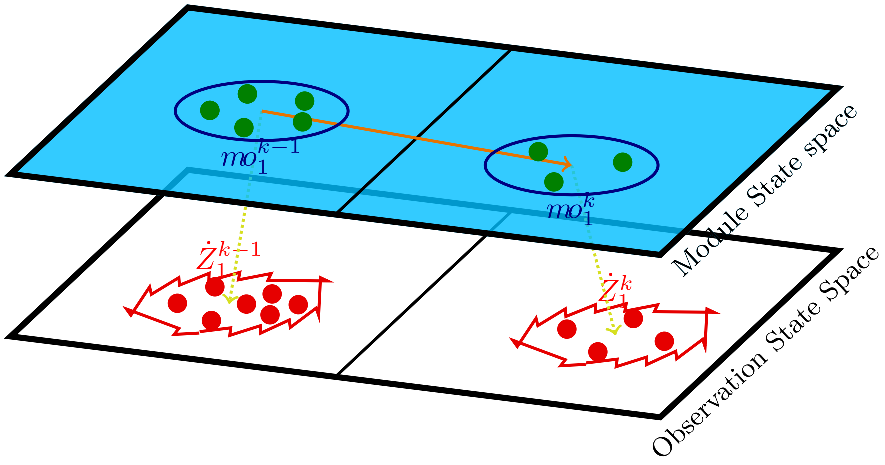


Figure 3.1: Hidden-Set Markov Model: illustration of state space and observation space.

nodes, is formulated as follows:

$$mo_i^k = \left(\bigcup_{x \in mo_i^{k-1}} \dot{S}^{k|k-1}(x) \right) \cup \dot{\Gamma}^k \tag{3.1}$$

where $\dot{S}^{k|k-1}(x_j^{k-1})$ is a RFS model of a node with a previous state x_j^{k-1}, that can take on either $\{x_j^k\}$ or $\emptyset$. $\dot{\Gamma}^k$ is a model for new nodes appearing in the module at time k spontaneously. mo_i^k is, therefore, a union of all the survived nodes and all newly born nodes.

The RFS observation model, which accounts for the detection uncertainty, is formulated as follows:

$$\dot{Z}^k = \bigcup_{x \in mo_i^k} \dot{\Theta}^k(x) \tag{3.2}$$

where $\Theta^k(x)$ is the model for observations that are captured present nodes in the module. This model takes a value $\dot{z}$ if the node is detected and $\emptyset$ otherwise.

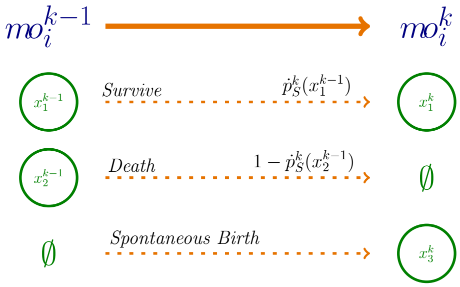


Figure 3.2: Module motion model.

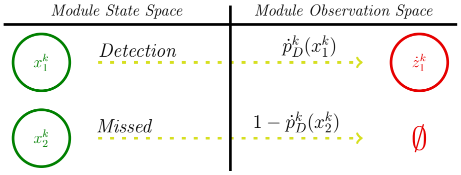


Figure 3.3: Module observation model.

3.5 Basic Ideas of the FISST Calculus

The Mahler's finite set statistics [45] generalizes the Bayesian framework for the study of random set-valued variables, and provides a means to estimate the state of a time sequence of random finite sets that are generated from an assumed point process. In this section, a Bayesian formulation of the module filtering problem is presented in the RFS framework. The RFS framework is a systematic and rigorous approach to RFS filtering [41]. The RFS Bayesian recursion is provided as solution to jointly estimate the cardinality of a module and state of its elements.

Suppose that we recursively observe a dynamic module. The following shows how to derive optimal formulae in a Bayesian sense to estimate a modules' cardinality, and state of module's members. To recall, integral transforms are fundamental to conventional probability, and the ideas of moment generating function and characteristic functions can be used as an alternative route to analytical results compared with working directly with probability density functions or cumulative distribution functions, and recover the nth statical moments of a random variable (e.g. Equations 3.3 and 3.4 for random variable A) from them. Also the probability distribution of a nonnegative integer random variable can be recovered from a *probability generating function* (p.g.f) (Equation 3.5 for nonnegative integer random variable J). Similarly, integral transforms can be generalize into random-set language. The *probability generating functional* (p.g.fl) can be regarded as the generalized probability generating function (p.g.f) to random set statistics.

$$\chi_A(x) \triangleq \int_{-\infty}^{\infty} e^{ixa} \cdot p_A(a)da \qquad = E[e^{ixA}] \tag{3.3}$$

$$M_a(x) \triangleq \int_{-\infty}^{\infty} e^{x \cdot a} \cdot p_A(a)da \qquad = E[x^{A \cdot x}] \tag{3.4}$$

$$G_J(x) \triangleq \sum_{n=0}^{\infty} p_J(n)x^n \qquad = E[x^J] \tag{3.5}$$

A spatial point process is defined as a random set of nodes $mo = \{x_1, x_2, \ldots, x_N\}$ where x_i belongs to a measurable space $\mathbb{S}$ such as $\mathbb{R}^d$ known as state space. Assume probability distribution p_{mo} on measure space $\mathbb{S}$ defines the distribution of the module's members, and a cardinality distribution p_C determines the total number of nodes and satisfies $\sum_{c=0}^{\infty} p_C(c) = 1$.

Let $p_n(y_i)$ be a probability distribution on state space of a node. For any random set $Y = \{y_1, y_2, \ldots, y_n\}$ where $|Y| = n$ and $y_i \in \mathbb{S}$, define the module probability distribution as following:

$$p_{mo}(Y) \triangleq n! \cdot p_C(n) \cdot p_n(y_1) \cdot p_n(y_2) \ldots p_n(y_n) \tag{3.6}$$

If we assign a Poisson distribution with Poisson parameter λ_p to p_C ($p_C = e^{-\lambda_p}\lambda_p^n/n!$), then

$$p_{mo}(Y) \triangleq e^{-\lambda_p} \cdot \lambda_p^n \cdot \prod_{y_i \in Y} p_n(y_i) \tag{3.7}$$

This is called multidimensional Poisson point process. Any module having p_{mo} as its distribution is a RFS Poisson point process.

If $h(x_i)$ is a nonnegative real-valued function of x_i that has not unit of measurement (usually called a test function), let us define the notations h^{mo} , $p_{mo}(\{x_1, x_2, \ldots, x_c\})$ (random set *mo* distribution) and *set integral* as follows:

$$h^{mo} \triangleq \begin{cases} 1 & \text{if } mo = \emptyset \\ \prod_{x_i \in mo} h(x_i) & \text{otherwise} \end{cases} \tag{3.8}$$

$$p_{mo}(\{x_1, x_2, \ldots, x_c\}) \triangleq c! \cdot p(x_1, x_2, \ldots, x_c) \tag{3.9}$$

$$\int f(X)\delta X \triangleq f(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int f(\{x_1, x_2, \cdots, x_n)\})dx_1 dx_2 \cdots dx_n \tag{3.10}$$

Then the *probability generating functional* (p.g.fl.) of *mo*'s probability distribution function is

$$G_{mo}[h] = \mathbb{E}[h^{mo}] \triangleq \int h^Y p_{mo}(Y)\delta Y \tag{3.11}$$

There are two basic properties of p.g.fl. If N_{mo} is the expected value of $|mo|$ (cardinality of module) and if σ_{mo}^2 is its variance, then [45]

$$N_{mo} = G'_{mo}(1) \tag{3.12}$$

$$\sigma_{mo}^2 = G''_{mo}(1) - N_{mo}^2 + N_{mo} \tag{3.13}$$

where $G'_{mo}()$ and $G''_{mo}()$ are the first and second derivatives of $G_{mo}()$.

The other properties of p.g.fl are following:

- $G_{mo}[0] = p_{mo}(\emptyset)$

- $G_{mo}[1] = \int p_{mo}(Y)\delta Y = 1$

- $G_{mo}[h|X] = \int h(Y)p_{mo}(Y|X)\delta Y$ (Conditioning p.g.fl)

A joint probability generating functional of two RFSs X and Y can be defined by

$$G_{X,Y}[g,h] \triangleq \int\int g^X \cdot h^Y \cdot p_{X,Y}(x,y)\delta X \delta Y \tag{3.14}$$

The p.g.fl. of a module as a Poisson point process is given by [45, page. 373]

$$G_{mo}[h] = e^{\lambda_p \int h(y)p(y)dy - \lambda_p} \tag{3.15}$$

3.6 Module Filtering

The optimal module Bayesian filter propagates the module posterior distribution $p^k_{mo}(mo^k|\dot{Z}^{1:k})$ conditioned on the sets of observations up to time k, $\dot{Z}^{1:k}$, with the following recursion via module Bayesian prediction and module Bayesian update [45]

$$p^{k|k-1}_{mo}(mo^k|\dot{Z}^{1:k-1}) = \int f^{k|k-1}_{mo}(mo^k|X)p^{k-1}_{mo}(X|\dot{Z}^{1:k-1})\delta X \tag{3.16}$$

$$p^k_{mo}(mo^k|\dot{Z}^{1:k}) = \frac{p^{k|k-1}_{mo}(mo^k|\dot{Z}^{1:k-1}) \cdot L^k_{mo}(\dot{Z}^k|mo^k)}{\int p^{k|k-1}_{mo}(X|\dot{Z}^{1:k-1}) \cdot L^k_{mo}(\dot{Z}^k|X)\delta X} \tag{3.17}$$

where $f^{k|k-1}_{mo}(.|.)$ and $L^k_{mo}(.|.)$ are the module transition density and module likelihood respectively. The module Bayesian filter alternatively can be written in terms of p.g.fl.s. The p.g.fl. form of the Bayesian filter is given by [45, chapter.14]

$$\begin{aligned}
G_{mo}^{k|k-1}[h|\dot{Z}^{1:k-1}] &= \int G_{f_{mo}^{k|k-1}}[h|X] \cdot p_{mo}^{k-1}(X|\dot{Z}^{1:k-1})\delta X \\
&= \int\int h^{Y} \cdot f^{k|k-1}(Y|X) \cdot p_{mo}^{k|k-1}(X|\dot{Z}^{1:k-1})\delta Y \delta X \qquad (3.18) \\
G_{mo}^{k}[h] &= \frac{\frac{\delta F}{\delta Z^{k}}[0,h]}{\frac{\delta F}{\delta Z^{k}}[0,1]} \qquad (3.19) \\
&= \int \frac{h^{Y} \cdot L_{mo}^{k}(\dot{Z}^{k}|Y) \cdot p_{mo}^{k|k-1}(Y|\dot{Z}^{1:k-1})}{\int L_{mo}^{k}(\dot{Z}^{k}|X) \cdot p_{mo}(X|\dot{Z}^{1:k-1})\delta X}\delta Y \qquad (3.20)
\end{aligned}$$

Here

$$F[l,h] \triangleq \int l^{Y} \cdot G^{k}[l|X] \cdot f_{mo}^{k|k-1}(X)\delta X \tag{3.21}$$

and

$$G^{k}[l|X] \triangleq \int l^{Z} \cdot L_{mo}^{k}(Z|X)\delta Z \tag{3.22}$$

where $G_{mo}^{k|k-1}[h]$ is p.g.fl. Bayesian prediction, and $G_{mo}^{k}[h]$ is p.g.fl Bayesian update.

CHAPTER 4

NETWORK AS A RANDOM FINITE SET OF MODULES

Thus far, we have shown how to characterize the uncertainty of a module in a network by modeling the module state and the module measurements as random finite sets (RFS). We also formulated the corresponding motion model and observation model, and mentioned that a network itself is a random finite set of modules with its own dynamics (Figure 2.3). Therefore, understanding the dynamics of a complex system that has a dynamic topology underlying network is the problem of characterizing the uncertainty of the underlying network, or in other words, detecting, identifying, classifying and estimating (tracking) the states of modules and their nodes at each time point (Figure 4.1).

4.1 Multiple Hidden Set Markov Model

Analogous to the multivariate HMM, the multiple hidden set Markov model jointly characterizes the uncertainty of a random finite set of random finite sets [42]. Mahler [42] has presented a theoretically unified, rigorous, and potentially practical approach to construct an optimal recursive Bayesian estimation for the multiple hidden set Markov models. Figure 4.2 illustrates a two-layered stochastic process of the multiple hidden set Markov model, where the network state-space represents a randomly

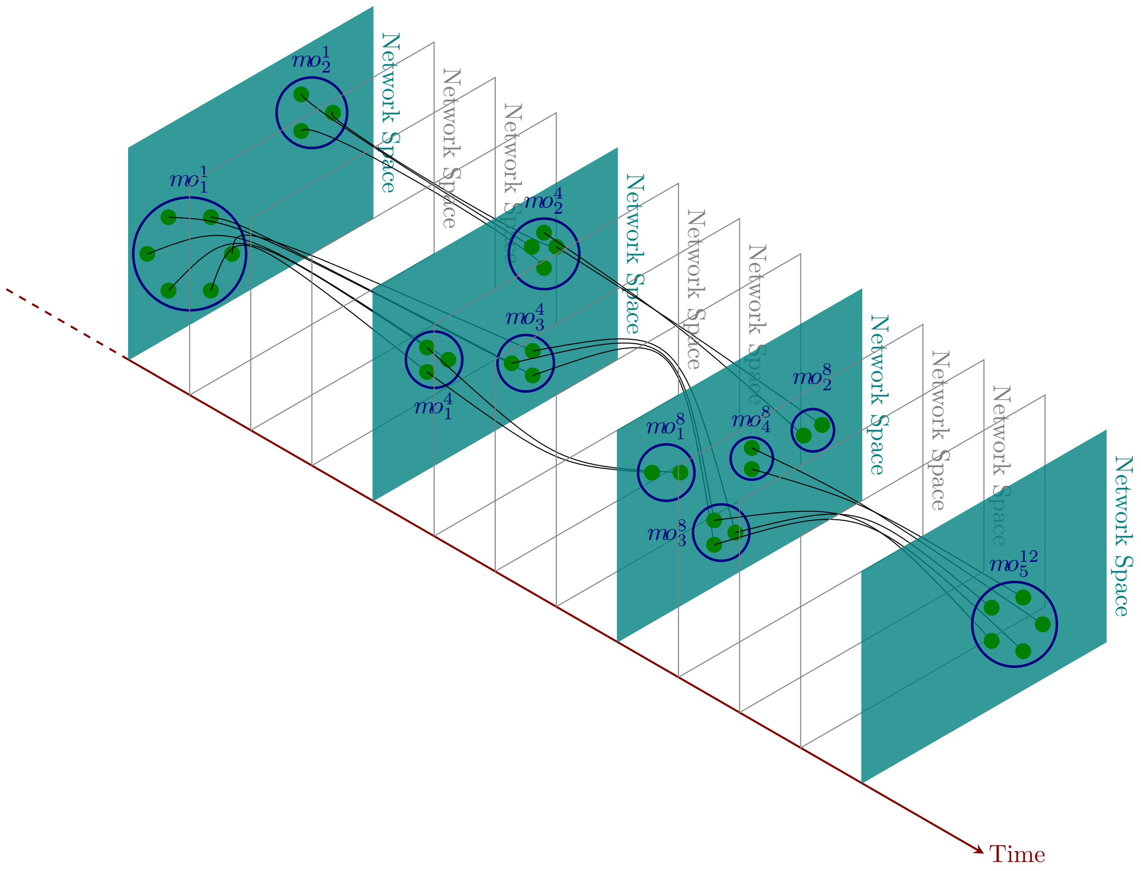


Figure 4.1: Example of a sequence of network states in the Network × Time space. Solid lines are signals that belong to modules.

varying network. The network state space can also have its own motion model (survival, birth and death) and its observation model (detection, missed-detection, and false alarms), which are more complex than a module's dynamic models that were described in chapter 3.4.

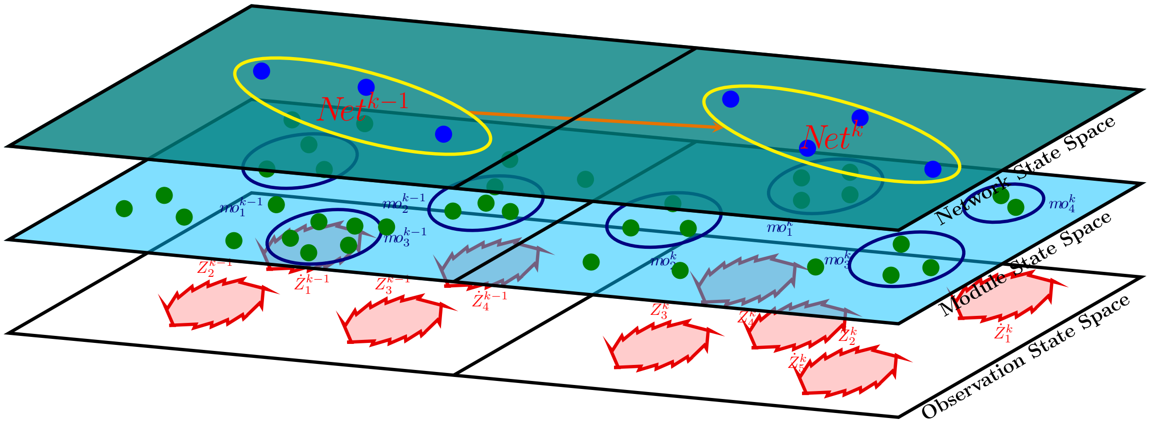


Figure 4.2: Multiple Hidden-Set Markov Model: illustration of network state space, module state space, and observation state space.

4.2 Motion Model and Observation Model of a Network

Within our framework, a network is viewed as a random set of modules that themselves are random set variables, and the same is true for the observations. However, the dynamics of a network is more complex than that of a module. Consider the following two examples:

$$Net^{k-1} = \{mo_1^{k-1}, mo_2^{k-1}, mo_3^{k-1}\}$$

$$Net^k = \{mo_1^k, mo_2^k, x_4^k, mo_5^k\} \qquad \text{Example.3}$$

$$Z^k = \{\dot{Z}_a, \dot{Z}_b, \dot{Z}_c, \dot{Z}_d\} \qquad \text{Example.4}$$

First, let us define the *network phase transition* as an event when new modules appear or old modules disappear in the network. Then the *network phase* is a period of time during which no network phase transition occurs, but each element of the network is still allowed to evolve.

In Example 3, for a given network state Net^{k-1} at time $k-1$ each $mo \in Net^{k-1}$ either continues to survive at time k with a probability $p_S^k(mo^{k-1})$ (e.g., mo_1^{k-1} and mo_2^{k-1}), or dies with a probability $(1-p_S^k(mo^{k-1}))$ (e.g., mo_3^{k-1}). A module dies when all its elements die and this module becomes empty by time k. Also, a new module can appear at time k (e.g., mo_4^k and mo_5^k) with a certain probability of birth. New modules can arise from three different scenarios, i.e. 1) by spontaneous birth, 2) spawning from a module at time $k-1$ (splitting of a module to smaller modules), or 3) merging of some modules to a bigger module.

Similar to the module observation model described in chapter 3, in the network observation model for a given network Net^k at time k each module can be detected by the measurement tools that produce observations. The sensors also can fail to detect the module, producing no measurement. In addition, false alarms in the observation set are possible. The uncertainty of the network observation model will increase when there is no information about association between modules and their observations (Example 4).

Figure 4.2 illustrates the state space of a network and corresponding observation space, whereas Figure 4.3 illustrates the network motion model from time $k-1$ to time k. The network observation model is illustrated in Figure 4.4.

Let a network at time k have a state Net^k. The evolution of the network state, which involves the motion of each individual module, as well as birth and death of modules, is formulated as follows:

$$Net^k = \left(\bigcup_{mo \in Net^{k-1}} S^{k|k-1}(mo) \right) \cup \left(\bigcup_{mo \in Net^{k-1}} \beta^{k|k-1}(mo) \right) \cup \Gamma^k \tag{4.1}$$

where $S^{k|k-1}(mo_i^{k-1})$ is a RFS model of a module with a previous state mo_i^{k-1}, which can take on either $\{mo_i^k\}$ or $\emptyset$. Γ^k is a model for new modules appearing spontaneously

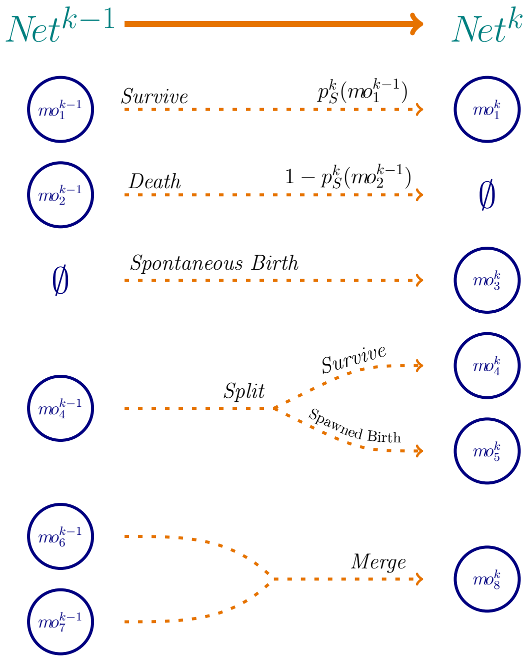


Figure 4.3: Network motion model.

in the network at time k. $\beta^{k|k-1}(mo^{k-1})$ denotes the RFS of new modules spawned from mo^{k-1}. The idea of merge of an unknown number of modules into one module at each time step is reflected in $S^{k|k-1}(mo_i^{k-1})$. Thus, Net^k is a union of all survived modules and all types of new modules.

The network's observation model that accounts for the detection uncertainty and false alarms is formulated as follows:

$$Z^k = \left(\bigcup_{mo \in Net^k} \Theta^k(mo) \right) \cup K^k \tag{4.2}$$

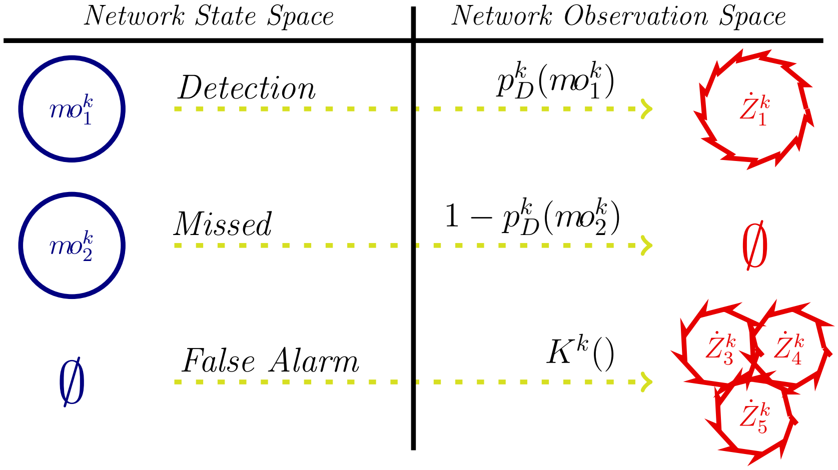


Figure 4.4: Network observation model.

where $\Theta^k(mo)$ is the model for observations captured from the present modules in the networks. This model takes a value $\dot{Z}$ if the module is detected and $\emptyset$ otherwise. K^k is a set of observed false alarms that has its own model.

4.3 Spatio-Temporal Cluster Process and General Cluster Process

A random cluster approach is widely used to model systems which undergo phase transitions or, more generally, systems with a graph structure. Cluster processes are a concept in the theory of point processes, and are described as a superposition of point processes of a cluster [69].

Let $Net = \{mo_1, mo_2, \cdots, mo_M\}$ where $mo_1, mo_2, \cdots, mo_M$ are statistically independent cluster processes. The probability density of Net is related to the probability density of $mo_1, mo_2, \cdots, mo_M$ as follows: (Fundamental convolution formula [45, page.385])

$$p_{Net} = \sum_{m_1 \uplus m_2 \uplus \cdots \uplus m_M = Net} p_{mo_1}(m_1) \cdot p_{mo_2}(m_2) \cdots p_{mo_M}(m_M) \tag{4.3}$$

where the summation is taken over all mutually disjoint modules $mo_1, mo_2, \cdots, mo_M$ of Net such that $mo_1 \cup mo_2 \cup \cdots \cup mo_M = Net$.

Cluster processes are a superposition of cluster centers (an unseen point process, we refer to them as parent processes), to which are associated a random number of nodes (a.k.a. daughter processes).

4.4 Network Filtering

The optimal network Bayesian filter propagates the network posterior density $p^k_{Net}(Net^k|Z^k)$ conditioned on the sets of observations up to time k, $Z^{1:k}$, with the following recursion via network Bayesian prediction and network Bayesian update [69].

$$\begin{aligned} p^{k|k-1}_{Net}(Net^k|Z^{1:k-1}) &= \int f^{k|k-1}(Net^k|net)p^{k-1|k-1}(net|Z^{1:k-1})\delta net & (4.4) \\ p^{k|k}_{Net}(Net^k|Z^{1:k}) &= \frac{L^k(Z^k|Net^k)p^{k|k-1}(Net^k|Z^{1:k-1})}{\int L^k(Z^k|net)p^{k|k-1}(net|Z^{1:k-1})\delta net} & (4.5) \end{aligned}$$

A network can be considered as general cluster processes that are characterized by a component (a module) process p.g.fl, G_{mo}, within a parent cluster (a network) center p.g.fl, G_{Net}, [68].

$$G_{Net}[G_{mo}[h|.]] \tag{4.6}$$

where $G_{mo}[h|.]$ is the p.g.fl of the daughter process for any particular realization of the parent processes. $G_{mo}[h|.]$ is treated as an argument of G_{Net}. We refer to realization of parent process as *virtual leader* modeling and it is described in detail in chapter

5.2. It was shown by Swain and Clark [69], the p.g.fl. forms of prediction and update formula, $p_{Net}^{k|k-1}(Net^k|Z^{1:k-1})$ and $p_{Net}^{k|k}(Net^k|Z^{1:k})$,(Equations 4.4, 4.5) are

$$G_{Net}^{k|k-1}[h] = G_{\Gamma}^{k|k-1}[h]G_{Net}^{k-1|k-1}[\Phi[h]] \tag{4.7}$$

$$G_{Net}^{k|k}[h] = \frac{\frac{\delta F}{\delta Z^k}[0,h]}{\frac{\delta F}{\delta Z^k}[0,1]} \tag{4.8}$$

where

$$G_{Net}^{k-1|k-1}[\Phi[h]] = G_{Net}[G_{mo}[\Phi[h]|.]] = G_{Net}\left(s_p[G_{mo}(s_d[\Phi[h]|.])]\right) \tag{4.9}$$

$$F[g,h] = G_{Net}\left(s_p[G_{mo}(s_d[h.G_L[g|.]|.])]\right) \tag{4.10}$$

$$G_L[g|.] = \int g(z)L_z(.)dz \tag{4.11}$$

$G_{\Gamma}^{k|k-1}[h]$ is the p.g.fl for the set of newly emerging modules, $s_p(u) = s^{k-1|k-1}(u)$ is the p.d.f for parent processes describing motion model of survived modules, and $s_d(w|u) = s^{k-1|k-1}(w|u)$ is the p.d.f. for the daughter process describing node motion models. Complete proofs of Equations 4.7 and 4.8 are provided in [69].

CHAPTER 5

RFS BAYESIAN ESTIMATORS

Estimating the state of a random set is a complex procedure and different statistical and non-statistical methods have been proposed for this propose. Depending on the application, different methods have different goals, e.g., realtime response, low computational cost, or mathematically optimal solution.

An optimal filter propagates the joint probability density of the elements of a random set given the data. Due to the dynamics of random sets, the conventional Bayesian filter is not applicable just by concatenating the elements and forming a random vector variable in a blind fashion and the classical Bayesian optimal state estimators are not applicable in general random finite set situations; therefore, a new estimator must be defined and demonstrated its statistical optimality behavior. In conventional statistics, assuming the prior distribution is unform, the maximum likelihood estimator (MLE) will be a special case of maximum a posteriori (MAP) and, as such, is optimal and convergent, but in random finite case is not true [41]. The MLE of a random finite set is defined as follows:

$$\{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_n\}^{MLE} \triangleq \arg \max_{n, x_1, x_2, \cdots, x_n} f(Z|\{x_1, x_2, \cdots, x_n\}) \tag{5.1}$$

In chapters 3 and 4, it is shown how the optimal Bayesian random finite set filter is capable of recursive propagation of the RFS p.g.fl. in time. However, it is not practical to obtain a sequence of states of RFS due to computational issues. Several computationally feasible approaches have been proposed as an alternative to approximate RFS Bayesian recursive estimator such as Marginal Multitarget Estimator (MaME) and Joint Multitarget Estimator (JoME) [41, 45]. Analogous to the Kalman filter, which is the most successful approximation method for matching the two first order moments (mean and covariance) of the Bayesian estimator, the first moment of the recursive RFS Bayesian estimator is the *Probability Hypothesis Density* (PHD) [43], denoted as $\upsilon^k(x)$ at time k. υ^k is an intensity function associated with RFS posterior. $\int_S \upsilon^k$ is the expected number of elements of a RFS in the hyper-space S.

The problem of our interest, to some extent is similar to the problem of multi-group multi-target tracking problem, but we do consider probabilistic associations between targets. We have used RFS framework to benefit from its mathematical tools, its flexibility and stay in rigorous framework. Although the classical approaches for multi-target tracking such as the multiple hypothesis tracker (MHT)[55] and the joint probabilistic data association filter (JPDAF)[3] could be used to estimate the simplified version of our problem, and also some multi-target tracking algorithms consider interacting targets [23].

5.1 Probability Hypothesis Density filter

The intensity function, or probability hypothesis density (PHD), of a point process is found by taking the functional derivative of the p.g.fl. evaluated at $h = 1$. For example PHD of a Poisson point process (Equation 3.15) is

$$\upsilon(x) = \frac{\delta}{\delta x} G_{mo}[h]|_{h=1} = \lambda_p p(y) \tag{5.2}$$

where λ_p is the expectation of cardinality of a module distributed according to $p(y)$

As mentioned in chapter 3, the RFS Bayesian recursion (Equations 3.16 and 3.17) can alternatively be stated in terms of p.g.fl (Equations 3.18 and 3.20). Let $\dot{\upsilon}^k$ and $\dot{\upsilon}^{k|k-1}$ denote respective intensities (PHD) associated with $G_{mo}^{k|k-1}[h|Z^{1:k-1}]$ and $G_{mo}^{k}[h]$ in the module prediction and the update recursive posterior. It has been shown that the posterior intensity can be propagated recursively in time via the PHD [43]:

$$\dot{\upsilon}^{k|k-1}(x) = \dot{\gamma}^k(x) + \int \dot{p}_S^k(\zeta) \dot{f}_{mo}^{k|k-1}(x|\zeta)\dot{\upsilon}^{k-1}(\zeta)d\zeta \tag{5.3}$$

$$\dot{\upsilon}^{k|k}(x) = [1-\dot{p}_D^k(x)]\dot{\upsilon}^{k|k-1}(x) + \sum_{\dot{z}\in\dot{Z}^k} \frac{\dot{p}_D^k(x)L_{mo}^k(\dot{z}|x)\dot{\upsilon}^{k|k-1}(x)}{\int \dot{p}_D^k(\zeta)L_{mo}^k(\dot{z}|\zeta)\dot{\upsilon}^{k|k-1}(\zeta)d\zeta} \tag{5.4}$$

where $\dot{\gamma}^k()$ is the intensity of the RFS spontaneous birth of a new node. $\dot{p}_S^k(x)$ is the probability that a node still belongs to the module. $\dot{p}_D^k(x)$ is the probability of having an observation from a node (detection). $f_{mo}^{k|k-1}(.|x)$ is the transition density function. The PHD filter has been drive from difrent perspectives, a FISST perspective can be found in [43] and a measure theoretic probability perspective is provided in [74].

A graphical presentation of the PHD filter is depicted in Figure 5.1(a). A random finite set and the number of its elements are shown as a plate, with the elements number in the corner of the plate. The shaded circles indicate observable parameters. The directed edges between variables indicate dependencies between the variables. The dashed edges indicate dependency between hidden variable and observable variables in case of detection.

5.2 Bayesian Estimator for Multiple Hidden-Set Markov Model (MHSMM)

Let us define *Level 1 data fusion* as a problem of detecting, identifying and tracking a module (Figures 3.1 and 5.1(a)), and *Level 2 data fusion* as a problem of detecting, identifying and tracking a random set of modules (Figures 4.2 and 5.1(b)). Mahler [42] has proposed a generalized and computationally tractable strategy for multiple hidden set Markov model (MHSMM), and has shown that this is a PHD of the RFS of hidden sets. In other words, the integral over PHD of MHSMM is the expected number of hidden sets. Swain and Clark [69] have derived the first moment approximation of the independent multiple hidden-set Markov model (MHSMM) Bayesian filter based on the concept of PHD filter for the hidden set Markov model. The posterior intensity can be propagated recursively in time via the PHD [69]:

$$\upsilon^{k|k-1}(\mu, x) = \gamma^{k|k-1}(\mu, x) + \int\int \upsilon^{k-1|k-1}(m,n) p_S(m,n) f^{k|k-1}(\mu, x|m,n) dmdn \tag{5.5}$$

where

$$\begin{aligned}
\gamma^{k|k-1}(\mu, x) &= \text{the intensity function of the network, which describes} \\
&\quad \text{spontaneously emerging modules at time } k && (5.6) \\
\upsilon^{k-1|k-1}(m, n) &= \text{intensity function of the network at time } k-1 && (5.7) \\
p_S(m, n) &= \text{the joint probability a node } n \text{ survival in a module } m && (5.8) \\
f^{k|k-1}(\mu, x|m, n) &= \text{the Markovian transition density for a node state } x \text{ in module } \mu \\
&\quad \text{given node state } w \text{ in module state } u \text{ at time } k && (5.9)
\end{aligned}$$

$$\upsilon^{k|k}(\mu, x) = \sum_{\mathfrak{P} \in \mathcal{P}(Z^k)} \omega_{\mathfrak{P}} \underbrace{\sum_{W \in \mathfrak{P}} \frac{s_1(\mu) L_{mo}(W|\mu)}{s_1[L_{mo}(W)]}}_{\text{intensity update for virtual leader}} \overbrace{\sum_{\dot{z} \in W} \frac{s_2(x|\mu) L_{\dot{z}}(z|x, \mu)}{s_2[L_{\dot{z}}|\mu]}}^{\text{intensity update for node}} \tag{5.10}$$

where

$$\omega_{\mathfrak{P}} = \frac{|\mathfrak{P}|! \rho_\mu(|\mathfrak{P}|) \Pi_{W \in \mathfrak{P}} s_1[L_{mo}(W)]}{\sum_{Q \in \mathcal{P}(Z)} |Q|! \rho_\mu(|Q|) \Pi_{W \in Q} s_1[L_{mo}(W)]} \tag{5.11}$$

$$L_{mo}(W|\mu) = |W|! \rho_x(|W||\mu) \Pi_{\dot{z} \in W} s_2[L_{\dot{z}}|\mu] \tag{5.12}$$

and

$$L_{\dot{z}} = \text{a node likelihood for individual measurements } \dot{z} \in W \tag{5.13}$$

$$s_1(\mu) = s^{k|k-1}(\mu) = \text{p.d.f of parent (virtual)process} \tag{5.14}$$

$$s_2(x|\mu) = s^{k|k-1}(x|\mu) = \text{p.d.f of daughter (node) process} \tag{5.15}$$

$$\rho_\mu, \rho_x = \text{predicted cardinalities of parent and daughter processes} \tag{5.16}$$

A partition of a observation set Z^k is defined as set of subsets of Z^k such that those subsets have not any intersect and the union of all is equal to Z^k. $\mathcal{P}(Z^k)$ in Equation 5.10 is the set of all partitions of the observation set Z^k.

5.3 Simplified MHSMM

The first moment density of the network Bayesian update (Equation 5.10) is presented under assumption of no false alarm and missed detection. Considering the false alarm and missed detection cases increases the complexity of the problem and computationally will be very expensive. If the problem is only to detect, estimate, and identify hidden sets and their elements, but not to precisely estimate the states of elements of the hidden sets, then MHSMM will be simplified to a HSMM, but still will preserve its dynamics (birth, split and merge). Figure 5.1(c) illustrates the plate notation of the simplified version of RFS Bayesian estimator for MHSMM. This simplification is based on a hypothetical proposition that any module of a network can be parameterized mathematically.

A group of elements (e.g., a module composed of network nodes) unavoidably develops properties which are not a simple summation of the properties of its elements (a phenomenon widely known as emergence). Hypothetically, the nodes of a module are coordinated. The way they are coordinated can change over time, and can be described with the help of a "*virtual leader*". A wide variety of parameters can be used as virtual leaders, for instance geometric centroid [12] or parameters of the probability distribution of nodes of a module (Figure 7.1). Given that definition of a virtual leader can be formulated, one can apply the PHD filter between the network state space and the module space (the upper and the middle levels in Figure 4.2).

Let us assume that nodes of a module belong to a Gaussian distribution, $\{x_i \sim \mathcal{N}(\mu_j, \sigma_j^2) | x_i \in \mathscr{mo}_j\}$. Then we can use μ_j and σ_j as the virtual leader of $\mathscr{mo}_j$. Consider υ^k and $\upsilon^{k|k-1}$ as the respective intensities (PHD) associated with $p^{k|k-1}(\mathscr{Net}^k|Z^{k-1})$ and $p^{k|k}(\mathscr{Net}^k|Z^k)$ in the prediction and the update recursive posterior (Equations 4.4 and 4.5). The posterior intensity can be propagated recursively in time via the PHD:[75]

$$\upsilon^{k|k-1}(\mu) = \int p_S^k(\zeta) f^{k|k-1}(\mu|\zeta)\upsilon^{k-1}(\zeta)d\zeta + \int \beta^{k|k-1}(\mu|\zeta)\upsilon^{k-1}(\zeta)d\zeta + \gamma^k(\mu) \quad (5.17)$$

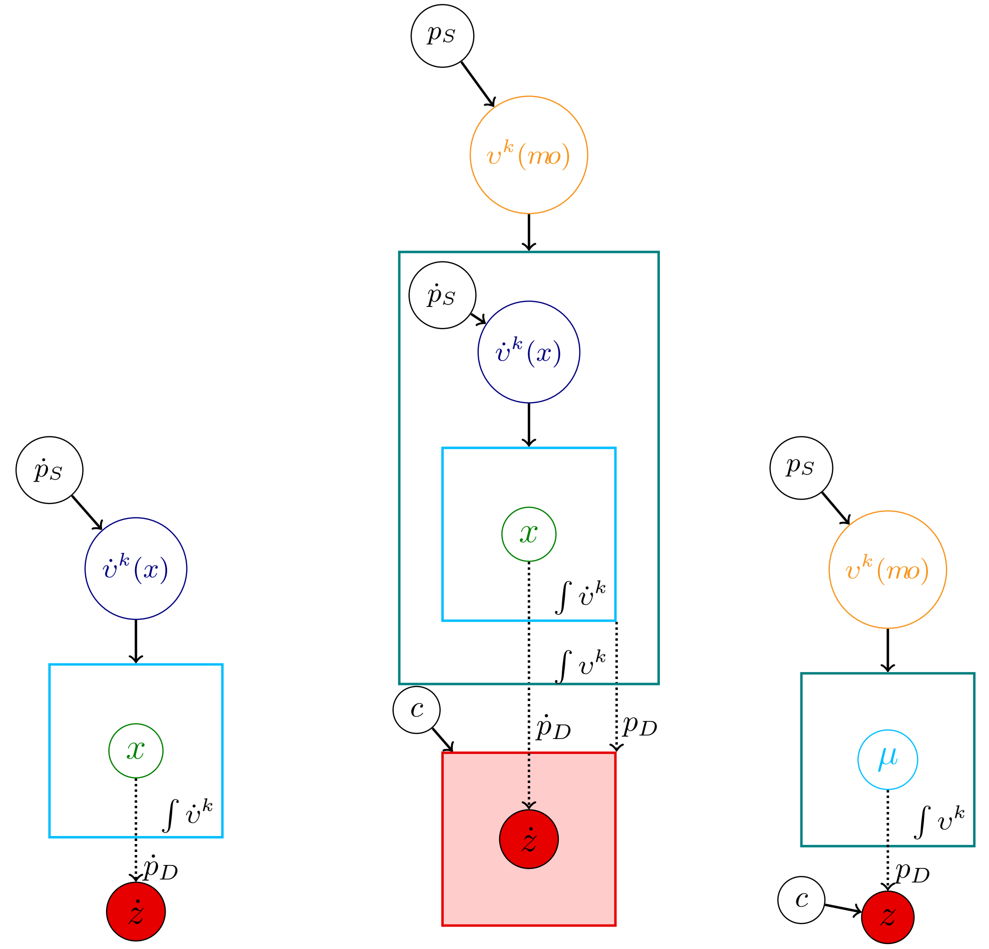
$$\upsilon^k(\mu) = [1 - p_D^k(\mu)]\upsilon^{k|k-1}(\mu) + \sum_{z \in \mathcal{Z}^k} \frac{p_D^k(\mu) g^k(z|\mu)\upsilon^{k|k-1}(\mu)}{\kappa^k(z) + \int p_D^k(\zeta) g^k(z|\zeta)\upsilon^{k|k-1}(\zeta)d\zeta} \quad (5.18)$$

where κ^k is the intensity of the false alarm. The difference between $\dot{\upsilon}^k(x)$ (Equations 5.3 and 5.4) and $\upsilon^k(\mu)$ (Equations 5.17 and 5.18) comes from the difference in motion and observation models of a module and a network (Figures 3.2, 3.1, 4.3, 4.4). Also it is required to have an tranformation function that transform nodes' observation to virtual leaders' observation. In following, an example shows a transformation.

$$Z = \{\dot{z}_1, \dot{z}_2, \dot{z}_3, \dot{z}_4, \dot{z}_5, \dot{z}_6, \dot{z}_7, \dot{z}_8\} \xrightarrow{\text{partition}} W = \{\{\dot{z}_1, \dot{z}_4, \dot{z}_7\}, \{\dot{z}_3, \dot{z}_5\}, \{\dot{z}_2, \dot{z}_6, \dot{z}_8\}\}\text{Ex.5}$$

$$\xrightarrow{\text{transform}} \mathcal{Z} = \{z_1, z_2, z_3\} \quad (5.19)$$

To have virtual leaders observation set $\mathcal{Z}$, one can partition off nodes observation set with any proper clustering algorrithm (e.g.W in Example 5) then uses the computed clustering parameters as representors of partion's elements (e.g.$\mathcal{Z}$ in example 5). For example, a part in a partition can be transform to a virtual leader's observation by assigning it to a distribution, or computing its mean, or median.



(a) Level 1 Data Fusion

(b) Level 2 Data Fusion

(c) Simplified version of Level 2

Figure 5.1: Plate model of the PHD filter.

CHAPTER 6

IMPLEMENTATION

This chapter discusses how the two main methods used for approximation of the PHD filter (SMC-PHD filter [76] and Gaussian Mixture PHD filter (GM-PHD) [75]) are applied as an solution for estimation of a random finite set state. We used the RFS framework to benefit from its rigorous mathematical tools and its flexibility. As it is mentioned before, the problem of inferring the state of a random finite set given its dynamic is, to some extent, similar to the problem of multi-target tracking problem. Although modified versions of classical approaches for multi-target tracking such as the multiple hypothesis tracker (MHT)[3] or the joint probabilistic data association filter (JPDAF)[3], could potentially be used to solve our problem,

6.1 Gaussian Mixture PHD Filter

The GM-PHD is a closed form solution for the PHD filter (Equestions 5.17 and 5.18) under the following assumptions [75]:

- Virtual leaders have linear Gaussian motion model and observation model during their life time.
- The intensity function of the spontaneous and spawned births of modules are Gaussian mixtures.

- There is state independent probability of survival and detection.

These assumptions are formulated for the problem of our interest as follows:

$$f^{k|k-1}(\mu|\zeta) = \mathcal{N}(\mu; F\zeta, Q^{k-1}) \tag{6.1}$$

$$L^k(z|\mu) = \mathcal{N}(z; H\mu, R^k) \tag{6.2}$$

$$p_S^k = p_S \tag{6.3}$$

$$p_D^k = p_D \tag{6.4}$$

where μ is a module's virtual leader state, F is the state transition matrix, Q^{k-1} is the process noise covariance, H is the observation matrix, and R^k is the observation noise covariance. The intensities of the spontaneous and spawned births are assumed to be Gaussian mixtures of the form

$$\gamma^k(\mu) = \sum_{i=1}^{J_\gamma^k} \omega_{\gamma,i}^k \mathcal{N}(\mu; m_{\gamma,i}^k, P_{\gamma,i}^k) \tag{6.5}$$

$$\beta^{k|k-1}(\mu|\zeta) = \sum_{j=1}^{J_\beta^k} \omega_{\beta,j}^k \mathcal{N}(\mu; F_\beta\zeta + d_{\beta,j}, Q_{\beta,j}^k) \tag{6.6}$$

where J_γ^k is the number of new virtual leaders spontaneous births and $\omega_{\gamma,i}^k$, $m_{\gamma,i}^k$, $P_{\gamma,i}^k$ are the weight, the mean, and the covariance of their intensity, respectively. $m_{\gamma,i}^k, i = 1, \ldots, J_\gamma^k$ correspond to the J_γ^k highest concentration. The covariance matrix $P_{\gamma,i}^k$ determines the spread of the birth intensity around $m_{\gamma,i}^k$, and $\omega_{\gamma,i}^k$ is a weight given to the new virtual leader originated from $m_{\gamma,i}^k$ [75]. Similarly, J_β^k, $\omega_{\beta,j}^k$, F_β, $d_{\beta,j}^{k-1}$, and $Q_{\beta,j}^{k-1}$ determine the shape of the spawning intensity of a virtual leader with a previous state ζ [75]. Spawned virtual leaders at time k are an affine function $(F_\beta\zeta + d_{\beta,j})$ of a virtual leader (parent) at state ζ at time $k-1$ [75]. The general form of GM-PHD filter can be found in [75] in details.

6.1.1 GM-PHD Prediction

Given the assumptions listed above, the predicted intensity from time $k-1$ to time k is a Gaussian mixture [75]:

$$\upsilon^{k|k-1}(\mu) = \upsilon_S^{k|k-1}(\mu) + \upsilon_\beta^{k|k-1}(\mu) + \gamma^k(\mu) \tag{6.7}$$

where

$$\begin{aligned}
\upsilon_S^{k|k-1}(\mu) &= p_S^k \sum_{j=1}^{J^{k-1}} \omega_j^{k-1} \mathcal{N}(\mu; m_{S,j}^{k|k-1}, P_{S,j}^{k|k-1}) && (6.8)\\
m_{S,j}^{k|k-1} &= F^{k-1} m_j^{k-1} && (6.9)\\
P_{S,j}^{k|k-1} &= Q^{k-1} + F^{k-1} P_j^{k-1} {F^{k-1}}^T && (6.10)\\
\upsilon_\beta^{k|k-1}(\mu) &= \sum_{j=1}^{J^{k-1}} \sum_{l=1}^{J_\beta^k} \omega_j^{k-1} \omega_{\beta,l}^k \mathcal{N}(\mu; m_{\beta,(j,l)}^{k|k-1}, P_{\beta,(j,l)}^{k|k-1}) && (6.11)\\
m_{\beta,(j,l)}^{k|k-1} &= F_{\beta,l}^{k-1} m_i^{k-1} + d_{\beta,l}^{k-1} && (6.12)\\
P_{\beta,(i,j)}^{k|k-1} &= Q_{\beta,l}^{k-1} + F_{\beta,l}^{k-1} P_{\beta,j}^{k-1} (F_{\beta,l}^{k-1})^T && (6.13)
\end{aligned}$$

6.1.2 GM-PHD Update

The posterior intensity at time k is also a Gaussian mixture [75]:

$$\upsilon^k(\mu) = (1 - p_D^k)\upsilon^{k|k-1}(\mu) + \sum_{z \in Z^k} \upsilon_D^k(\mu; z) \tag{6.14}$$

where

$$\upsilon_D^k(\mu; z) = \sum_{j=1}^{J^{k|k-1}} \omega_j^k(z)\mathcal{N}(\mu, m_j^{k|k}(z), P_j^{k|k}), \tag{6.15}$$

$$\omega_j^k(z) = \frac{p_D^k \omega_j^{k|k-1} q_j^k(z)}{\kappa^k(z) + p_D^k \sum_{l=1}^{J^{k|k-1}} \omega_l^{k|k-1} q_l^k(z)} \tag{6.16}$$

$$q_j^k(z) = \mathcal{N}(z; H^k m_j^{k|k-1}, R^k + H^k P_j^{k|k-1} H^{k^T}) \tag{6.17}$$

$$m_j^{k|k}(z) = m_j^{k|k-1} + K_j^k(z - H^k m_j^{k|k-1}) \tag{6.18}$$

$$P_j^{k|k} = [I - K_j^k H^k] P_j^{k|k-1} \tag{6.19}$$

$$K_j^k = P_j^{k|k-1} H^{k^T}(H^k P_j^{k|k-1} H^{K^T} + R^k)^{-1} \tag{6.20}$$

6.2 Merge of Modules

Merging of modules occurs when they get so close (similar) to each other. Vo and Ma [75], introduced a heuristic pruning algorithm ro reduce the number of Gaussian components propagated to the next time step. In theory, it approximates some of the close Gaussian component by a single Gaussian component.

Let $\mathbb{I}^k$ indicate indices of modules at time k, and $I \subset \mathbb{I}^k$ indicate indices of modules that have got closed together. These modules can be approximated with one Gaussian component as follows [75]:

$$\tilde{\omega}_l^k = \sum_{i \in I} \omega_i^k \tag{6.21}$$

$$\tilde{m}_l^k = \frac{1}{\tilde{\omega}_l^k} \sum_{i \in I} \omega_i^k m_i^k \tag{6.22}$$

$$\tilde{P}_l^k = \frac{1}{\tilde{\omega}_l^k} \sum_{i \in I} \omega_i^k (P_i^k + (\tilde{m}_l^k - m_i^k)(\tilde{m}_l^k - m_i^k)^T) \tag{6.23}$$

6.3 State Estimation

For a RFS $\mathcal{N}et^k$ with a probability distribution p, the integral of $\upsilon^{k|k}$ over the network state space gives the expected number of modules of $\mathcal{N}et^k$. Hence, $\hat{M}^k = \int \upsilon(\mu)\partial\mu$. The local maxima of the intensity $\upsilon^{k|k}$ are the highest local concentration of the modules, and hence the $[\hat{M}^k]$ highest peaks from the intensity function can be selected as the estimation of the state of each module's virtual leader ([.] means rounded number).

Given the Gaussian mixture intensities $\upsilon^{k|k-1}$ and $\upsilon^{k|k}$, the corresponding expected number of virtual leaders (number of modules) $\hat{M}^{k|k-1}$ and $\hat{M}^k$ can be obtained by summing up the appropriates weights [75]:

$$\hat{M}^{k|k-1} = \hat{M}^{k-1|k-1}\left(p_S^k + \sum_{j=1}^{J_\beta^k} \omega_{\beta,j}^k\right) + \sum_{j=1}^{J_\gamma^k} \omega_{\gamma,j}^k \tag{6.24}$$

$$\hat{M}^{k|k} = \hat{M}^{k|k-1}(1 - p_D^k) + \sum_{z \in Z^k} \sum_{j=1}^{J^{k|k-1}} \omega_j^k(z) \tag{6.25}$$

In parametric state estimation, state of virtual leaders are represented as mixture of Gaussian models with parameters mean, covariance and mixing proportions (weights). Theoretically, the $\hat{M}^k$ highest components are the locations of the virtual leaders, but since each peak also is described by the weight and covariance, it is possible that a peak correspond to a Gaussian component with a weak weight but a large height, a better alternative is to first filter out the Gaussian components with small weight first [75];

One of the main criticisms of the PHD filter is that there is no means of associating the same virtual leader between time frames. But this is of advantage for our problem, because our main goals are the number and the locations of virtual leaders. It is trivial to identify the trajectories of different virtual leader by tracking their members.

6.3.1 Gaussian Mixture Implementation of PHD Recursion

The steps of GM-PHD filter implementation is described in Algorithm 1 as given in [75]

```
Data: $\{\omega_i^{k-1}, m_i^{k-1}, P_i^{k-1}\}_{i=1}^{J^{k-1}}$ and $Z^k$
Result: $\{\omega_i^k, m_i^k, P_i^k\}_{i=1}^{J^k}$
Step 1. prediction for birth modules;
$i = 0$;
for $j = 1$ to $J_\gamma^k$ do
    $i := i + 1$;
    $\omega_i^{k|k-1} := \omega_{j,\gamma}^k, \quad m_i^{k|k-1} := m_{j,\gamma}^k, \quad P_i^{k|k-1} := P_{j,\gamma}^k$;
end
for $j = 1$ to $J_\beta^k$ do
    for $\ell = 1$ to $J^{k-1}$ do
        $i := i + 1$;
        $\omega_i^{k|k-1} := \omega_{j,\beta}^k \times \omega_l^{k-1}$;
        $m_i^{k|k-1} := d_{j,\beta}^{k-1} + F_{j,\beta}^{k-1} \times m_l^{k-1}$;
        $P_i^{k|k-1} := Q_{j,\beta}^{k-1} + F_{j,\beta}^{k-1} \times P_l^{k-1} \times (F_{j,\beta}^{k-1})^T$;
    end
end
Step 2. prediction for existing modules;
for $j = 1$ to $J^{k-1}$ do
    $i := i + 1$;
    $\omega_i^{k|k-1} := p_S \times \omega_j^{k-1}$;
    $m_i^{k|k-1} := F^{k-1} \times m_j^{k-1}$;
    $P_i^{k|k-1} := Q^{k-1} + F^{k-1} \times P_j^{k-1} \times (F^{k-1})^T$;
end
$J^{k|k-1} := i$        %$J_\gamma^k + J^\beta + J^{k-1}$ ;
%Continue...;
```

Algorithm 1: Psedocode for GM-PHD Filter

```
%Continue from previous page;
Step 3. construction of PHD update components;
for $j = 1$ to $J^{k|k-1}$ do
    $\eta_j^{k|k-1} := Hm_j^{k|k-1}$;    $S_j^k := R^k + HP_j^{k|k-1}(H^k)^T$;
    $K_j^k := P_j^{k|k-1}(H^k)^T[S_j^k]^{-1}$;    $P_j^{k|k}[I - K_j^k H^k]P_j^{k|k-1}$;
end
Step 4. update ;
for $j = 1$ to $J^{k|k-1}$ do
    $\omega_j^k := (1 - p_D^k)\omega_j^{k|k-1}$;
    $m_j^k := m_j^{k|k-1}$;    $P_j^k := P_j^{k|k-1}$;
end
$l := 0$;
foreach $z \in Z^k$ do
    $\ell := \ell + 1$;
    for $j = 1$ to $J^{k|k-1}$ do
        $\omega^k_{\ell J^{k|k-1}+j} := p_D \omega_j^{k|k-1} \mathcal{N}(z; \eta_j^{k|k-1}, S_j^k)$;
        $m^k_{\ell J^{k|k-1}+j} := m_j^{k|k-1} + K_j^k(z - \eta_j^{k|k-1})$;
        $P^k_{\ell J^{k|k-1}+j} := P_j^{k|k}$;
    end
    for $j = 1$ to $J^{k|k-1}$ do
        $\omega^k_{\ell J^{k|k-1}+j} := \frac{\omega^k_{\ell J^{k|k-1}+j}}{K^k(z)+\sum_{i=1}^{J^{k|k-1}} \omega^k_{\ell J^{k|k-1}+i}}$;
    end
end
$J^k := (\ell + 1)J^{k|k-1}$;
```

Algorithm 2: Psedocode for GM-PHD Filter (continued)

6.4 Sequential Monte Carlo Implementation of PHD Filter

The sequential Monte Carlo implementation of PHD filter [76] was proposed as a practical suboptimal alternative to the optimal PHD filter. Briefly, the sequential Monte Carlo (SMC) propagates particles in the prediction stage using a prior distribution, dynamic model and noise process of the system, then each particle will be assigned a weight calculated based on likelihood of particle (statistical distance of particle to the set of observation). The weighted particles are representation of the PHD and the sum of weight gives the estimated cardinality. For any $k \geq 1$, let $\{\ddot{\omega}_i^k, \ddot{x}_i^k\}_{i=1}^{\ddot{J}^k}$ be the particle representation of intensity function (υ^k). Algorithm 3 summarizes SMC implementation of PHD filter as given in [76]. The particle are sampled from two importance (or proposal) densities $q^k(.|\ddot{x}_i^{k-1}, Z^k)$ and $q_b^k(.|Z^k)$. $\ddot{J}_b^k$ denotes new particles arise from the birth process. $\phi^{k|k-1}$ is the survived and spawned intensity function and $\gamma^k()$ is the spontaneous birth intensity function.

```
At time $k \geq 1$;
**Step 1.** Prediction;
**for** $i = 1$ **to** $\ddot{J}^{k-1}$ **do**
    Sample $\tilde{\ddot{x}}_i^k \sim q^k(.|\ddot{x}_i^{k-1}, Z^k)$;
    $\tilde{\ddot{\omega}}_i^{k|k-1} = \frac{\phi^{k|k-1}(\tilde{\ddot{x}}_i^k, Z^k)}{q^k(\tilde{\ddot{x}}_i^k|\ddot{x}_i^{k-1}, Z^k)} \ddot{\omega}_i^{k-1}$;
**end**
$\ddot{J}_b^k = \ddot{J}_\gamma^k + \ddot{J}_\beta^k$;
**for** $i = \ddot{J}^{k-1} + 1$ **to** $\ddot{J}^{k-1} + \ddot{J}_b^k$ **do**
    Sample $\tilde{\ddot{x}}_i^k \sim q_b^k(.|Z^k)$;
    $\tilde{\ddot{\omega}}_i^{k|k-1} = \frac{1}{\ddot{J}_b^k} \frac{\gamma^k(\tilde{\ddot{x}}_i^k)}{q_b^k(\tilde{\ddot{x}}_i^k|Z^k)}$;
**end**
**Step 2.** Update ;
**foreach** $z \in Z^k$ **do**
    $C^k(z) = \sum_{j=1}^{\ddot{J}^{k-1}+\ddot{J}_b^k} p_D^k(\tilde{\ddot{x}}_j^k) L^k(z|\tilde{\ddot{x}}_j^k) \tilde{\ddot{\omega}}_j^{k|k-1}$;
**end**
**for** $i = 1$ **to** $\ddot{J}^{k-1} + \ddot{J}_b^k$ **do**
    $\tilde{\ddot{\omega}}_i^k = [(1 - p_D^k(\tilde{\ddot{x}}_i^k)) + \sum_{z \in Z^k} \frac{p_D^k(\tilde{\ddot{x}}_i^k) L^k(z|\tilde{\ddot{x}}_j^k)}{\kappa^k(z) + C^k(z)}] \tilde{\ddot{\omega}}_i^{k|k-1}$;
**end**
**Step 3.** Resampling ;
$\hat{M}^{k|k} = \sum_{j=1}^{\ddot{J}^{k-1}+\ddot{J}_b^k} \tilde{\omega}_j^k$;
Resample $\{\frac{\tilde{\ddot{\omega}}_i^k}{\hat{M}^{k|k}}, \tilde{\ddot{x}}_i^k\}_{i=1}^{\ddot{J}^{k-1}+\ddot{J}_b^k}$ to get $\{\frac{\ddot{\omega}_i^k}{\hat{M}^{k|k}}, \ddot{x}_i^k\}_{i=1}^{\ddot{J}^k}$;
Multiply the weights by $\hat{M}^{k|k}$ to get $\{\ddot{\omega}_i^k, \ddot{x}_i^k\}_{i=1}^{\ddot{J}^k}$;
```

Algorithm 3: Psedocode for SMC-PHD Filter

6.5 Smoothing Algorithms for the PHD Filter

As forward-backward algorithm (a.k.a smoother) is an inference algorithm in HMM to improve the estimation result, one can extended it for the PhD filter to correct the unexpected changes on the state estimations. This inference task is usually called smoothing. Briefly the idea of smoother is computing $p(Net^k|Z^{1:T})$ instead of $p(Net^k|Z^{1:k})$, where $Z^{1:T}$ means observations up to time T and $Z^{1:k}$ means observations up to time k and $T > k$.

Nandakumaran et al. [48, 50] has proposed approximate backward PHD filter smoothers under Poisson assumption and applied it on sonar data [49]. Mahler et al. [46] derived the backward PHD smoother recursion mathematically and Vo et al. [77] proposed a closed form solution to the PHD filter under linear Gaussian assumption. Also Hernández [29] has derived two smoothing algorithms and provided their sequential Monte Carlo implementations.

Clark and Vo [13] have proven the GM-PHD filter maintains a suitable approximation error in each time step, and its error converges to zero uniformly as the number of Gaussian components tends to infinity. Also Clark and Bell [11] have provided mathematical proofs for SMC-PHD filter and bounds for the mean square error.

In case of PHD filter, the sensor noise and uncertainty of a RFS's cardinality propagate in time. In order to improve these limitations, Mahler [44] derived a generalization of the PHD recursion known as the cardinalized PHD (CPHD) filter by relaxing the first order assumption on the cardinality of a RFS. CPHD filter jointly propagates the intensity function and the cardinality distribution.

CHAPTER 7

STOCHASTIC EVOLUTION DIAGRAM (SED)

The trajectory of each virtual leader can be projected on a d-dimensional coordinate space (d is the number of parameters which represent coordinates of a virtual leader). For example, assume that virtual leaders are 1-dimensional, the footprints in Figure 7.1 show the states of such modules.

Similar to the multivariate Markov model of time series that reconstructs Markov chains, the multiple hidden set Markov model reconstructs the *stochastic evolution diagram*. We introduce this term to denote a collection of Markov chains, of which some chains are tied together at certain time points.

A Markov chain can be considered as a directed spatiotemporal graph where each vertex represents the state of the system at a particular time. Defined formally, the stochastic evolution diagram is a directed spatiotemporal graph where each node belongs to the $\mathbb{R}^{|d|} \times \mathbb{T}$ space, $\mathbb{R}$ is the real number space, $|d|$ is the dimension of coordinates of virtual leader, and $\mathbb{T}$ is the time space. The vertices of this graph represent states of each module of the system. If the indegree or outdegree of a vertex is not one, it is a change-point (network phase transition) and the edges represent a Markov chain of the estimated states of the modules' virtual leaders. A schematic example of a stochastic evolution diagram for 1-dimensional virtual leaders is illustrated in

Figure 7.2. The stochastic evolution diagram visualizes all information about the state of the system, such as lifetime of modules (birth, death, split and merge time), members of modules and number of modules at each time. Consider the following

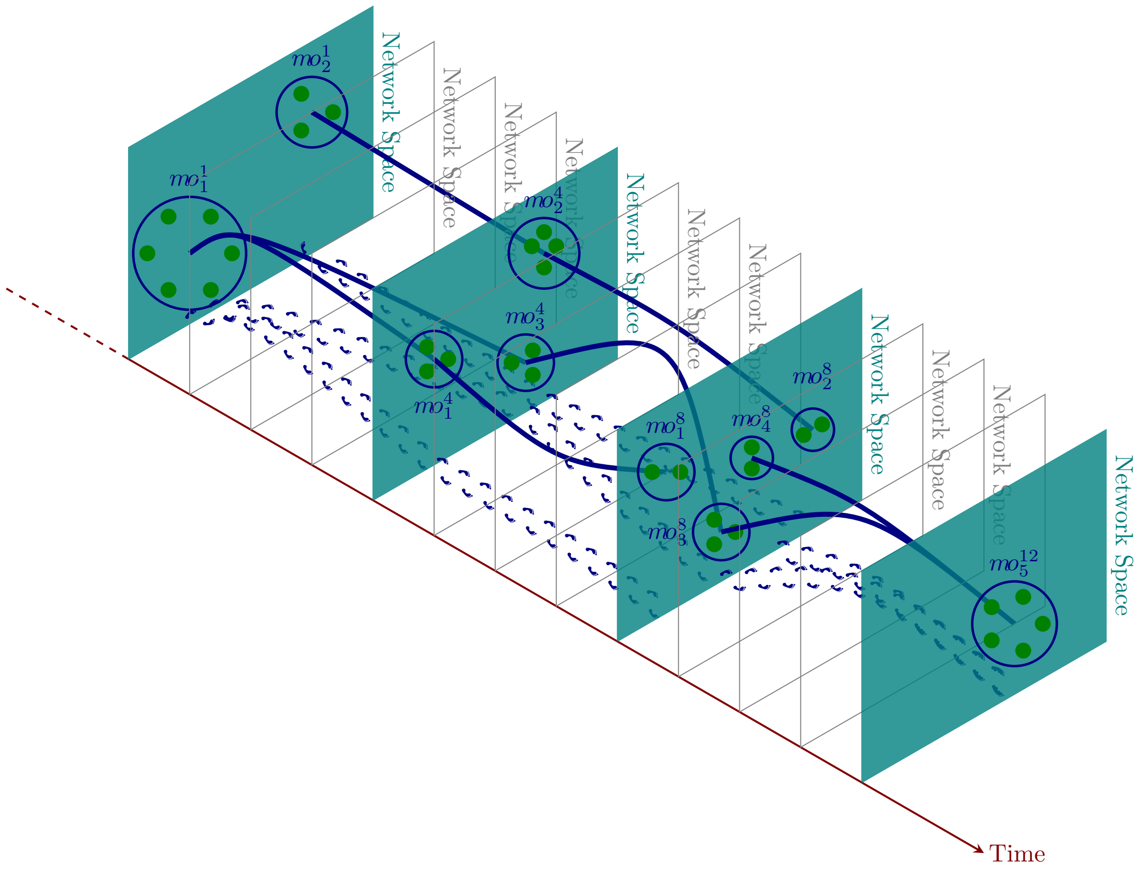


Figure 7.1: Schematic illustration of a stochastic evolution diagram, where the bold blue lines are trajectories of modules' virtual leaders. The footprints are projections of virtual leaders on the coordinate system.

Markov chain from Figure 7.2

$$\{x_2, x_3, x_4, x_5\} \xrightarrow{split} \{x_1, x_2, x_3\} \xrightarrow{split} \{x_1, x_2\} \tag{7.1}$$

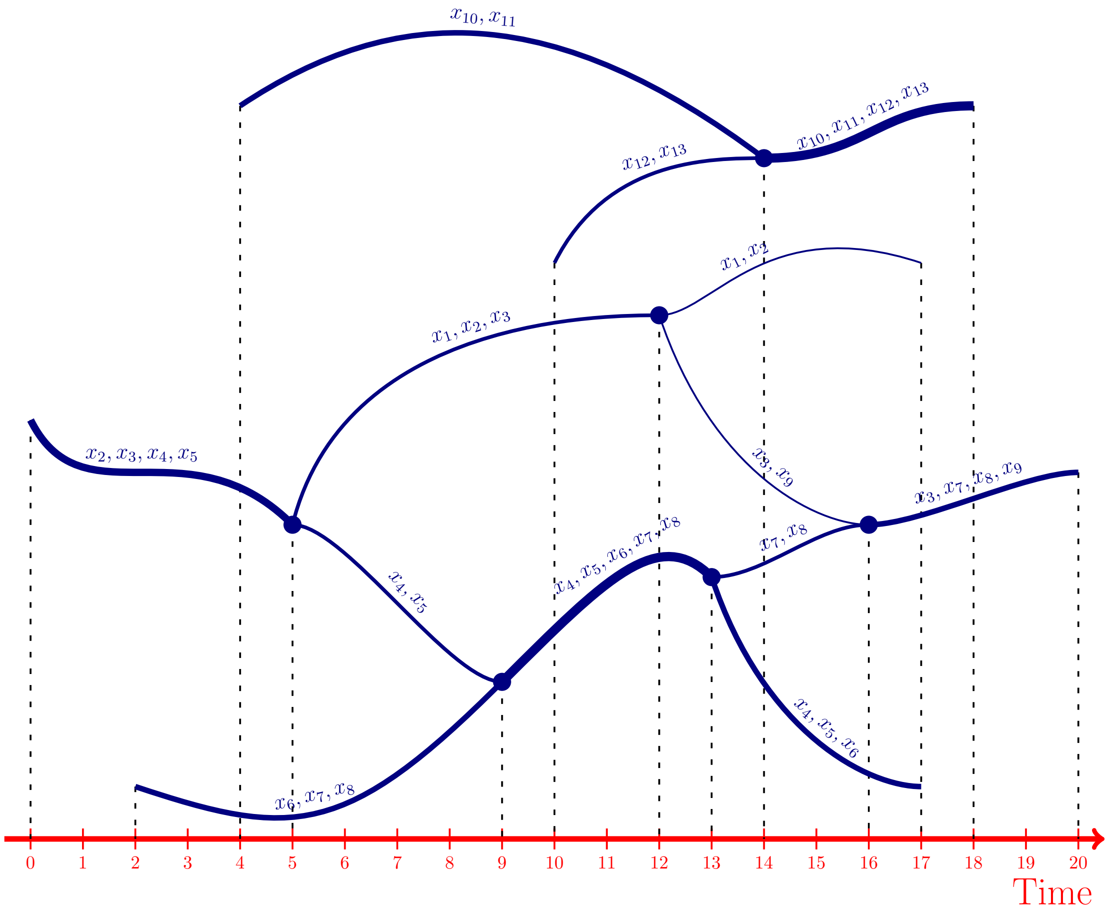


Figure 7.2: The Stochastic Evolution Diagram (only vertices that are change-points are shown by bold blue circles).

This expression shows that, before the network phase transition at time 5 there was a module with four elements, but at time 5 the module split into two modules, and a new element x_1 became a new member. After the phase transition the module was following steady rules for its dynamics up to the next network phase transition at time 12. Although two network transitions happened at time 9 and 10, they did not affect on that module. Also, one can examine the features of the system individually

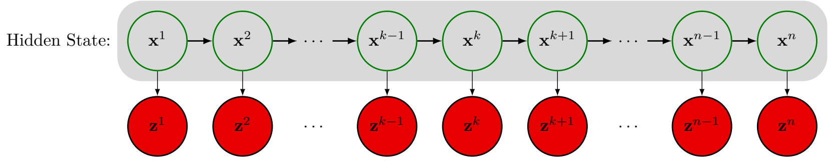


Figure 7.3: HMM.

and track their effective time (a feature is effective when it belongs to a module) . For example, x_6 only belongs to the underlying network from time 2 to 17.

7.1 Generalization of HMM Step by Step

In this section, I will explain the generalized model of the Hidden Markov Model. I use a multi-target tracking example and present how to generalize the hidden Markov model step by step. In this research, a model generalization means introducing a new model which has fewer limiting assumptions than the old one. In conventional HMM, there are two assumptions. Here is a simple object tracking example:

Suppose the HMM is used to track objects. In such a model one needs to assume that:

1. Each target always exists in the space (probability of survival (p_S) is one)
2. The sensor always detects the objects in the space and has a noisy observation (probability of detection (p_D) is one).

These two assumptions result in a Markov chain (Figure 7.3).

The first-step generalization of the HMM is as follows:

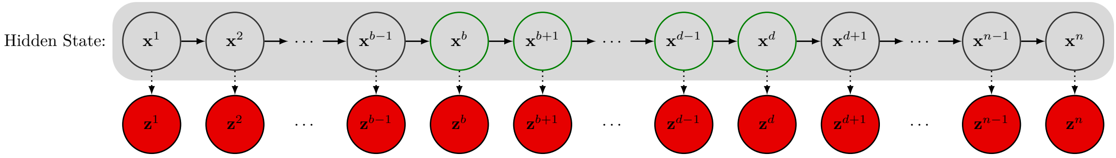


Figure 7.4: Appearing and disappearing a target. Black circles and green circles in the hidden state space represent state of the target when it is out of field of view and in the field of view respectively.

1. There is a possibility of disappearing of the object in the space at any time ($p_S \leqslant 1$), and there is a possibility of new objects to appear at any time.

2. It is possible that an object exists in the space but the sensor doesn't detect the object because of its temporary defection ($p_D \leqslant 1$).

The two assumptions above yield a segmented Markov chain or, in other words, a Markov chain which is disjointed at some time points (phase transition times; Figures 7.4 and 7.5). Black circles and green circles in the hidden state space represent state of the target when it is out of field of view and in the field of view respectively. For the second-step generalization, consider the multi-target tracking scenario, where each target not only can appear and disappear, but also targets can merge into a single target (Figure 7.6). Conversely, a target can split to some smaller target (Figure 7.7). Figures 7.4, 7.6 and 7.7 are probability graphical representations of Figure 1.1.

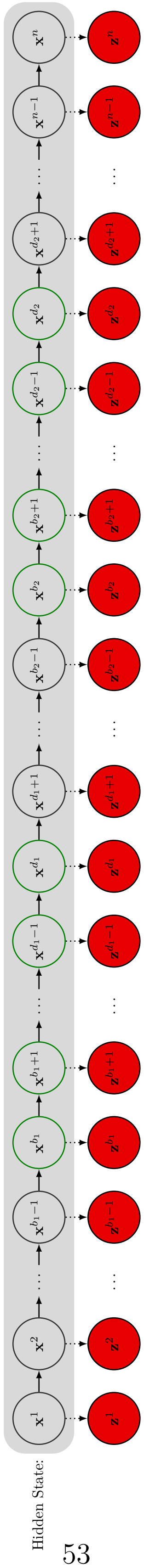


Figure 7.5: Appearing and disappearing targets.

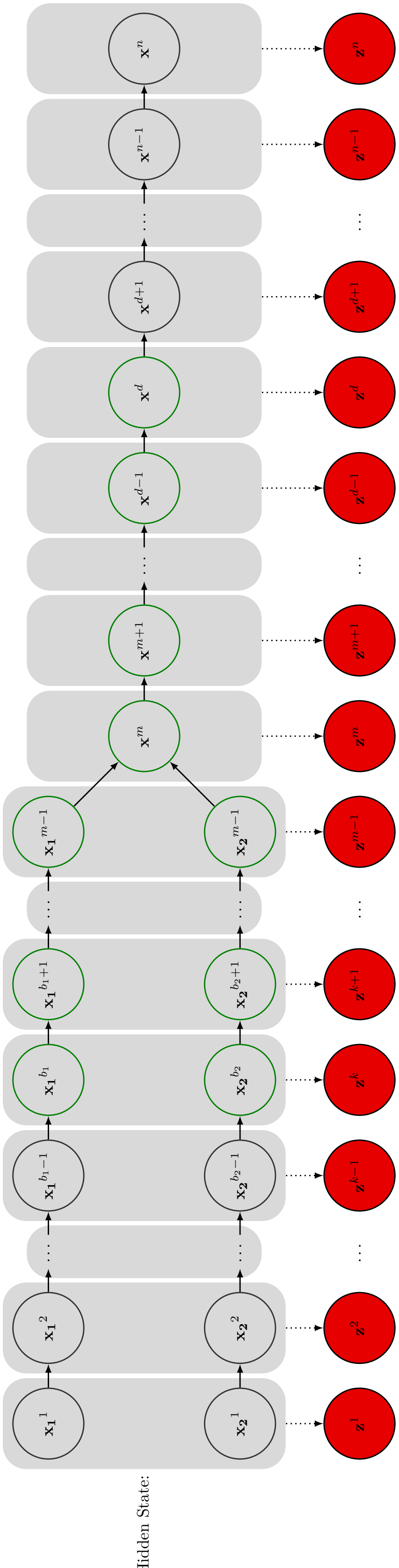


Figure 7.6: Merge of two targets into a bigger target.

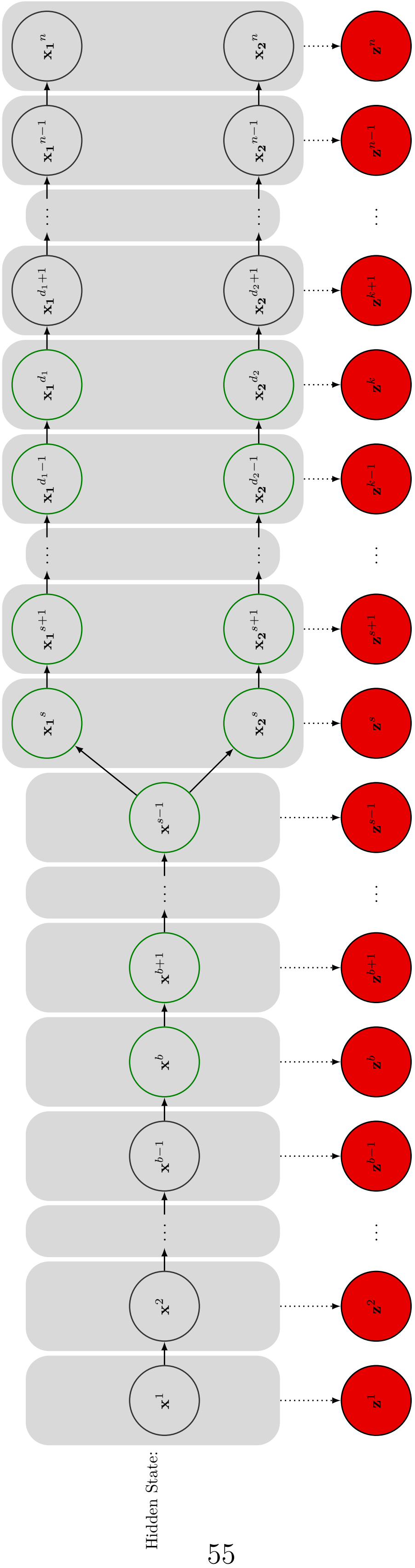


Figure 7.7: Split of a target to two smaller targets.

CHAPTER 8

IDENTIFICATION OF THE STOCHASTIC EVOLUTION DIAGRAM (SED)

We begin this chapter with a review of hidden Markov model (HMM) elements and focus on the three fundamental problems for HMM design and identification. Briefly these are the problems, 1) evaluation of the probability of an observed sequence given a specific HMM, 2) inferring the highest probable sequence of hidden state variables, 3) how to learn model parameters to maximize the likelihood function.

8.1 HMM Elements

Let $h = \{x^1, \cdots, x^T\}$ and $O = \{z^1, \cdots, z^T\}$ construct a HMM depicted in Figure 7.3 and $x^i \in \mathbb{S}$ and $z^i \in \mathbb{O}$ represent hidden state variables and observation variables. $\mathbb{S}$ is the state space and $\mathbb{O}$ is the observation space. An HMM is characterized by the following:

1. The state transition function $f(.)$, that returns probability of reaching from one state to another one in a single step.

2. The observation function $\mathcal{L}^k(.)$, that returns probability of observing a particular value when a hidden variable is in a particular state.

3. The initial state p^1.

An HMM identification requires to specify $\mathbb{S}, \mathbb{O}, f(), \mathcal{L}(), p^1$. For convenience the compact variable $\lambda = \{f(), \mathcal{L}(), p^1\}$ is defined. In the next section, a brief review of the three fundamental problem for HMM identification is presented. In the section 8.2, it is shown how one can extend these problems for SED identification.

8.1.1 Evaluation of an Observation Sequence Probability

The first problem is how to compute the probability of an observation sequence, $O = \{z^1, \cdots, z^T\}$, given λ?

Consider $h \in \mathbb{H}$ where $\mathbb{H}$ is the all possible state sequence space. The probability of O given λ is computable by marginalizing joint probability over state sequence space $\mathbb{H}$.

$$p(O|\lambda) = \sum_{h \in \mathbb{H}} p(O, h|\lambda) = \sum_{h \in \mathbb{H}} p(O|h, \lambda) p(h|\lambda) \tag{8.1}$$

where

$$p(O|h, \lambda) = \prod_{k=1}^{T} \mathcal{L}^k(z^k|x^k) \tag{8.2}$$

$$p(h|\lambda) = p^1(x_1) \prod_{k=2}^{T} f(x^k|x^{k-1}) \tag{8.3}$$

8.1.2 Inferring Optimal State Sequence

Depending on the definition of the optimal state sequence, there are several ways of finding an optimal sequence. One possible definition is to find a state sequence, which is maximizing $p(h, O|\lambda)$.

$$\hat{h} = \arg \max_{x^1, \cdots, x^T} p(h, O|\lambda) \tag{8.4}$$

There are various filtering algorithms for estimating highest probable sequence $\hat{h}$. Veterbi algorithm [73] and Kalman filter [33] are two well developed methods applied in many real world applications.

8.1.3 Maximum Likelihood Estimation (Learning λ)

The third problem is about how to choose λ such that its likelihood, $\mathcal{L}(\lambda|O)$, is maximized. Generally, there is no analytical solution , however one can use iterative procedures (e.g.expectation-maximization (EM) [15]) or gradient techniques. Ghahramani and Hinton [20] have proposed an EM algorithm for linear dynamic systems, and Roweis and Ghahramani [57] have proposed an EM algorithm for nonlinear dynamic systems. The EM algorithm for linear systems is described in details in the section 8.4.

8.2 Stochastic Evolution Diagram (SED) Identification

In this section, it is shown how to identify stochastic evolution diagram (SED). All parameters and variables of SED as a model are divided into three compact variables. Let $\theta = \{p_S, p_D, c\}$ indicates the probability of survival, detection, and false alarm rate (Figures 4.3, 4.4 and 5.1(c)). $\lambda = \{f(), \mathcal{L}^k()\}$ indicates all the other parameters of motion and observation models, and $\mathcal{S}$ indicates all hidden variables which represents structure of the stochastic evolution diagram (hidden states of virtual leaders). Let

$h = \{\mathcal{S}, \theta\}$, and denote the stochastic evolution diagram model as $\Psi = \{h, \lambda\}$. Then,

$$\Psi = \{h, \lambda\} \tag{8.5}$$

$$h = \{\mathcal{S}, \theta\} \tag{8.6}$$

$$\theta = \{p_S, p_D, c\} \tag{8.7}$$

$$\mathcal{S} = \{\mu_i^k | i < \hat{M}^k, 1 \leq k \leq T\} \tag{8.8}$$

$$\lambda = \{f(), \mathcal{L}()\} \tag{8.9}$$

Analogous to HMM, for our model, we aim to solve the following three problems: 1) what is the probability of an observation sequence $\mathcal{Z}^{1:T}$ given λ, $p(\mathcal{Z}^{1:T}|\lambda)$; 2) what is the most probable SED state given $\mathcal{Z}^{1:T}$ and λ, $p(\mathcal{S}|\lambda, \mathcal{Z}^{1:T})$; and 3) how can one learn λ to maximize the likelihood $\mathcal{L}(\lambda|\mathcal{Z}^{1:T})$.

We wish to calculate the log likelihood function of an observation sequence, $\mathcal{L}(\lambda|\mathcal{Z}^{1:T})$. It can be done by marginalizing joint probability distribution of $\mathcal{Z}^{1:T}$ and h in following way:

$$\mathcal{L}(\lambda|\mathcal{Z}^{1:T}) = \ln \int_{\mathbb{H}} p(h, \mathcal{Z}^{1:T}|\lambda) \partial h \tag{8.10}$$

where $\mathbb{H}$ is the evolution diagram space ($h \in \mathbb{H}$) . Typically it is difficult to compute analytically the marginal likelihood. Below we describe the solution for the introduced problems and show how to maximize the log likelihood function. We followed the same approach that [57] used for identification of nonlinear dynamic systems.

Assume $\mathcal{Q}(h)$ is the distribution of the hidden variables h. Using concavity property of the log function and Jensen's inequality, we will have:

$$\ln \int_{\mathbb{H}} p(h, \mathcal{Z}^{1:T}|\lambda)\partial h = \ln \int_{\mathbb{H}} \mathcal{Q}(h)\frac{p(h, \mathcal{Z}^{1:T}|\lambda)}{\mathcal{Q}(h)}\partial h \tag{8.11}$$

$$\geqslant \int_{\mathbb{H}} \mathcal{Q}(h) \ln \frac{p(h, \mathcal{Z}^{1:T}|\lambda)}{\mathcal{Q}(h)}\partial h \tag{8.12}$$

$$= \int_{\mathbb{H}} \mathcal{Q}(h) \ln p(h, \mathcal{Z}^{1:T}|\lambda)\partial h - \int_{\mathbb{H}} \mathcal{Q}(h) \ln \mathcal{Q}(h)\partial h \tag{8.13}$$

$$= \mathcal{F}(\mathcal{Q}, \lambda) \tag{8.14}$$

We apply the *expectation-maximization* algorithm [15] to maximize $\mathcal{F}$ as a lower bound of log likelihood function $\mathcal{L}$. In the EM algorithm, the objective function has two compact variables (h and λ) and the method alternates between two steps to maximize the objective function with respect to the two compact variables, respectively, by holding the other one fixed. The algorithm starts from a initial parameter λ_0 and iteratively applies the following steps:

$$\textbf{E-step:} \qquad \mathcal{Q}_{i+1} \leftarrow \arg\max_{\mathcal{Q}} \mathcal{F}(\mathcal{Q}, \lambda_i) \tag{8.15}$$

$$\textbf{M-step:} \qquad \lambda_{i+1} \leftarrow \arg\max_{\lambda} \mathcal{F}(\mathcal{Q}_{i+1}, \lambda) \tag{8.16}$$

The following subsections describe calculating each of these.

8.3 E-step: Learning the Structure (h)

The function $\mathcal{F}$ is at its maximum in E-step when $\mathcal{Q}(h) = p(h|\mathcal{Z}^{1:T}, \lambda) = p(\mathcal{S}, \theta|\mathcal{Z}^{1:T}, \lambda)$ [57]. In stationary dynamic systems, the problem of estimating the state of hidden variables to maximize the E-step of EM algorithm corresponds exactly to the smoothing or filtering problems [57]. If we assume that θ is given, one can apply filtering to infer $\mathcal{S}$. Due to dependency of $\mathcal{S}$ on θ, we have

$$p(\mathcal{S}, \theta | \mathcal{Z}^{1:T}, \lambda) = p(\mathcal{S} | \theta, \mathcal{Z}^{1:T}, \lambda) \times p(\theta) \tag{8.17}$$

By applying model selection methods and searching in space Θ, where $\theta \in \Theta$, one can maximize $\mathcal{Q}$ by maximizing the products on the right side of Equation 8.17. In section 8.3.2 three model selection criteria are provided. In contrast to the EM algorithm for identification of dynamic systems that the E-step is just an inference problem, here in our case it is a model selection problem that includes inferring the structure of the evolution diagram as well.

Consider $\mathcal{S} = \mathcal{N}et^{1:T} = \{\mathcal{N}et^1, \ldots, \mathcal{N}et^T\}$. We should emphasize again that although variable $\mathcal{N}et$ stands for a network, but it denotes a random finite set of modules' virtual leaders and just considers nodes. In the next chapter, it is shown how to include the edges in the model. Here, we just show how to estimate number of modules and their members at each time step. Given λ and θ, finding the fittest structure $\mathcal{S}$ that has generated the observation, has serval solutions. But first the optimality criteria should be declared. We wish to find a sequence of network state that maximizes $p(\mathcal{N}et^1, \cdots, \mathcal{N}et^T | \mathcal{Z}^{1:T}, \lambda, \theta)$:

$$\widehat{\mathcal{N}et}^{1:T} = \{\mathcal{N}et^{1:T} | \arg \max_{\mathcal{N}et^1, \cdots, \mathcal{N}et^T} p(\mathcal{N}et^1, \cdots, \mathcal{N}et^T | \mathcal{Z}^{1:T}, \lambda, \theta)\} \tag{8.18}$$

8.3.1 Inferring the Stochastic Evolution Diagram's Structure

The problem of inferring the stochastic evolution diagram's structure given λ and θ is the problem of estimating fittest sequence of a random finite set state. This problem is discussed in details in chapters 5 and 6. The main two methods used for approximation of the PHD filter are the SMC-PHD filter [76] and the Gaussian

Mixture PHD filter (GM-PHD) [75]. Both methods can be applied as an approximate solution for estimation of the structure $\mathcal{S}$ of the evolution diagram.

For a RFS $\mathcal{N}et^k$ with a intensity function (PHD) υ, the integral of υ^k over the network state space gives the expected number of modules. Hence, $\hat{M}^k = \int \upsilon^k(\mu)\partial\mu$ ($\hat{M}^k$ denotes estimated number of modules at time k). The local maxima of the intensity υ are the highest local concentration of the modules, and hence the $[\hat{M}^k]$ highest peaks from the intensity function can be selected as the estimation of the state of each module ([.] means rounded number).

Given the Gaussian mixture intensities $\upsilon^{k|k-1}$ and υ^k, the corresponding expected number of virtual leaders (number of modules) $\hat{M}^{k|k-1}$ and $\hat{M}^k$ can be obtained by summing up the appropriates weights [75]:

$$\hat{M}^{k|k-1} = \hat{M}^{k-1}\left(p_S^k + \sum_{j=1}^{J_\beta^k} \omega_{\beta,j}^k\right) + \sum_{j=1}^{J_\gamma^k} \omega_{\gamma,j}^k \tag{8.19}$$

$$\hat{M}^k = \hat{M}^{k|k-1}(1 - p_D^k) + \sum_{z \in Z^k} \sum_{j=1}^{J^{k|k-1}} \omega_j^k(z) \tag{8.20}$$

The PHD filter gives the expected location of the virtual leaders. One of the main criticisms of the PHD filter is that there is no means of associating the same virtual leader between time frames. But this is an advantage for our problem, because our main concerns are the number of virtual leaders and their states. It is trivial to identify the trajectories of different modules by looking into overlaps between modules' members.

Thus far, we have shown how to estimate the structure of the evolution diagram. To summarize, the structure is the life time of parameters of a network, and the corresponding virtual leader. Life-time parameters of a network such as the module's birth time, death time, spawning and merging times are considered as change points. The SMC-PHD and GM-PHD filter implementations are described in chapter 6.

8.3.2 Learning θ

We followed a model selection approach to find θ that maximizes locally the log likelihood function given λ. Any information criterion techniques applicable for mixture models [59, 62, 83] such as BIC, AIC and DIC can be employed to penalize the model complexity. For example, assume AIC^k is an defined AIC score function for mixture of modules at time k. The score assigned to a structure for a specified θ can be obtained by averaging AIC^k over the entire time.

8.4 M-step: Learning Model Parameters (λ)

The structure of an evolution diagram is a directed graph, which is decomposable to its pathes $\mathcal{S}_E$. A path $q \in \mathcal{S}_E$ is a Markov chain from a root to a leaf. Roots are nodes with zero indegree, and leaves are nodes with zero outdegree. In other words, roots are birth times of modules (t_b) and leaves are death times (t_d). A path q is a sequence of hidden states of the virtual leaders of modules. For example, $q = \{\mu^{t_b}, \mu^{t_b+1}, \ldots, \mu^{t_d}\}$. It has discussed in section 8.1.3 a likelihood function for a Markov chain such as $q = \{\mu^{t_b}, \mu^{t_b+1}, \ldots, \mu^{t_d}\}$ if there are an observation from each hidden variable ($p_D = 1$). So, if we consider the missed observation case ($p_D \leq 1$)

the likelihood function for a Markov chain will be generalized as follows:

$$\mathcal{L}_q(\lambda|\mathcal{Z}^{1:T}) = \ln\left(p(\mu^{t_{b,q}}) \prod_{k=t_{b,q}+1}^{t_{d,q}} f(\mu_q^k|\mu_q^{k-1}) \prod_{k=t_{b,q}}^{t_{d,q}} \mathcal{L}^k(\mathcal{Z}^k|\mu_q^k)\right) \tag{8.21}$$

$$\mathcal{L}_q(\lambda|\mathcal{Z}^{1:T}) = \ln\left(p(\mu_q^{t_{b,q}}) \prod_{k=t_{b,q}+1}^{t_{d,q}} f(\mu_q^k|\mu_q^{k-1}) \prod_{k=t_{b,q}}^{t_{d,q}} \overbrace{\Big(\prod_{x_i^k\in o_q(mo^k)} \dot{\mathcal{L}}^k(\dot{z}^k|x_i^k)\Big)^{\eta^k}}^{\mathcal{L}^k()}\right) \tag{8.22}$$

$$\mathcal{L}_q(\lambda|\mathcal{Z}^{1:T}) = \ln\left(p(\mu_q^{t_{b,q}}) \prod_{k=t_{b,q}+1}^{t_{d,q}} f(\mu_q^k|\mu_q^{k-1}) \prod_{k=t_{b,q}}^{t_{d,q}} \Big(\prod_{x_i^k\in o_q(mo^k)} \dot{\mathcal{L}}^k(\dot{z}^k|\mu_q^k)\Big)^{\eta^k}\right) \tag{8.23}$$

$$\mathcal{L}_q(\lambda|\mathcal{Z}^{1:T}) = \ln\left(p(\mu_q^{t_b}) \prod_{k=t_{b,q}+1}^{t_{d,q}} f(\mu_q^k|\mu_q^{k-1}) \prod_{k=t_{b,q}}^{t_{d,q}} \mathcal{L}^k(H|o_q(mo^k))\right) \tag{8.24}$$

$$\mathcal{L}_q(\lambda|\mathcal{Z}^{1:T}) = \ln(p(\mu^{t_{b,q}})) + \sum_{k=t_{b,q}+1}^{t_{d,q}} \ln(f(\mu_q^k|\mu_q^{k-1})) + \sum_{k=t_{b,q}}^{t_{d,q}} \ln(\mathcal{L}^k(H|o_q(mo^k)) \tag{8.25}$$

A virtual leader's observation model ($\mathcal{L}^k()$) is defined as the average of nodes' log-likelihoods for all those nodes which are detected and also belong to the module that is leaded by the virtual leader μ_q^k (Equation 8.22). The equation 8.22 is the likelihood function when we are interested in to estimate the virtual leaders's state and nodes's state both, $\eta_k = 1/|o_q(mo^k)|$ is a normalization factor, function $o_q(mo^k)$ returns elements with available observations, which belong to a module on path q at time k. $|.|$ returns the cardinality of a set. As we simplified the problem in section 5.3, we don't estimate nodes' state, and just approximate them by their virtual leader's state ($x_i^k \simeq \mu_q^k$), so the Equation 8.22 will be simplified to Equation 8.23.

If we assume that the node's observation is a linear Gaussian (in case of detection), it is possible to replace $\dot{\mathcal{L}}^k()$ by Equation 6.2. $\mathcal{L}^k(H|o_q(mo^k)$ in Equation 8.24 denotes the virtual leader's observation model.

$$\mathcal{L}^k(H|o_q(mo^k) = (\prod_{x_i^k \in o_q(mo^k)} \dot{\mathcal{L}}^k(\dot{z}^k|x_i^k))^{\eta^k} \tag{8.26}$$

$$\dot{\mathcal{L}}^k(\dot{z}|x) \simeq \mathcal{N}(\dot{z}; H\mu_q^k, R_k) \tag{8.27}$$

Assume $\widehat{\mathcal{L}}$ is the likelihood function of the stochastic evolution diagram.

$$\widehat{\mathcal{L}}(\lambda|\mathcal{S}_E, \mathcal{Z}) = E[\mathcal{L}_q(\lambda|\mathcal{Z})] = \sum_{q \in \mathcal{S}_E} \mathcal{L}_q(\lambda|\mathcal{Z}) f_\lambda(q) \tag{8.28}$$

$$\widehat{\mathcal{L}}(\lambda|\mathcal{S}_E, \mathcal{Z}) = \sum_{q \in \mathcal{S}_E} f_\lambda(q) \left(\sum_{k=t_{b,q}+1}^{t_{d,q}} \ln(f(\mu^k|\mu^{k-1})) + \sum_{k=t_{b,q}}^{t_{d,q}} \ln(\mathcal{L}^k(H|o_q(mo^k))) + \ln(p(\mu^{t_{b,q}})) \right) \tag{8.29}$$

It is defined as expectation of SED all pathes' likelihood. In simpler words, it is weighted sum of all pathes' likelihood. $f_\lambda(q)$ is a probability distribution associated with each path:

$$f_\lambda(q) = \frac{\parallel q \parallel}{\sum_{q_i \in \mathcal{S}_E} \parallel q_i \parallel} \tag{8.30}$$

$\parallel q_i \parallel$ returns the longevity of the life time of path q_i.

Based on Equations (6.1) and (6.2), one can drive the conditional densities for the transition, observation and initial state as follows:

$$\ln(f(\mu^k|\mu^{k-1})) = -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln(|Q|) - \frac{1}{2}[\mu^k - F\mu^{k-1}]' Q^{-1}[\mu^k - F\mu^{k-1}] \tag{8.31}$$

$$\ln(\mathcal{L}^k(H|o_q(mo^k))) = p(z^k|\mu^k) = -\frac{d'}{2}\ln(2\pi) - \frac{1}{2}\ln(|R|) - \frac{1}{2}[z^k - H\mu^k]' R^{-1}[z^k - H\mu^k] \tag{8.32}$$

$$\ln(p(\mu^{t_b})) = -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln(|P_{t_b}|) - \frac{1}{2}[\mu^{t_b} - F\mu^0]' P_{t_b}^{-1}[\mu^{t_b} - F\mu^0] \tag{8.33}$$

Ghahramani and Hinton [20] have shown how to estimate F, H, Q, R and V for a linear dynamic system by setting to zero the corresponding partial derivative of the expected log likelihood. It is assumed that each path is a linear dynamic system, hence we have:

$$\frac{\partial \mathcal{L}_q}{\partial H} = -\sum_{k=t_{b,q}}^{t_{d,q}} R^{-1} Z^k \mu^{k^{'}} + \sum_{t_{b,q}}^{t_{d,q}} R^{-1} H P^k = 0 \tag{8.34}$$

$$\widehat{H}_q = \left(\sum_{k=t_{b,q}}^{t_{d,q}} z^k \mu^{k'} \right) \left(\sum_{k=t_{b,q}}^{t_{d,q}} P^k \right)^{-1} \tag{8.35}$$

$$= \left(\sum_{k=t_{b,q}}^{t_{d,q}} \eta^k \sum_{x_i^k \in o_q(mo^k)} \dot{z}^k \mu_q^{k^{'}} \right) \left(\sum_{k=t_{b,q}}^{t_{d,q}} P^k \right)^{-1} \tag{8.36}$$

$$\widehat{H}_{\mathcal{S}_E} = \sum_{q \in \mathcal{S}_E} \widehat{H}_q f_\lambda(q) \tag{8.37}$$

where $\widehat{H}_q$ is the new estimated observation matrix based only path q. $\mu^{'}$ denotes μ transpose. $\widehat{H}_{\mathcal{S}_E}$ is the new estimated observation matrix based on the structure of $\mathcal{S}_E$. We can follow the same fashion for $\frac{\partial \mathcal{L}_q}{\partial R^{-1}}$, $\frac{\partial \mathcal{L}_q}{\partial F}and\frac{\partial \mathcal{L}_q}{\partial Q^{-1}}$:

$$\widehat{R}_q = \frac{1}{t_{d,q} - t_{b,q}} \sum_{k=t_{b,q}}^{t_{d,q}} (z^k z^{k'} - \widehat{H}_q \mu_q^k z^{k'}) \tag{8.38}$$

$$\widehat{R}_{\mathcal{S}_E} = \sum_{q \in \mathcal{S}_E} \widehat{R}_q f_\lambda(q) \tag{8.39}$$

$$\widehat{F}_q = \left(\sum_{k=t_{b,q}+1}^{t_{d,q}} P^{k|k-1} \right) \left(\sum_{k=t_{b,q}+1}^{t_{d,q}} P^{k-1} \right)^{-1} \tag{8.40}$$

$$\widehat{F}_{\mathcal{S}_E} = \sum_{q \in \mathcal{S}_E} \widehat{F}_q f_\lambda(q) \tag{8.41}$$

$$\widehat{Q}_q = \frac{1}{t_{d,q} - t_{b,q} - 1} \left(\sum_{k=t_{b,q}+1}^{t_{d,q}} P^k - \widehat{F}_q \sum_{k=t_{b,q}+1}^{t_{d,q}} P^{k|k-1} \right) \tag{8.42}$$

$$\widehat{Q}_{\mathcal{S}_E} = \sum_{q \in \mathcal{S}_E} \widehat{Q}_q f_\lambda(q) \tag{8.43}$$

One can directly take partial derivative from $\widehat{\mathcal{L}}$

$$\widehat{\mathcal{L}}(\lambda|\mathcal{S}_E, \mathcal{Z}) = \sum_{q\in\mathcal{S}_E} f_\lambda(q) \sum_{k=t_{b,q}+1}^{t_{d,q}} \ln(f(\mu_q^k|\mu_q^{k-1})) + \sum_{q\in\mathcal{S}_E} f_\lambda(q) \sum_{k=t_{b,q}}^{t_{d,q}} \sum_{x_i^k\in mo_q^k} \ln(\mathcal{L}^k(z^k|\mu_q^k))+ \sum_{q\in\mathcal{S}_E} \sum_{k=t_{b,q}}^{t_{d,q}} \ln(\eta^k) f_\lambda(q) + \sum_{q\in\mathcal{S}_E} \ln(p(\mu^{t_{b,q}})) f_\lambda(q) \tag{8.44}$$

In the next two following chapters, theoritical and practical application of stochastic evolotion diagram (SED) will be dicussted. In chapter 9, it is shown how teoritically extend idea of Bayesian network for non-stationary dynamical system. In chapter 10, direct application of SED in high dimentional nonstationary dynamical systems is presented.

CHAPTER 9

EVOLUTION OF BAYESIAN NETWORKS UNDERLYING COMPLEX SYSTEMS

Several methods have been recently proposed to infer dynamic topology of different types of networks (Bayesian network, Gaussian Graphical Models, etc.). Depending on the assumptions about the system of interest, and also the type of the network, one can have different models. Herein, we show how to reconstruct the sequence of the Bayesian networks from a time series produced by an underlying random cluster processes system.

Before describing how to reconstruct the underlying Bayesian networks, one needs to answer the following question: what is the relationship between the finite set statistics (FISST) and the conventional probability? The importance of this question is determined by the fact that the FISST is based on the belief mass function.Vo et al. [76] have shown that "the set derivative of a belief mass function of a RFS is closely related to its probability density". This relationship allows us to factorize the belief mass function of a RFS to its Bayesian network form.

9.1 Preliminaries

A Bayesian network (BN) describes a unique joint probability distribution over a fixed number of variables of a static system. The mathematical representation of a Bayesian network B given the graph G and parameters Θ is as follows:

$$p(x_1, \cdots, x_n) = \prod_{i=1}^{n} p(x_i | \pi(x_i)) \tag{9.1}$$

where $X = \{x_1, \cdots, x_n\}$ denotes the set of variables, p denotes joint probability distribution, G is a Directed Acyclic Graph (DAG) whose nodes correspond to X components, $\pi(x_i)$ is the set of parents of x_i, and Θ represents the set of parameters that quantify the graph. Formally, a BN for X is a pair $B = (G, \Theta)$.

The Dynamic Bayesian network (DBN) is the extension of BN to model temporal processes. To present the idea of Bayesian networks for time series data, we should obtain the joint probability distribution over the random variables $\{X^1 \cup \cdots \cup X^T\}$, where $X^k = \{x_1^k, \cdots, x_n^k\}$ and T is the number of time samples; in other words, we need to factorize $p(x_1^1, \cdots, x_n^T)$. Apparently, such a distribution is high-dimensional and extremely complex.

By assuming that the temporal process is the first order *Markovian*, p can be factorized in the following way:

$$p(X^1, \cdots, X^T) = \prod_{k=1}^{T} p(X^k | X^{k-1}) \tag{9.2}$$

It also is assumed that the process is *stationary*, which means that $\forall k_1, k_2 \leqslant T \;\; p(X^{k_1} | X^{k_1 - 1}) = p(X^{k_2} | X^{k_2 - 1})$. Under this condition, $p(X^k | X^{k-1})$ can also be decomposed into the Bayesian network form:

$$p(X^k|X^{k-1}) = \prod_{i=1}^{n} p(x_i^k|\pi(x_i^k)) \tag{9.3}$$

We then obtain a DBN model from equations (9.2) and (9.3) in the form:

$$p(\underbrace{x_1^1, \cdots, x_n^T}_{n \times T \text{ variables}}) = \prod_{k=1}^{T} \prod_{i=1}^{n} p(x_i^k|\pi(x_i^k)) \tag{9.4}$$

where $\pi(x_j^1) = \phi$. The term "dynamic" in DBN does not mean that the topology of BN evolves over time. Instead, it only emphasizes that the underlying Bayesian network of a dynamic system is under assumptions of stationarity and the first order Markovian process.

9.2 The Fittest Sequence of BNs

Assume an unknown non-stationary process that generates a multivariate time series data Z of n random variables for T discrete time points. In this context, the term "non-stationary process" means that the conditional dependency between random variables is *partially stationary* and changes over time. The researcher aims at learning a sequence of Bayesian networks $\mathbf{G} = \{Net_1 \cdots Net_\Phi\}$, not only one Bayesian network. Let us assume that Net_ϕ can be replaced by $Net_{\phi+1}$ if a *network phase transition* occurs. As was described in chapter (4.2), when a network phase transition happens in a system, it can affect only some modules of the network or even all of them, and then we can expect to have topology change only in the affected modules. A sequence of networks can have two graphical representations: first, as a sequence of Bayesian networks, where each network represents the topology of BN in one specific phase; second, as one network with labeled edges, where the label of each

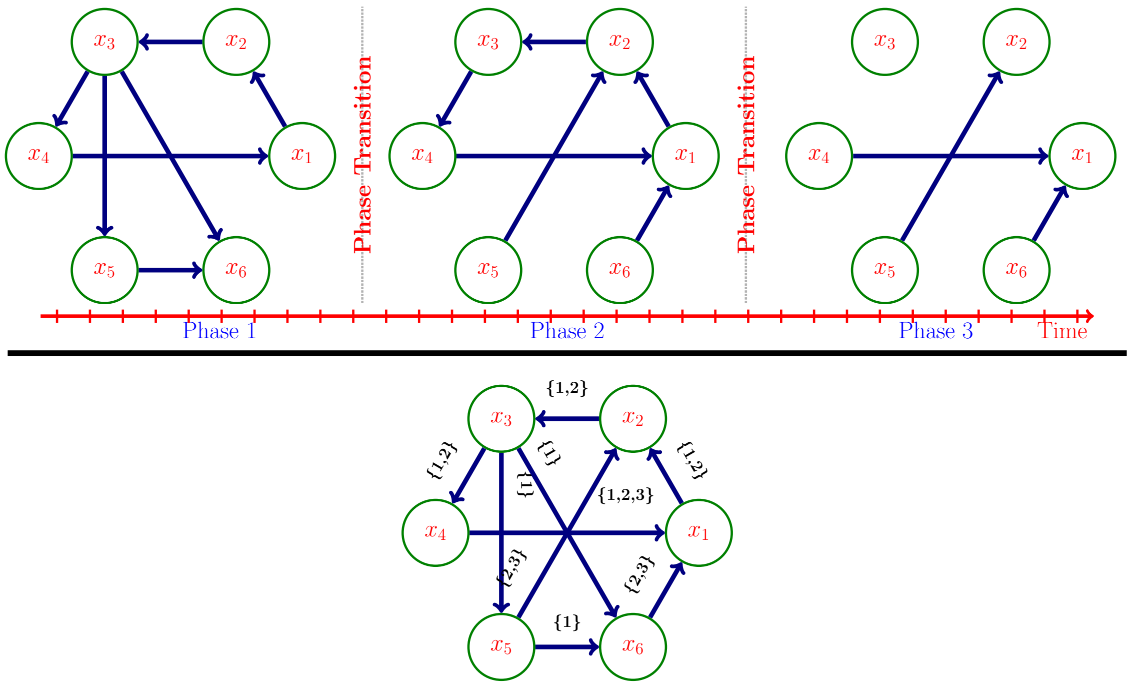


Figure 9.1: An example of two graphical representations of evolution in topology of networks.

edge represents the lifetime of that edge. Figure 9.1 demonstrates the evolution of a simple Bayesian network using these two approaches.

9.3 Modeling

Just as we have extended the BN for Markovian and stationary dynamic systems to the DBN, in this section we extend the BN for non-stationary time series produced from complex systems. To do this, we generalize the first order Markov process assumption to random cluster processes, and instead of reconstructing the multivariate Markov chains, we reconstruct multiple set Markov chains (chapter 7). Then we can factorize the complete joint probability distribution of all variables and the phase as:

$$p(x_1^1, \cdots, x_n^T, phase|Z) = p(x_1^1, \cdots, x_n^T|phase, Z)p(phase|Z) \tag{9.5}$$

where $p(phase|Z)$ is the phase distribution. The stochastic evolutionary diagram which is defined in chapter 7 is able to provide the phase distribution.

According to the definition of complex systems, the joint probability distribution can be factorized to components of a complex system as follows:

$$p(x_1^1, \cdots, x_n^T|phase, Z) = \prod_{l=1}^{N_c} p(C_l) \tag{9.6}$$

where N_c is the total number of random cluster processes. C_l is the set of all random variables that belong to the lth random cluster process. In other words, $C_l = \{x_{i'}^{k'}|1 \leqslant i' \leqslant n, 1 \leqslant k' \leqslant T, x_{i'}^{k'} \in mo_l^{k'}\}$, and by assuming that each component (process) is a *random cluster process*, we will have:

$$p(C_l) = \prod_{k=1}^{T} p(mo_l^k|mo_l^{k-1}) \tag{9.7}$$

Then from Equations (9.6) and (9.7)

$$p(x_1^1, \cdots, x_n^T|phase, Z) = \prod_{k=1}^{T}\prod_{l=1}^{N_c} p(mo_l^k|mo_l^{k-1}) \tag{9.8}$$

Also, if we assume that each random cluster process is *stationary* during its life time, $p(C_l)$ can be factorized in the Bayesian network form:

$$p(C_l) = \prod_{k=1}^{T}\prod_{i=1}^{n} \mathbf{p}^l(x_i^k|\pi^l(x_i^k)) \tag{9.9}$$

where

$$\mathbf{p}^l(x_i^k|\pi^l(x_i^k)) = \begin{cases} p(x_i^k|\pi^l(x_i^k)) & x_i^k \in mo_l^k \\ 1 & otherwise \end{cases} \tag{9.10}$$

and $\pi^l(x_i^k) \subset mo_{k-1}^{\sigma_1(l)}$, where $\sigma_1(l)$ is an index function; it returns the index of a module at time k which belongs to the lth random cluster process. π^l is a subset of the parent configuration of x_i^k.

Equation (9.9) is similar to Equation (9.4) but stands for a random cluster process. For all the random cluster processes, we will have

$$p(x_1^1, \cdots, x_n^T | phase, Z) = \prod_{k=1}^{T} \prod_{l=1}^{N_c} \prod_{i=1}^{n} \mathbf{p}^l(x_i^k | \pi^l(x_i^k)) \tag{9.11}$$

$$= \prod_{k=1}^{T} \prod_{m=1}^{\hat{M}^k} \prod_{x_i^k \in mo_m^k} p(x_i^k | \pi^l(x_i^k)) \tag{9.12}$$

where $\hat{M}^k$ is the number of alive modules at time k (as defined in the section 6.3). Equations (9.11) and (9.12) are equal, but the Equation (9.12) is computationally more efficient. In a special case, when there is no phase transition and all random variables belong to a module, Equation (9.12) will be equal to Equation (9.4). This shows that a Markovian and stationary dynamic system is a simplified version of our complex systems of our interest, and the presented way of factorizing the joint probability distribution generalizes the DBN.

9.4 Bayesian Dirichlet (BD) Metric

Figure 9.1 shows that it is possible to present a sequence of Bayesian networks by a labeled-edge Bayesian network. Let the triple $\mathbf{G} = (G_T, G_P, \Psi)$ parameterize a labeled-edge Bayesian network, in which G_T is a DAG denoting the Bayesian network topology, and G_P is a vector whose values denote the conditional probability assignments associated with the Bayesian network topology G_T [14]. Ψ contains all information encapsulated in the stochastic evolution diagram. In this section, similar to works of [28] and [14], we present a metric for evaluating the probabilities of different DAGs given the *discrete data* and the stochastic evolution diagram.

Suppose the variables $X = \{x_1^1, \ldots, x_n^T\}$ is a set of n discrete features observed during T time samples, where a feature x_i can have s_i possible states or values: $\{v_{i,1}, \ldots, v_{i,s_i}\}$. Let $\mathcal{D}$ denote discretized Z and w_i denote a list of the unique instantiations for parents of x_i as seen in $\mathcal{D}$, and w_{ij} denote the jth unique instantiation of π_i relative to $\mathcal{D}$, and there are q_i such unique instantiations of π_i (we are following the notation of [14]). Let π_i^l be a set of elements of π_i, which also belong to the cluster process l; then $\pi_i^l \subset \pi_i$, and let w_{ij}^l denote the instantiation of π_i^l. N_{ijs}^l is defined as a number that is proportionate to cases in the time series in which the feature $x_i \in C_l$ holds on the value $v_{i,s}$, and π_i^l is instantiated as w_{ij}^l. N_{ijs}^l is formulated as follows:

$$N_{ijs}^l = \frac{1}{|\pi_i|} \sum_{k=2}^{T} \Delta^k \tag{9.13}$$

where

$$\Delta^k = \begin{cases} |w_{ij}^l| & \pi_i^l = w_{ij}^l \\ 0 & otherwise. \end{cases} \tag{9.14}$$

and $|.|$ means the cardinality of the set. The expression $\pi_i^l = w_{ij}^l$ means that π_i^l is instantiated as w_{ij}^l.

If we are able to obtain $p(G_T, \mathcal{D}|\Psi)$, we can rank the probabilities of different topologies. A BD metric of the likelihood function $p(D|G_T, G_P)$ can be proved by marginalizing $p(G_T, G_P, \mathcal{D}|\Psi)$ as follows:

$$p(G_T, \mathcal{D}|\Psi) = p(G_T|\Psi) \int p(\mathcal{D}|G_T, G_P) f(G_P|G_T, \Psi) dG_P \tag{9.15}$$

The closed form expression for the BD metric is obtained as:

$$
\begin{aligned}
p(\mathcal{D}|G_T, \Psi) = \prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} & \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij}^{l} + \alpha_{ij})} \\
& \prod_{s=1}^{s_i} \frac{\Gamma(N_{ijs}^{l} + \alpha_{ijs})}{\Gamma(\alpha_{ijs})} \qquad (9.16)
\end{aligned}
$$

where $N_{ij}^{l} = \sum_{s=1}^{s_i} N_{ijs}^{l}$, α_{ijs}s are Dirichlet hyper-parameters of prior probability distribution of the DAG topology, $\alpha_{ij} = \sum_{s=1}^{s_i} \alpha_{ijs}$, and $\Gamma(.)$ is a gamma function. The proof is provided in appendix A

CHAPTER 10

RESULTS ON SIMULATED AND REAL DATASETS

In this chapter, we examine the applicability, accuracy, and performance of the presented identification method of the stochastic evolution diagram. The proposed methods in chapter 8 to infer and learn model parameters are tested on simulated examples. The method was applied first to a simple example of a simulated complex system with 150 features over 400 time steps, then the performance evaluated on 100 randomly generated scenarios. Furthermore, inferring the structure of the evolution diagram given the parameters θ and λ also applied to a time course gene expression data set obtained from a cell culture model of TGF-β-induced epithelial-mesenchymal transition (EMT) [58]

I would like to emphasize that the main goal of this work was to formulate the framework for evolutionary dynamics of complex systems, show how to identify the system theoretically by defining the inference and learning problems and show that the stochastic evolution diagram is obtainable mathematically.

10.1 Simulated Dataset

The ith feature at time k has been defined by a state vector $x_i^k = [s_i^k, v_i^k]$, where $s_i^k \in \{active, inactive\} \times [0, 100]$ space (a feature is active when it belongs to a module) and $v_i^k \in [0, 2]$ represents a kinetic variable (e.g., velocity). The measurement has been denoted by $z_i^k \in \{\emptyset \cup [0, 100]\}$ ($\emptyset$ means there is no measurement from the feature).

For simplicity, I assume there is a maximum one spawning module birth, and four spontaneous module births at each time step. The birth process for actual modules is a Poisson RFS with a Gaussian mixtures intensities. The false alarm rate for observations is modeled as a Poisson RFS K^k with intensity $\kappa^k(z) = \lambda_c V u(z)$, where $u(.)$ is the uniform density over $[0, 100]$ space, and $\lambda_c = 10 \times 10^{-2}$ is the average number of detected false alarm modules per unit per observation that relates to 10 false alarms returned over the observation space. $V = 100$ is the volume of the observation space.

Figure 10.1 (top left) shows true trajectories of modules' virtual leaders . Figure 10.1 (down left) illustrates some signals that cause birth of a module at time 150 (emergence), and Figure 10.1 (top right) shows signals that split from a module to a smaller module at time 100 and then die at time 250. A signal is plotted in two colors, green color means the feature belongs to a module and it is active, and gray color means the feature is inactive. Figure 10.1 (down right) shows a noisy observation from some features.

As the number of components in the posterior intensity (in GM-PHD filter) can increase without limit, it is necessary to define some threshold to make the algorithm computationally faster. I defined a truncation threshold $\mathfrak{T}$ to discard Gaussian components with weak weights and also have set a threshold $\mathfrak{J}_{max}$ for maximum allowable number of Gaussian terms in the posterior intensities. Moreover, I introduced a threshold $\mathfrak{r}$ to use as a radius of a region where components should be merged to a single component. The threshold $\mathfrak{M}_S$ is the minimum number of elements of a set to

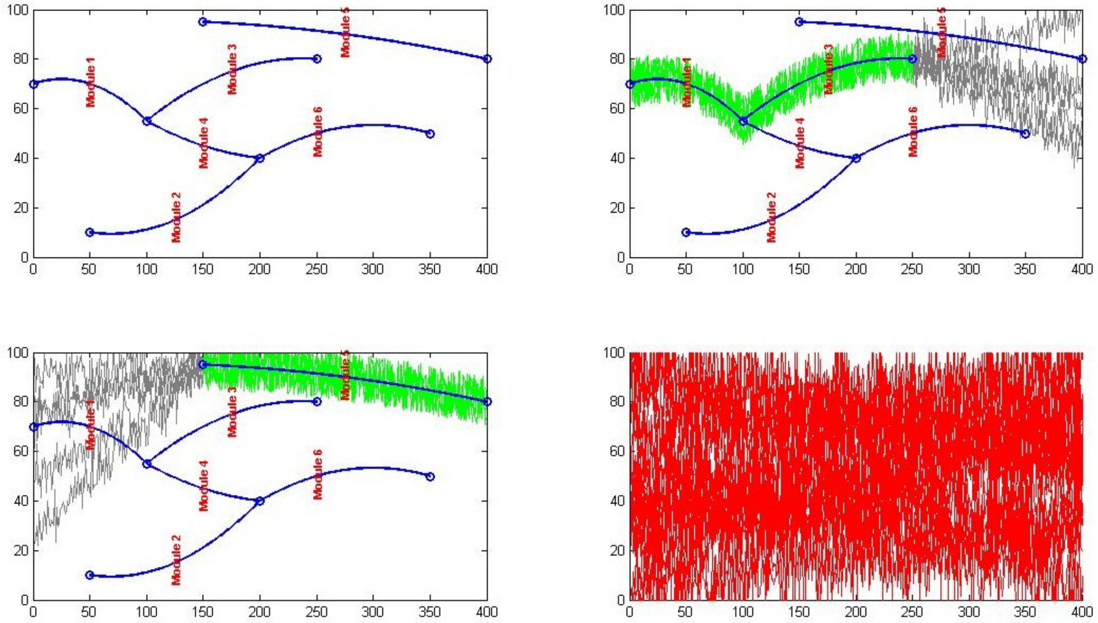


Figure 10.1: Example scenario examined in the simulation study (top left). Emergence of a module (bottom left). Split and death of a module (top right). Observation signals (bottom right). Green color shows that a signal is active, while grey color shows that it is not. Blue lines are trajectories of virtual leaders. Red line represent noisy observed data. The horizontal axis is time and the vertical axis is the state space.

consider it as a module. In this example thresholds are set as $\mathfrak{T} = 10^{-5}$, $\mathfrak{J}_{max} = 20$, $\mathfrak{r} = 5$, and $\mathfrak{M}_S = 20$.

Features and also virtual leaders follow the linear Gaussian dynamics (Equation 6.1) and the observation model (Equation 6.2) with

$$F = \begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0.06 & 0.12 \\ 0.12 & 0.25 \end{bmatrix}, \qquad H = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad R = 4 \tag{10.1}$$

where $\Delta = 1$ is the sampling period. For simplicity, I limit the probability of survival and probability of detection to the discrete $[0.9, 1]$ space and fixed the false alarm rate.

Figure 10.2 shows the results of detecting, tracking, and identification of virtual leaders. The method successfully detected spontaneous births, spawned births, and merged modules. It also detected some unexpected short trajectories that appeared due to the random generation of signals. I set a threshold to filter out trajectories shorter than 10 time steps. Parameters of the transition model and of the observation

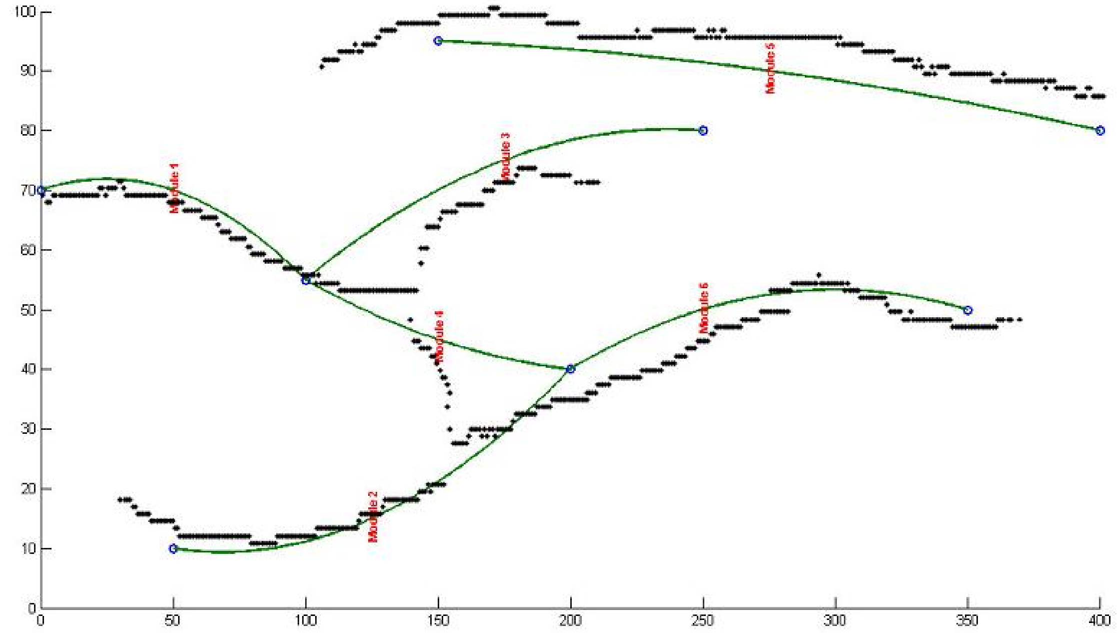


Figure 10.2: Estimated trajectories of the module virtual leader after filtering out the short trajectories. The green lines are true trajectories and black lines are estimated state of trajectories. The horizontal axis is time and the vertical axis is the state space.

model were learned as follows:

$$\begin{aligned}
&\widehat{F} = \begin{bmatrix} 1.022 & 0.984 \\ 0.002 & 0.990 \end{bmatrix}, \qquad \widehat{Q} = \begin{bmatrix} 0.080 & 0.108 \\ 0.108 & 0.212 \end{bmatrix}, \qquad \widehat{H} = \begin{bmatrix} 0.988 \\ 0.002 \end{bmatrix}, \\
&\widehat{R} = 5.764, \quad P_S = 0.98, \quad P_D = 0.94
\end{aligned} \tag{10.2}$$

10.2 Evaluation of Different Scenarios

In this section, we evaluated the performance of our method using 100 randomly generated scenarios. The experimental settings are the same as in the previous example (section 10.1), but scenarios are not fixed. The wasserstein metric has been used to capture the state estimation errors and cardinality errors. The wasserstein metric provides a tool for measuring the distance between two nonempty finite sets, as described in [31].

Let $\hat{X}^k$ and X^k be finite sets of the estimated state and true state of the system at time k, respectively. Standard performance evaluation methods such as the mean square distance-error are not applicable to estimate the state of a random set. Figure 10.3 shows the expectation of the wasserstien L_∞ distance [31] for 100 experiments run with setting that were described in section 10.1. Given a weighted complete bipartite graph $G = (X^k \cup \hat{X}^k; X^k \times \hat{X}^k)$, where edge $x_i\widehat{x_j}$ has weight $c(x_i\widehat{x_j})$, Wasserstein distance is the solution for finding a matching $\mathbb{M}$ from X^k to $\hat{X}^k$ with a minimum weight, or in other words, it is the solution for generalizations of optimal assignment problems. The Wasserstein distance is defined as follows [10]:

$$d_\infty^W(X^k, \hat{X}^k) = \inf_C \max_{x_i \in X^k, \hat{x}_j \in \hat{X}^k} \tilde{C}_{ij} d_p(x_i, \hat{x}_j) \tag{10.3}$$

where $d_p(x, y)$ is the order p Euclidian distance (Minkowski distance), $\tilde{C}_{ij} = 1$ if $C_{ij} > 0$ and $\tilde{C}_{ij} = 0$ if $C_{ij} = 0$, and C is an $|X^k| \times |\hat{X}^k|$ matrix $\{C_{ij}\}$ such that

$$\begin{gathered}\forall i = 1 \cdots |X^k|, \forall j = 1 \cdots |\hat{X}^k| : \\ \sum_{i=1}^{|X^k|} C_{ij} = \frac{1}{|\hat{X}^k|}, \sum_{i=1}^{|\hat{X}^k|} C_{ij} = \frac{1}{|X^k|}\end{gathered}$$

Figure 10.4 shows the $E[||\widehat{X}^k| - |X^k||]$.

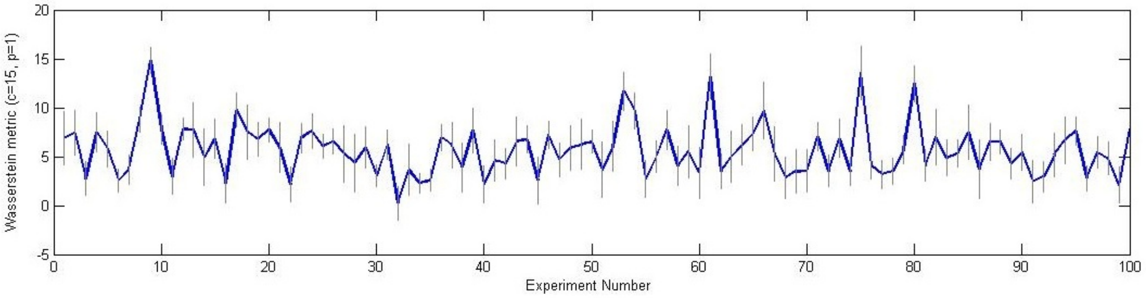


Figure 10.3: Averaged ($\pm\sigma$) Wasserstein metric (c=15, p=1) for each experiment in the simulation study.

10.3 Application to a Real Dataset

In this section I demonstrate the performance of BASED in application to a real dataset. We used publicly available microarray time series data that model transforming growth factor beta-induced epithelial-mesenchymal transition (EMT) in a human lung cancer cell line [58]. During EMT, epithelial cells acquire migratory phenotype typical of cancer cells due to *de novo* expression of mesenchymal-specific proteins. TGF-beta has been shown to play a key role in this transition by triggering certain

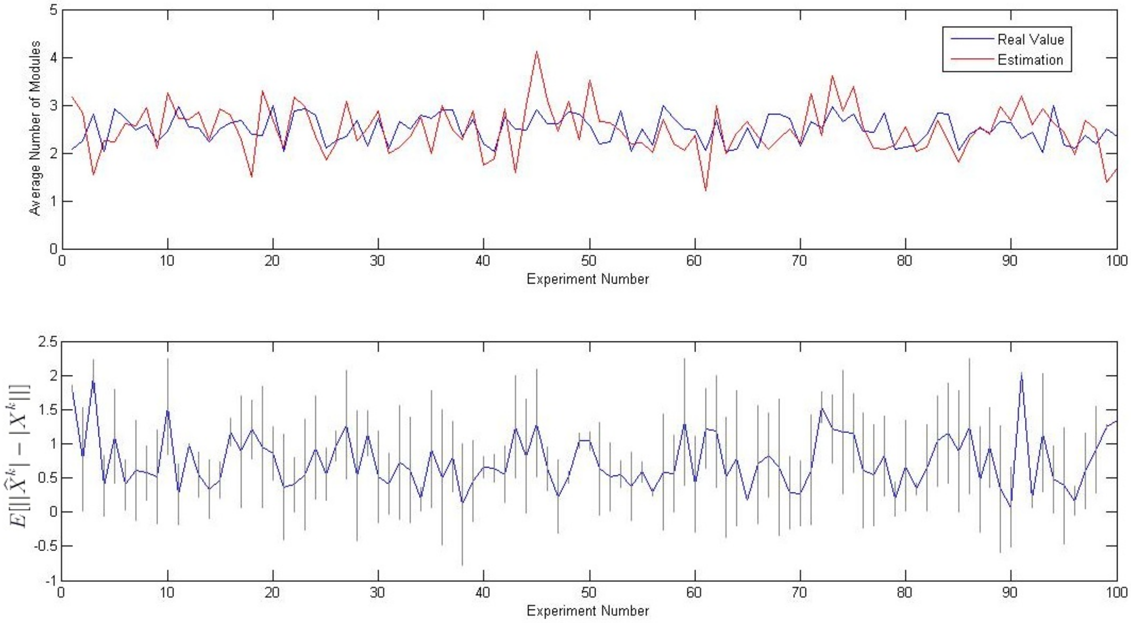


Figure 10.4: Average number of modules and its estimation in each simulated experiment (top). Absolute Errors of the estimates of module numbers in each experiment, and their variation ($\pm\sigma$) (bottom)

signaling pathways leading to down-regulation of epithelia-specific proteins [34, 35]. Sartor et al. [58] have explored in details the timing of cell-type transition, and therefore their dataset represents a very suitable model system for testing our method. Gene expression data were downloaded from the NCBI's Gene Expression Omnibus (accession number GSE17708). The data were already preprocessed as described in [35] and thus could be directly fed into BASED. However, to reduce the amount of noise we filtered genes with the lowest variance, resulting in 6320 genes entering the analysis. The thresholds used for reconstruction of the evolution diagram were the same as in simulation experiments described in section 10.1. A total of 9 time points were available, including control (no TGF-beta added to the cell culture) and 0.5, 1, 2, 4, 8, 16, 24, and 72 hour post treatment. As it can be seen from the reconstructed evolution diagram (Figure 10.5), the experimental system was in a steady state until 2 h after treatment with TGF-beta. However, a considerable rewiring in the network representing gene associations occurred from 2 to about 16 hour post treatment, leading the system to a new steady state. This dynamics reconstructed with the help of BASED corresponds to the cell's epithelial-mesenchymal transition and is in full agreement with timing discussed in the original work by [58] (e.g., see Figure 3 in their paper). This result demonstrates the ability of our method to detect critical time-resolved events in complex systems described with thousands of variables.

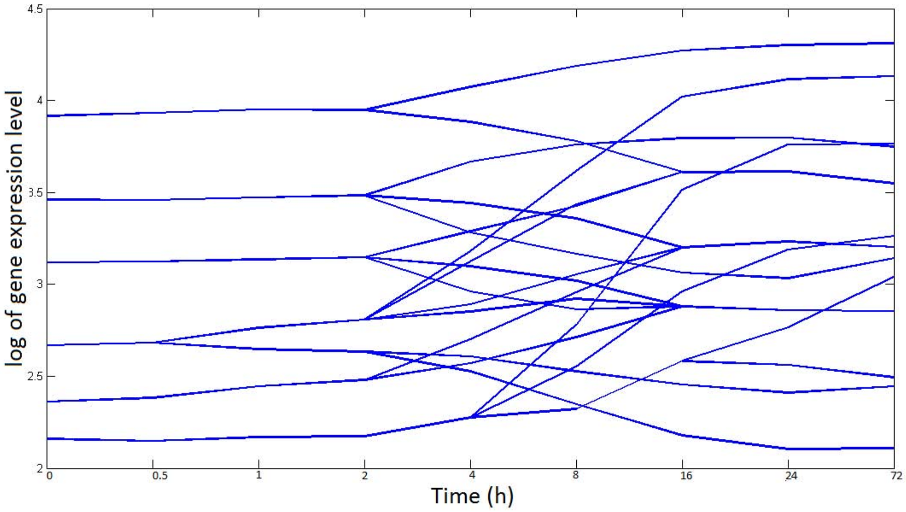


Figure 10.5: Stochastic evolution diagram reconstructed from the data of [58].

10.4 From Time Series to Gannt Chart Workflow

In particular, biological processes often represent a complex sequence of parallel events and sub-processes. Localizing these events in time and constructing the corresponding temporal maps can generate many interesting hypotheses for further testing, such as transcriptional regulatory events. I have shown how to reconstruct the underlying Gantt chart of epithelial-mesenchymal transition by using the stochastic evolution diagram.

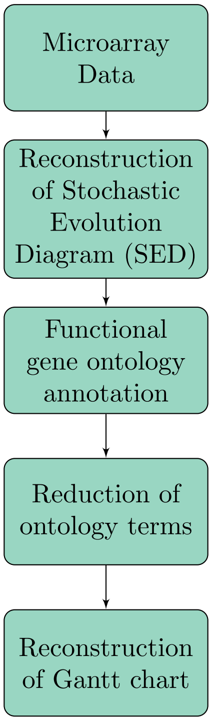

Microarray Data
Reconstruction of Stochastic Evolution Diagram (SED)
Functional gene ontology annotation
Reduction of ontology terms
Reconstruction of Gantt chart

Figure 10.6: Workflow of the experiment

The Figure 10.6 shows the designed workflow that I followed. First I applied our method on time series gene expression profiles obtained from TGF-beta-induced epithelial-mesenchymal transition (EMT) in human lung cancer cells with 9 time samples over course of 72 hours [58], then filtered out genes with low-variance expression levels. In the second step,I reconstructed the evolution diagram and removed the small and short lifetime modules (less than three consecutive time step), then divided it to segments in such a way that each segment is a collection of all genes that belong to a path between two consecutive critical points (i.e. birth, death, split, and merge). In Figure 10.7, ellipses show four segments of SED (stochastic evolution diagram)for example.

In the third and fourth, I have annotated each segments with functional GO terms by DAVID [32], reduced redundant GO terms by REViGO [67], and then I picked top significant GO terms with p-value less than 0.05.

Finally, as I have mentioned before, biological processes can be described as ordered and synchronized events. A Gannt chart can be used to illustrate the start and finish time of events and shows assigned elements of each event. Figure 10.8 shows the Gannt chart that summarizes the generated hypothesis about lifetime of each function. Numbers on the green bars are segments numbers, i.e. the corresponding function is linked to genes that belong to the respective segments.

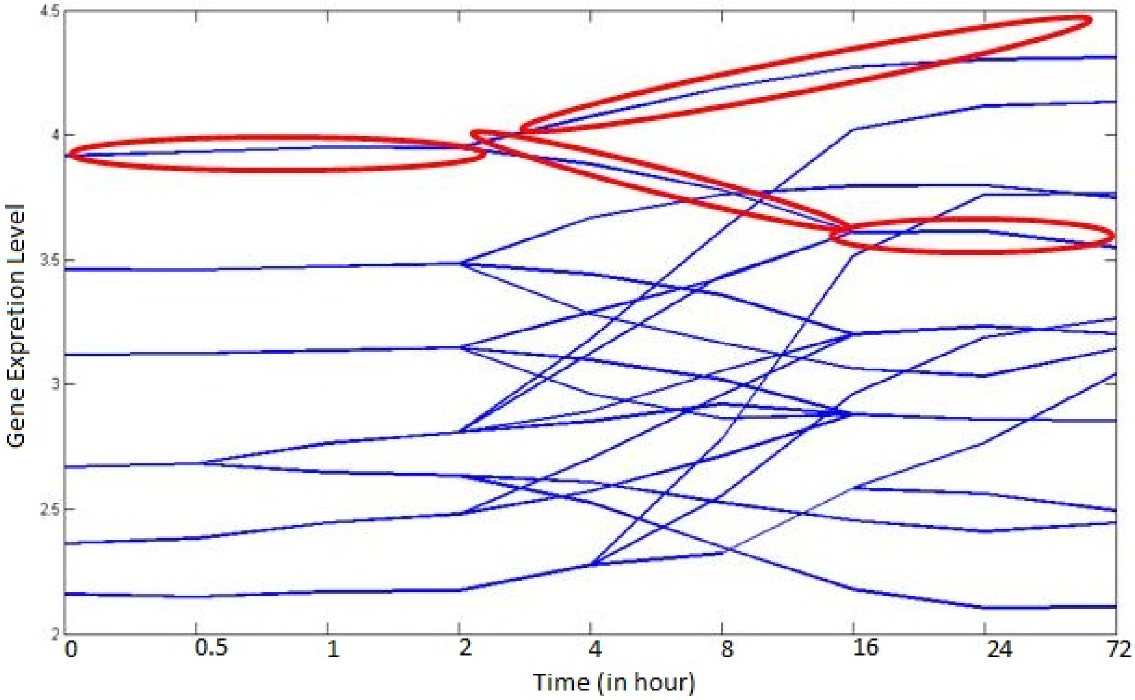


Figure 10.7: Obtained evolution diagram and four segments of it.

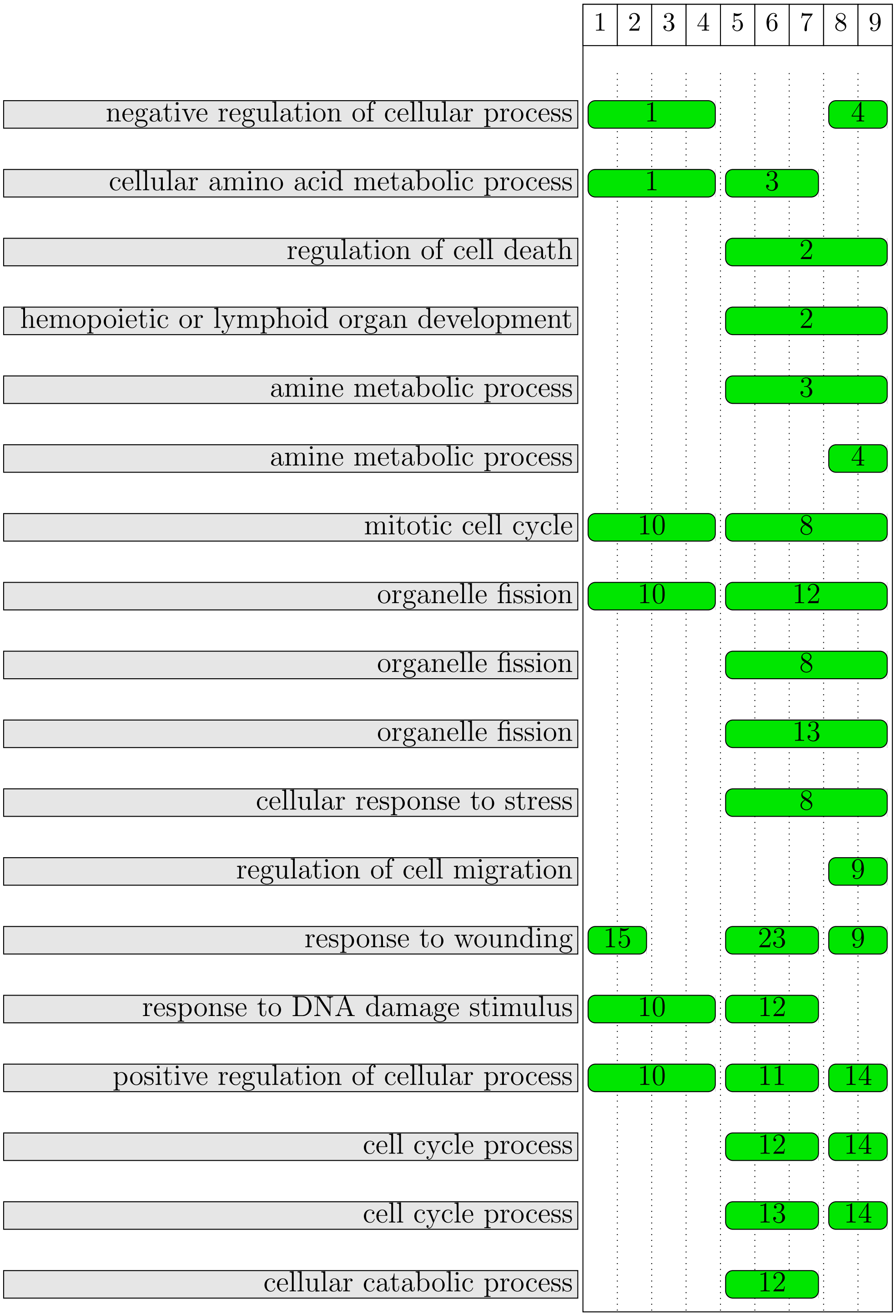


Figure 10.8: Reconstructed Gannt Chart of gene expression data (part one). A number is assigned to each group of genes (numbers written on green segments).

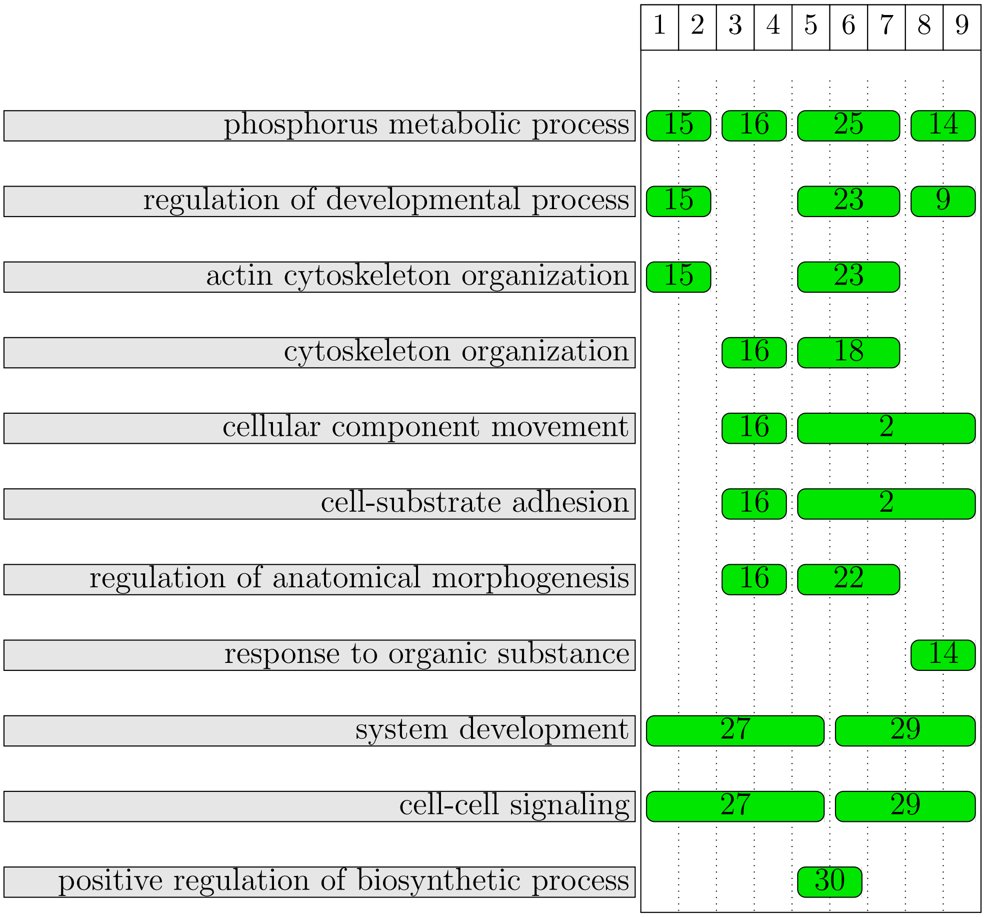


Figure 10.9: Reconstructed Gannt Chart of gene expression data (part two). A number is assigned to each group of genes (numbers written on green segments).

CHAPTER 11

DISCUSSION

As the name suggests, *stationary systems* are always governed by a constant set of rules. In contrast, systems undergoing *evolutionary dynamics* are controlled by non-constant sets of rules over the duration of development. This second type of complex systems was the focus of the present thesis. In this chapter, I provide a brief overview over the methods that have been used to infer the evolutionary dynamics of non-stationary systems. Figure 11.1 has summarized the three main domains that have modeled complex systems and their relations to our model, BASED.

11.1 Clustering

Clustering is a fundamental approach in machine learning and statistical data analysis. Traditional clustering algorithms have recently been extended to ”evolutionary clustering”, which considers a complex system as composed of evolving collections of objects. The goal of the evolutionary clustering is to detect, identify, classify, and track these collections over time [9]. The following two groups of approaches used for evolutionary clustering seem to be particularly important.

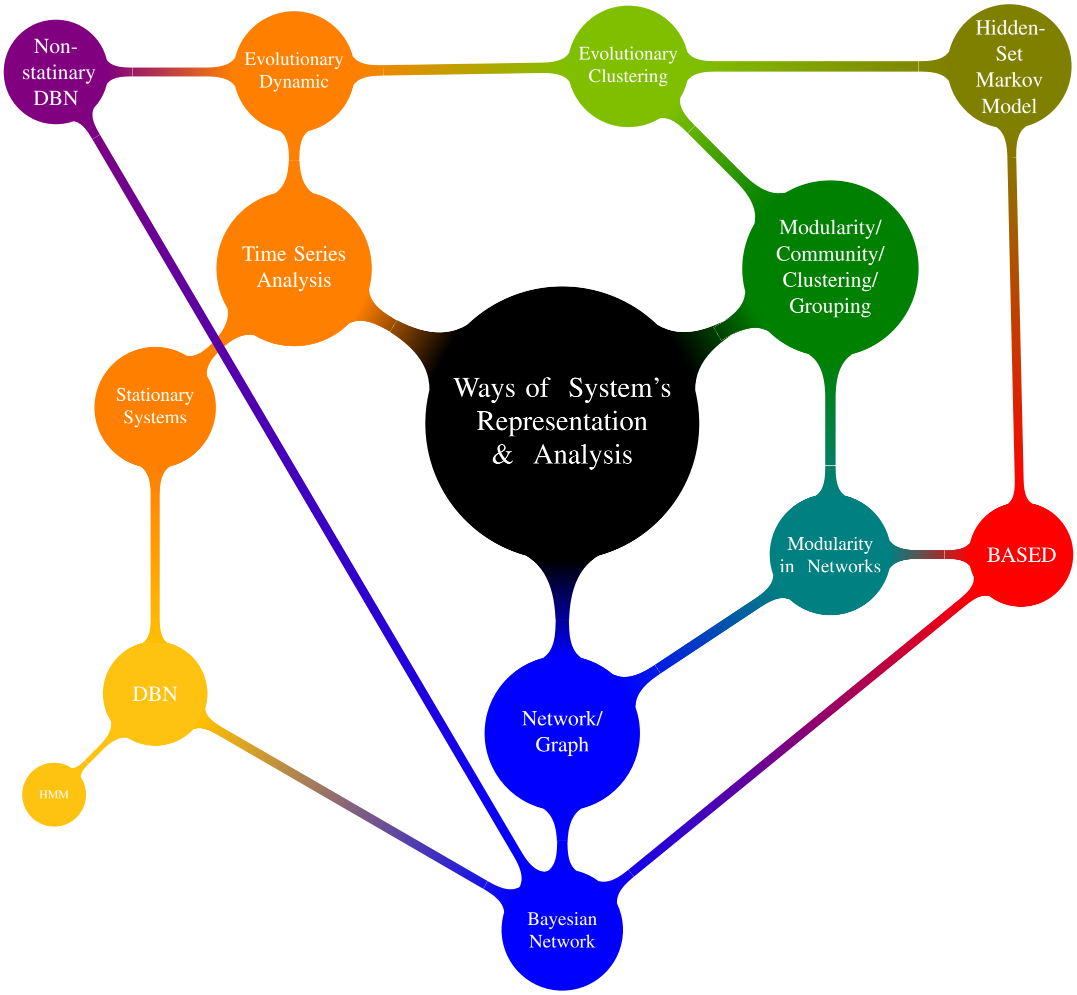


Figure 11.1: Three main domains of models that describe complex systems, and the relation of our method, BASED, to these domains.

11.1.1 Smooth Function Approaches

Several algorithms have introduced temporal smooth functions reflecting the trajectories of clusters (but not individual members of the clusters). In the method of Chakrabarti et al. [9], the smoothness is constrained in the sense that the numbers of clusters are not allowed to evolve dramatically. The cost function is optimized with regard to the the *snapshot quality* and *history quality*, which are the measures of how well the data are clustered at a specific time point and how smooth the current clustering configuration is as compared to the configurations at adjacent time points, respectively. Conceptually similar extensions have been made to the the k-means algorithm [9, 80, 84], agglomerative hierarchical clustering [9, 80], Gaussian mixture and multinomial mixture models [84], and spectral clustering of time series [80].

11.1.2 Approaches Based on the Dirichlet Process

The Dirichlet process mixture model (DPM) is able to learn the number of components of a mixture distribution from the observations. It has been extended to address the evolutionary clustering problem in several ways. The latent Dirichlet allocation (LDA) has been proposed by [6] to model a known number of topics in a body of text, and later to capture the evolution of topics in a sequentially organized corpus of documents [5]. Similar LDA-based methods can be found in studies by [25] and [78].

Teh et al. [70] have proposed a hierarchical nonparametric Bayesian solution to model a sequence of mixture models such that the mixtures' components are allowed to be shared in the sequence.

Many authors attempted to extend the idea of hidden Markov model (HMM) for infinite number of states. Beal et al. [4] have defined a non-parametric Bayesian HMM by a two-level hierarchical Dirichlet process and have introduced the infinite hidden Markov model (iHMM), also called HDP-HMM. Xu et al. [81], Teh et al. [70], Fox et al. [19] and Ni et al. [52] have modified and developed the iHMM to have more effective and efficient learning from the data.

11.2 Time Series Analysis

Time series generated from a natural system are typically non-linear and non-stationary, and the idea of probabilistic time series analysis is the presentation of hidden switching Markov model for time series, which is a generalization of both State Space Model (SSM) or stochastic linear dynamics systems [21]. These approaches segment the data into some regimes. A wide range of statistical methods is applicable for non-linear non-stationary time series produced by complex systems. In a series of his publications on time-series segmentation (a.k.a. change-point detection), Fearnhead [18] has demonstrated that, in comparison to an earlier approach of Punskaya et al. [54], his dynamic programming algorithm based on Bayesian inference improves the accuracy of computation of the posterior distribution for the number and location of change-points in time-series. Xuan and Murphy [82] have further extended Fearnhead's algorithm [18] for decomposable multivariate time-series (change-points occur for all components of the multivariate vector at the same time) by estimating the graph structure for each segment. Robinson and Hartemink [56] have generalized the problem and considered the change-point effects for subsets of variables, and have used rjMCMC to estimate a joint posterior probability over all networks.

11.3 Networks and Graph-Based Methods

Representation of the system under study by a network which shows the associations between variables (nodes) has attained particularly attention in studies of high-dimensional complex systems. There are two main categories of network analysis. In the first category the topology of the network is known and the goal is to study the network by applying network-related statistical methods. In the second category, the constructed underlying network can be used as a prior knowledge to analyze the system.

Since 2000, a number of network models have been used to reveal either a sequence of dependent networks [40, 56, 82] or a common network that would reflect the structure of shared information through time [26]. The paper by Robinson and Hartemink [56] provides a summary of the related works on reconstructing a sequence of networks.

11.4 Conclusion and future work

The behavior of complex systems can be fitted with a number of well developed models. However, these models tend to either consider only the modularity of a system, ignoring the evolution of the modules, or describe the dynamics of the system without taking its modularity into account. As a result, the investigator cannot fully understand the structure and dynamics of the system. To address this issue, I used the framework of Random Set Theory and developed a model that allows to describe the dynamics of complex systems more realistically by reconstructing their stochastic evolution diagrams. This generalized model can be applied in various research fields that deal with complex systems. In particular, we tested our model on simulated data as well as real dataset that contains time series of gene expression levels.

In this work, we attempted to examine an evolving complex system as a dynamic system of dynamic systems, and the general idea was to consider all random variables of a complex system as elementary particles with unknown lifetime. The dynamic behavior of the systems was modeled by considering both their modularity and evolutionary characteristics. In the Random Set Theory's framework I developed a corresponding model and defined a few new concepts such as the hidden set Markov model and the multiple hidden sets Markov model, and used the concept of random cluster processes in the finite set statistics (FISST). I introduced the concept of the stochastic evolution diagram, and also derived a BD metric to score a labeled-edge Bayesian network (a sequence of Bayesian networks). Using this novel model, an investigator can reconstruct the stochastic evolution diagram and find the highest probable labeled-edge Bayesian network to understand the structure and dynamics of complex systems. The stochastic evolution diagram and the labeled-edge Bayesian network models can be applied in various research fields that deal with time series with a large number of random variables.

APPENDIX A

PROOF OF BD METRIC FROM CHAPTER 9

A.1 Problem Definition

It is possible to present a sequence of Bayesian networks by a labeled-edge Bayesian network (Figure 9.1). Let the triple $\mathbf{G} = (G_T, G_P, \Psi)$ parameterize a labeled-edge Bayesian network, where G_T is a DAG denoting the Bayesian network topology, and G_P is a vector whose values denote the conditional probability assignments associated with Bayesian network topology G_T [14]. Ψ is all information encapsulated in the stochastic evolution diagram. In this section, similar to works of Heckerman et al. [28] and Cooper and Herskovits [14], we present a metric for evaluating the probabilities of different DAGs given the *discrete data* and the stochastic evolution diagram.

Suppose the variables $X = \{x_1^1, \ldots, x_n^T\}$ be a set of n discrete observed features during T time samples, where a feature x_i can have s_i possible states or values: $\{v_{i,1}, \ldots, v_{i,s_i}\}$. Let w_i denote a list of the unique instantiations for parents of x_i as seen in $\mathcal{D}$, and w_{ij} denote the jth unique instantiation of π_i relative to $\mathcal{D}$, and there are q_i such unique instantiations of π_i (we have followed the notation of Cooper and Herskovits [14]). Let π_i^l to be a set of elements of π_i which also belong to the cluster process l, then $\pi_i^l \subset \pi_i$, and let w_{ij}^l denotes instantiation of π_i^l. N_{ijs}^l is defined as a number which is proportionate to cases in the time series in which the feature $x_i \in C_l$ holds on the value $v_{i,s}$ and the π_i^l is instantiated as w_{ij}^l. N_{ijs}^l is formulated below:

$$N_{ijs}^l = \frac{1}{|\pi_i|} \sum_{k=2}^{T} \Delta^k \tag{A.1}$$

where

$$\Delta^k = \begin{cases} |w_{ij}^l| & \pi_i^l = w_{ij}^l \\ 0 & otherwise. \end{cases} \tag{A.2}$$

and $|.|$ means the cardinality of the set.

If we are able to obtain the $p(G_T, \mathcal{D}|\Psi)$, we can rank the probabilities of different topologies. We have proven a BD metric of likelihood function $p(D|G_T, G_P)$ by marginalizing $p(G_T, G_P, \mathcal{D}|\Psi)$ as follows:

$$p(G_T, \mathcal{D}|\Psi) = p(G_T|\Psi) \int_{G_P} p(\mathcal{D}|G_T, G_P) f(G_P|G_T, \Psi) dG_P \tag{A.3}$$

We have obtained the closed form expression below for the BD metric:

$$\begin{aligned} p(G_T, \mathcal{D}|\Psi) = p(G_T|\Psi) \prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij}^l + \alpha_{ij})} \\ \prod_{s=1}^{s_i} \frac{\Gamma(N_{ijs}^l + \alpha_{ijs})}{\Gamma(\alpha_{ijs})} \end{aligned} \tag{A.4}$$

where $N_{ij}^l = \sum_{s=1}^{s_i} N_{ijs}^l$, α_{ijs}s are Dirichlet hyper-parameters of prior probability distribution of the DAG topology, $\alpha_{ij} = \sum_{s=1}^{s_i} \alpha_{ijs}$, and $\Gamma(.)$ is the gamma function.

A.2 Proof

To derive the equation A.4, I followed Cooper and Herskovits's [14] procedure:

$$p(G_T, D|\Psi) = \int_{G_P} p(D|G_T, G_P) f(G_P|G_T, \Psi) p(G_T|\Psi) dG_P \tag{A.5}$$

Here $f(.)$ is the conditional probability density function over G_P given G_T.

$p(G_T|\Psi)$ is a constant within equation A.5, we can move it outside of the integral.

$$p(G_T, D|\Psi) = p(G_T|\Psi) \int_{G_P} p(D|G_T, G_P) f(G_P|G_T, \Psi) dG_P \tag{A.6}$$

By assuming the undergoing process is random cluster process

$$p(G_T, D|\Psi) = p(G_T|\Psi) \int_{G_P} \left[\prod_{l=1}^{N_c} \prod_{k=2}^{\mathbb{K}} p(R_k|G_T, N_p, \Psi) \right] f(G_P|G_T) dG_P \tag{A.7}$$

where $R_k = mo^l_k \cup mo^l_{k-1}$, we can write $p(R_k|G_T, N_p, \Psi)$ in its Bayesian network form as:

$$\begin{aligned} p(G_T, D|\Psi) =& p(G_T|\Psi) \int_{G_P} \left[\prod_{l=1}^{N_c} \prod_{k=2}^{\mathbb{K}} \prod_{x_{i,k} \in mo^l_k} p(x_{i,k} = d_{i,k}|\pi^l_i = w^l_{i,\sigma_2(i,k-1)}, G_P, \Psi) \right] \times \\ & f(G_P|G_T) dG_P \end{aligned} \tag{A.8}$$

$d_{i,k}$ denotes the value assigned to the $x_{i,k}$ in D. $\sigma_2(i, k-1)$ is an index function such that the initiation of π^l_i at time $k-1$ is the $\sigma_2(i, k-1)$th element of w^l_i.

By grouping the terms, we can rewrite the inner product in equation A.8 as:

$$\begin{aligned} p(G_T, D|\Psi) =& p(G_T|\Psi) \int_{G_P} \left[\prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} \prod_{s=1}^{s_i} p(x_i = v_{i,s}|\pi^l_i = w^l_{ij}, G_P, \Psi)^{N^l_{ijs}} \right] \times \\ & f(G_P|G_T) dG_P \end{aligned} \tag{A.9}$$

N^l_{ijs} in equation (A.9) can be interpreted as the *prior observation count* for events governed by $p(x_i = v_{i,s}|\pi^l_i = w^l_{ij}, G_P, \Psi)$. An important point is the random variable $x_{i,k}$ in equation(A.8) which is changed to the feature x_i in equation(A.9) because of the stationarity assumption during the life time of each random cluster processes.

Let $\theta_{ijs} = p(x_i = v_{i,s}|\pi^l_i = w^l_{ij}, G_P, \Psi)$, then the $(\theta_{ij1}, \ldots, \theta_{ijs_i})$ is a list of probabilities. Let $f(\theta_{ij1}, \ldots, \theta_{ijs_i})$ denote the probability density function over $\{\theta_{ij1}, \ldots, \theta_{ijs_i}\}$.

The function $f(.)$ is called a second-order probability distribution because it is a probability distribution over a probability distribution [14].

$f(\theta_{ij1}, \dots, \theta_{ijs_i})$ is independent of the distribution $f(\theta_{i'j'1}, \dots, \theta_{i'j's_i})$ for $1 \leqslant i, i' \leqslant n, 1 \leqslant j \leqslant q_i, 1 \leqslant j' \leqslant q_{i'}$, and $ij \neq i'j'$, because we are indifferent regarding which numerical probabilities to assign to the Bayesian network with topology G_T. Then

$$f(G_P|G_T) = \prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} f(\theta_{ij1}, \dots, \theta_{ijs_i}) \tag{A.10}$$

By substituting θ_{ijs} for $p(x_i = v_{i,s}|\pi_i^l = w_{ij}^l, G_P, \Psi)$ in equation A.9, and substituting equation A.10 into equation A.9, we obtain:

$$p(G_T, D|\Psi) = p(G_T|\Psi) \int \underbrace{\dots}_{\theta_{ijs}} \int \left[\prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} \prod_{s=1}^{s_i} \theta_{ijs}^{N_{ijs}^l} \right] \left[\prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} f(\theta_{ij1}, \dots, \theta_{ijs_i}) \right] d\theta_{111}, \dots, d\theta_{n,q_n,s_n} \tag{A.11}$$

$$p(G_T, D|\Psi) = p(G_T|\Psi) \prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} \int \underbrace{\dots}_{\theta_{ijs}} \int \left[\prod_{s=1}^{s_i} \theta_{ijs}^{N_{ijs}^l} \right] f(\theta_{ij1}, \dots, \theta_{ijs_i}) d\theta_{ij1}, \dots, d\theta_{ijs_i} \tag{A.12}$$

We assume $f(\theta_{ij1}, \dots, \theta_{ijs_i})$ has a Dirichlet distribution:

$$f(\theta_{ij1}, \dots, \theta_{ijs_i}) = \frac{\Gamma(\alpha_{ij})}{\prod_{s=1}^{s_i} \Gamma(\alpha_{ijs})} \theta_{ij1}^{\alpha_{ij1}-1} \times \dots \times \theta_{ijs_i}^{\alpha_{ijs_i}-1} \tag{A.13}$$

where $\alpha_{ij} = \sum_s^{s_i} \alpha_{ijs}$. By substituting equation (A.13) in equation (A.12)

$$p(G_T, D|\Psi) = p(G_T|\Psi) \prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\prod_{s=1}^{s_i} \Gamma(\alpha_{ijs})} \int \dots \int \left[\prod_{s=1}^{s_i} \theta_{ijs}^{N_{ijs}^l + \alpha_{ijs} - 1} \right] d\theta_{ij1}, \dots, d\theta_{ijs_i} \tag{A.14}$$

The multiple integral in equation (A.14) is a Dirichlet integral, and has the following solution:

$$p(G_T, D|\Psi) = p(G_T|\Psi) \prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\prod_{s=1}^{s_i} \Gamma(\alpha_{ijs})} \times \frac{\prod_{s=1}^{s_i} \Gamma(N_{ijs}^l + \alpha_{ijs})}{\Gamma(N_{ij}^l + \alpha_{ij})} \tag{A.15}$$

By re-formulating the equation (A.15):

$$p(G_T, D|\Psi) = p(G_T|\Psi) \prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij}^l + \alpha_{ij})} \prod_{s=1}^{s_i} \frac{\Gamma(N_{ijs}^l + \alpha_{ijs})}{\Gamma(\alpha_{ijs})} \tag{A.16}$$

Or:

$$p(D|G_T, \Psi) = \prod_{l=1}^{N_c} \prod_{x_i \in C_l} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij}^l + \alpha_{ij})} \prod_{s=1}^{s_i} \frac{\Gamma(N_{ijs}^l + \alpha_{ijs})}{\Gamma(\alpha_{ijs})} \tag{A.17}$$

□

APPENDIX B

SYSMBOLS

$Notation$	$Description$
k	time index (always superscript)
i, j	a feature or a module index (always subscript)
T	time sample length
$\mathbb{T}$	time space
x_i	ith feature
$\hat{x}_i$	ith feature state estimation
x_i^k	ith feature at time k
X	a set of random variables
2^X	power set of set X
$\emptyset$	empty set
mo	a random finite set that represents a module
mo_i^k	ith module at time k
$\mathcal{N}et$	a random finite set that represents a network
net	a RFS state variable belongs to network state space
$\mathcal{N}et^k$	state of the network at time k
$p(.)$	probability distribution
$p^{k\|k-1}(.)$	predicted probability distribution form time $k-1$ to k
$p^{k\|k}(.)$	updated probability distribution at time k
$p_n(.)$	a node probability distribution
p_{mo}	a module probability distribution
Z	observation set
Z^k	observation at time k
$Z^{1:k}$	observations from time 1 to k
$\bar{Z}$	a modules' observation set
$\dot{z}$	a nodes' observation
z	a virtual leaders' observation set
$\mathcal{Z}$	virtual leaders observation
$\mathcal{P}(Z)$	the set of all partitions of set Z

Notation	*Description*
N_m^k	number of nodes that belong to module m at time k
$\hat{N}_m^k$	estimated number of nodes that belong to module m at time k
M^k	number of modules in the network at time k
$\hat{M}^k$	estimated number of modules in the network at time k
$f^{k\|k-1}(.)$	Markov transition density function
$f_{mo}^{k\|k-1}(.)$	a modules' Markov transition density function
$L^k(.\|.)$	network likelihood function
$L_{mo}^k(.\|.)$	modules' likelihood function
$p_S^k()$	a modules' probability of survive from time $k-1$ to k
$\dot{p}_S^k()$	a nodes' probability of survive from time $k-1$ to k
$p_D^k()$	a modules' probability of detection
$\dot{p}_D^k()$	a nodes' probability of detection
$\mathbb{R}$	real number
$\mathbb{R}^d$	d dimensional Euclidean space
$p_C()$	a cardinality distribution
$\pi(x_i)$	set of x_i's parent in a Bayesian network
$h()$	a test function
$\mathbb{E}[]$	expectation operator
c_{mo}	module cardinality expected value
${\sigma_{mo}}^2$	module cardinality variance
$G_{mo}[h]$	p.g.fl of a modules' probability distribution function
$G_{mo}^{'}()$	first derivative of $G_{mo}[]$
$G_{mo}^{''}()$	second derivative of $G_{mo}[]$
$G_{X,Y}[g,h]$	a joint p.g.fl of two RFSs
$\beta^{k\|k-1}$	spawned birth model of a module
Γ	spontaneous birth model of a module
$\bar{\Gamma}$	spontaneous birth model of a node
$S^{k\|k-1}$	survival model of a module
$\bar{S}^{k\|k-1}$	survival model of a node
Θ^k	detection model of a module
$\bar{\Theta}^k$	detection model of a node
K^k	false alarm model

Notation	*Description*
$\upsilon^{k\|k-1}()$	a network probability intensity function (PHD) prediction
$\dot{\upsilon}^{k\|k}()$	a module probability intensity function (PHD) update
$\dot{\upsilon}^{k\|k-1}()$	a module probability intensity function (PHD) prediction
$\gamma^k()$	intensity of a new module spontaneous birth
$\dot{\gamma}^k(x)$	intensity of a node spontaneous birth
$\mathcal{S}$	SED state
$\mathbb{S}$	HMM state space
$\mathbb{O}$	HMM observation space
$\mathbb{H}$	all possible state sequence space
$\mathbbm{H}$	all possible SED state space
$\mathcal{L}$	likelihood function
$\mathcal{L}_q$	likelihood of Markov chain q
$\mathcal{L}^k$	likelihood at time k
$\dot{\mathcal{L}}^k$	a node's log-likelihood at time k
λ_p	Poisson parameter
Ψ	stochastic evolution diagram

BIBLIOGRAPHY

[1] A. Ahmed and E. P. Xing. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences of the United States of America*, 106(29):11878–11883, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19570995. <11>

[2] M. M. Babu, N. M. Luscombe, L. Aravind, M. Gerstein, and S. A. Teichmann. Structure and evolution of transcriptional regulatory networks. *Curr Opin Struct Biol*, 14(3):283–291, 2004. ISSN 0959440X. doi: 10.1111/j.1365-2125.2011.03933.x. URL http://www.ncbi.nlm.nih.gov/pubmed/15193307. <1>

[3] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*, volume 16. YBS Publishing, 1995. ISBN 0964831201. doi: 10.1109/MCS.1996.482170. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=482170. <32, 39>

[4] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The Infinite Hidden Markov Model. *Advances in Neural Information Processing Systems*, 1:577–584, 2002. ISSN 10495258. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.16.2929&amp;rep=rep1&amp;type=pdf. <2, 92>

[5] D. M. Blei and J. D. Lafferty. Dynamic topic models. *Proceedings of the 23rd international conference on Machine learning ICML 06*, (1):113–120, 2006. doi: 10.1145/1143844.1143859. URL http://portal.acm.org/citation.cfm?doid=1143844.1143859. <91>

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003. ISSN 15324435. doi: 10.1162/jmlr.2003.3.4-5.993. URL http://www.crossref.org/jmlr_DOI.html. <2, 91>

[7] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, 2006. ISSN 03701573. doi: 10.1016/j.physrep.2005.10.009. URL http://linkinghub.elsevier.com/retrieve/pii/S037015730500462X. <1>

[8] S. Borgatti. Models of core/periphery structures. *Social Networks*, 21(4):375–395, 2000. ISSN 03788733. doi: 10.1016/S0378-8733(99)00019-2. URL http://linkinghub.elsevier.com/retrieve/pii/S0378873399000192. <1>

[9] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining KDD 06*, page 554, 2006. doi: 10.1145/1150402.1150467. URL http://portal.acm.org/citation.cfm?doid=1150402.1150467. <2, 89, 91>

[10] D. E. Clark. *Multiple Target Tracking with The Probability Hypothesis Density Filter*. PhD thesis, 2006. <80>

[11] D. E. Clark and J. Bell. Convergence results for the particle PHD filter. *IEEE Transactions on Signal Processing*, 54(7):2652–2661, 2006. ISSN 1053587X. doi: 10.1109/TSP.2006.874845. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1643904. <47>

[12] D. E. Clark and S. J. Godsill. Group Target Tracking with the Gaussian Mixture Probability Hypothesis Density Filter. *2007 3rd International Conference on Intelligent Sensors Sensor Networks and Information*, pages 149–154, 2007. doi: 10.1109/ISSNIP.2007.4496835. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4496835. <36>

[13] D. E. Clark and B.-N. Vo. Convergence Analysis of the Gaussian Mixture PHD Filter. *IEEE Transactions on Signal Processing*, 55(4):1204–1212, 2007. <47>

[14] G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992. ISSN 08856125. doi: 10.1007/BF00994110. URL http://www.springerlink.com/index/10.1007/BF00994110. <73, 74, 95, 96, 98>

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via Em Algorithm. *Journal of the Royal Statistical Society Series BMethodological*, 39(1):1–38, 1977. ISSN 00359246. URL http://apps.isiknowledge.com/full_record.do?product=WOS&search_mode=GeneralSearch&qid=14&SID=P2779A543d3mM44CgpM&page=1&doc=1. <58, 60>

[16] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks. *Advances in Physics*, 51(4):1079–1187, 2002. ISSN 00018732. doi: 10.1080/00018730110112519. URL http://www.informaworld.com/openurl?genre=article&doi=10.1080/00018730110112519&magic=crossref. <1>

[17] J. Ernst, O. Vainas, C. T. Harbison, I. Simon, and Z. Bar-Joseph. Reconstructing dynamic regulatory maps. *Molecular Systems Biology*, 3(74):74, 2007. URL http://www.ncbi.nlm.nih.gov/pubmed/17224918. <11>

[18] P. Fearnhead. Exact Bayesian curve fitting and signal segmentation. *IEEE Transactions on Signal Processing*, 53(6):2160–2166, 2005. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1433145. <2, 92>

[19] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Developing a tempered HDP-HMM for Systems with State Persistence. *Technology*, (November): 1–44, 2007. <2, 92>

[20] Z. Ghahramani and G. E. Hinton. Parameter Estimation for Linear Dynamical Systems. *University of Toronto technical report CRGTR962*, 6(CRG-TR-96-2):1–6, 1996. ISSN 00207179. doi: 10.1080/00207177208932224. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.131.8274&amp;rep=rep1&amp;type=pdf. <58, 65>

[21] Z. Ghahramani and G. E. Hinton. Variational Learning for Switching State-Space Models. *Neural Computation*, 12(4):831–864, 2000. ISSN 08997667. doi: 10.1162/089976600300015619. URL http://www.mitpressjournals.org/doi/abs/10.1162/089976600300015619. <2, 92>

[22] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, 2002. URL http://arxiv.org/abs/cond-mat/0112110. <1>

[23] A. Gning, L. Mihaylova, S. Maskell, S. Pang, and S. Godsill. Group object structure and state estimation with evolving networks and Monte Carlo methods. 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5677610&tag=1. <32>

[24] I. R. Goodman, R. Mahler, and H. T. Nguyen. *Mathematics of Data Fusion*. Kluwer Academic Publishers, 1997. ISBN 0792346742. <14>

[25] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl(Suppl 1): 5228–5235, 2004. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=387300&tool=pmcentrez&rendertype=abstract. <91>

[26] M. Grzegorczyk and D. Husmeier. Non-stationary continuous dynamic Bayesian networks. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, number

Mcmc, pages 1–9. 2009. URL http://books.nips.cc/papers/files/nips22/NIPS2009_0646.pdf. <2, 93>

[27] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402(6761):C47–52, 1999. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:From+molecular+to+molecular+cell+biology#0. <1, 10, 11>

[28] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3): 197–243, 1995. ISSN 08856125. doi: 10.1007/BF00994016. URL http://www.springerlink.com/index/10.1007/BF00994016. <74, 95>

[29] S. Hernández. Smoothing Algorithms for the Probability Hypothesis Density Filter. URL http://130.203.133.150/viewdoc/versions;jsessionid=F4DB0049D5C5F061AF33B12260931EB3?doi=10.1.1.225.4967. <47>

[30] A. Hintze and C. Adami. Evolution of complex modular biological networks. *PLoS Computational Biology*, 4(2):e23, 2008. ISSN 15537358. doi: 10.1371/journal.pcbi.0040023. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2233666&tool=pmcentrez&rendertype=abstract. <11>

[31] J. R. Hoffman and R. P. S. Mahler. Multitarget miss distance via optimal assignment. *IEEE Transactions on Systems Man and Cybernetics Part A Systems and Humans*, 34(3):327–336, 2004. ISSN 10834427. doi: 10.1109/TSMCA.2004.824848. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1288344. <80>

[32] D. W. Huang, B. T. Sherman, and R. A. Lempicki. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, 4(1):44–57, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19131956. <85>

[33] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal Of Basic Engineering*, 82(Series D):35–45, 1960. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.6247&amp;rep=rep1&amp;type=pdf. <58>

[34] V. G. Keshamouni, G. Michailidis, C. S. Grasso, S. Anthwal, J. R. Strahler, A. Walker, D. A. Arenberg, R. C. Reddy, S. Akulapalli, V. J. Thannickal, T. J. Standiford, P. C. Andrews, and G. S. Omenn. Differential protein expression

profiling by iTRAQ-2DLC-MS/MS of lung cancer cells undergoing epithelial-mesenchymal transition reveals a migratory/invasive phenotype. *Journal of Proteome Research*, 5(5):1143–1154, 2006. URL http://www.ncbi.nlm.nih.gov/pubmed/16674103. <82>

[35] V. G. Keshamouni, P. Jagtap, G. Michailidis, J. R. Strahler, R. Kuick, A. K. Reka, P. Papoulias, R. Krishnapuram, A. Srirangam, T. J. Standiford, P. C. Andrews, and G. S. Omenn. Temporal quantitative proteomics by iTRAQ 2D-LC-MS/MS and corresponding mRNA expression analysis identify post-transcriptional modulation of actin-cytoskeleton regulators during TGF-beta-Induced epithelial-mesenchymal transition. *Journal of Proteome Research*, 8(1): 35–47, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19118450. <82>

[36] H. Kitano. Systems biology: a brief overview. *Science*, 295(5560):1662–1664, 2002. URL http://www.ncbi.nlm.nih.gov/pubmed/11872829. <1, 2, 10, 11>

[37] M. Kolar, L. Song, A. Ahmed, and E. P. Xing. Estimating time-varying networks. *Arxiv preprint arXiv08125087*, stat.ML(1):1–35, 2008. URL http://arxiv.org/abs/0812.5087. <11>

[38] M. Krawczyk. Communities in Social Networks. In *2009 International Conference on Biometrics and Kansei Engineering*, pages 111–116. IEEE, June 2009. doi: 10.1109/ICBAKE.2009.20. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5223241. <1>

[39] A. Kreimer, E. Borenstein, U. Gophna, and E. Ruppin. The evolution of modularity in bacterial metabolic networks. *Proceedings of the National Academy of Sciences of the United States of America*, 105(19):6976–6981, 2008. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2383979&tool=pmcentrez&rendertype=abstract. <10, 11>

[40] S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, and G. Lelandais. Statistical inference of the time-varying structure of gene regulation networks. *BMC systems biology*, 4(1):130, 2010. ISSN 17520509. doi: 10.1186/1752-0509-4-130. URL http://www.biomedcentral.com/1752-0509/4/130. <2, 11, 93>

[41] R. Mahler. Random set theory for target tracking and identification. *Handbook of Multisensor Data Fusion*, 2001. <18, 31, 32>

[42] R. Mahler. Detecting, tracking, and classifying group targets: a unified approach. In *Proceedings of SPIE*, volume 4380, pages 217–228. Spie, 2001. doi: 10.1117/12.436950. URL http://link.aip.org/link/?PSI/4380/217/1&Agg=doi. <23, 33>

[43] R. Mahler. Multitarget bayes filtering via first-order multitarget moments. *Ieee Transactions On Aerospace And Electronic Systems*, 39(4):1152–1178, 2003. ISSN 00189251. doi: 10.1109/TAES.2003.1261119. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1261119. <32, 33>

[44] R. Mahler. A theory of PHD filters of higher order in target number. In *Proceedings of SPIE*, volume 6235, pages 62350K–62350K–12. SPIE, June 2006. URL http://adsabs.harvard.edu/abs/2006SPIE.6235E..19M. <47>

[45] R. Mahler. *Statistical multisource-multitarget information fusion*. Artech House Publishers, 2007. ISBN 9781596930926. URL http://www.gbv.de/dms/ilmenau/toc/52413586X.PDF. <14, 15, 18, 20, 21, 22, 28, 32>

[46] R. Mahler, B.-N. Vo, and B.-T. Vo. The Forward-Backward Probability Hypothesis Density Smoother. *Proceedings 13th International Conference on Information Fusion*, (Imm), 2010. <47>

[47] M. D. Mesarovic, S. N. Sreenath, and J. D. Keene. Search for organising principles: understanding in systems biology. *Systems Biology*, 1(1):19–27, 2004. URL http://www.ncbi.nlm.nih.gov/pubmed/17052112. <1>

[48] N. Nandakumaran, K. Punithakumar, and T. Kirubarajan. Improved multitarget tracking using probability hypothesis density smoothing. In *Proceedings of SPIE*, volume 6699, pages 66990M–66990M–8. SPIE, Aug. 2007. doi: 10.1117/12.734656. URL http://adsabs.harvard.edu/abs/2007SPIE.6699E..19N. <47>

[49] N. Nandakumaran, R. Tharmarasa, T. Lang, and T. Kirubarajan. Gaussian mixture probability hypothesis density smoothing with multistatic sonar. *Proceedings of SPIE*, 6968(May 2010):696807–696807–8, 2008. ISSN 0277786X. doi: 10.1117/12.779236. URL http://link.aip.org/link/PSISDG/v6968/i1/p696807/s1&Agg=doi. <47>

[50] N. Nandakumaran, T. Kirubarajan, T. Lang, M. Mcdonald, and K. Punithakumar. Multitarget Tracking using Probability Hypothesis Density Smoothing. *IEEE Transactions on Aerospace and Electronic Systems*, 47(4):2344–2360, 2011. ISSN 0018-9251. doi: 10.1109/TAES.2011.6034637. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6034637. <47>

[51] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103(23):8577–8582, 2006. URL http://arxiv.org/abs/physics/0602124. <1, 10, 11>

[52] K. Ni, L. Carin, and D. Dunson. Multi-task learning for sequential data via iHMMs and the nested Dirichlet process. *Proceedings of the 24th International Conference on Machine Learning (2007)*, pages 689–696, 2007.

doi: 10.1145/1273496.1273583. URL http://portal.acm.org/citation.cfm?doid=1273496.1273583. <92>

[53] A. P. Parikh, W. Wu, R. E. Curtis, and E. P. Xing. TREEGL: reverse engineering tree-evolving gene networks underlying developing biological lineages. *Bioinformatics*, 27(13):i196–i204, 2011. URL http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/btr239. <11>

[54] E. Punskaya, C. Andrieu, A. Doucet, and W. J. Fitzgerald. Bayesian curve fitting using MCMC with applications to signal segmentation. *IEEE Transactions on Signal Processing*, 50(3):747–758, 2002. ISSN 1053587X. doi: 10.1109/78.984776. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=984776. <92>

[55] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979. ISSN 00189286. doi: 10.1109/TAC.1979.1102177. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1102177. <32>

[56] J. W. Robinson and A. J. Hartemink. Learning Non-Stationary Dynamic Bayesian Networks. *Journal of Machine Learning Research*, 11:3647–3680, 2010. <2, 11, 92, 93>

[57] S. Roweis and Z. Ghahramani. An EM algorithm for identification of nonlinear dynamical systems. *Kalman filtering and neural networks*, pages 175–220, 2000. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.7053. <58, 59, 60>

[58] M. A. Sartor, V. Mahavisno, V. G. Keshamouni, J. Cavalcoli, Z. Wright, A. Karnovsky, R. Kuick, H. V. Jagadish, B. Mirel, T. Weymouth, B. Athey, and G. S. Omenn. ConceptGen: a gene set enrichment and gene set relation mapping tool. *Bioinformatics (Oxford, England)*, 26 (4):456–63, Feb. 2010. ISSN 1367-4811. doi: 10.1093/bioinformatics/btp683. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2852214&tool=pmcentrez&rendertype=abstract. <x, 4, 76, 81, 82, 83, 85>

[59] G. Schwarz. Estimating Dimension of a Model. *Annals of Statistics*, 6(2):461–464, 1978. ISSN 00905364. <63>

[60] F. Schweitzer, G. Fagiolo, D. Sornette, F. Vega-Redondo, A. Vespignani, and D. R. White. Economic networks: the new challenges. *Science*, 325(5939):422–425, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19628858. <1>

[61] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–176, 2003. URL http://www.ncbi.nlm.nih.gov/pubmed/12740579. <1, 11>

[62] O. Shamir and N. Tishby. Model Selection and Stability in k-means Clustering. *Machine Learning*, 2008. URL http://eprints.pascal-network.org/archive/00004154/. <63>

[63] V. A. Smith, J. Yu, T. V. Smulders, A. J. Hartemink, and E. D. Jarvis. Computational Inference of Neural Information Flow Networks. *PLoS Computational Biology*, 2(11):14, 2006. URL http://www.ncbi.nlm.nih.gov/pubmed/17121460. <11>

[64] L. Song, M. Kolar, and E. P. Xing. KELLER: estimating time-varying interactions between genes. *Bioinformatics*, 25(12):i128–i136, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19477978. <2, 11>

[65] L. Song, M. Kolar, and E. P. Xing. Time-Varying Dynamic Bayesian Networks. *Advances in Neural Information Processing Systems*, pages 1–9, 2009. URL http://books.nips.cc/papers/files/nips22/NIPS2009_0858.pdf. <11>

[66] V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences of the United States of America*, 100(21):12123–12128, 2003. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=218723&tool=pmcentrez&rendertype=abstract. <1>

[67] F. Supek, M. Bosnjak, N. *S*kunca, and T. *S*muc. REVIGO summarizes and visualizes long lists of gene ontology terms. *PloS one*, 6(7):e21800, Jan. 2011. ISSN 1932-6203. URL http://dx.plos.org/10.1371/journal.pone.0021800. <85>

[68] A. Swain and D. Clark. Extended object filtering using spatial independent cluster processes. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5711886. <29>

[69] A. Swain and D. E. Clark. First-moment filters for spatial independent cluster processes. *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2010. <28, 29, 30, 34>

[70] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006. ISSN 01621459. doi: 10.1198/016214506000000302. URL http://pubs.amstat.org/doi/abs/10.1198/016214506000000302. <2, 91, 92>

[71] A. Vázquez, R. Dobrin, D. Sergi, J. P. Eckmann, Z. N. Oltvai, and A. L. Barabasi. The topological relationship between the large-scale attributes and local interaction patterns of complex networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(52):17940–17945, 2004. URL http://arxiv.org/abs/cond-mat/0408431. <10, 11>

[72] A. Vespignani. Evolution thinks modular., 2003. URL http://www.ncbi.nlm.nih.gov/pubmed/14517536. <10, 11>

[73] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, Apr. 1967. ISSN 0018-9448. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1054010. <58>

[74] B. Vo, S. Singh, and A. Baddeley. Technical aspects of the probability hypothesis density recursion, Oct. 2005. URL http://publications.eng.cam.ac.uk/326703/. <33>

[75] B.-N. Vo and W. K. Ma. The Gaussian Mixture Probability Hypothesis Density Filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, 2006. ISSN 1053587X. doi: 10.1109/TSP.2006.881190. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1710358. <36, 39, 40, 41, 42, 43, 44, 62>

[76] B.-N. Vo, S. Singh, and A. Doucet. Sequential monte carlo methods for multitarget filtering with random finite sets. *Ieee Transactions On Aerospace And Electronic Systems*, 41(4):1224–1245, 2005. ISSN 00189251. doi: 10.1109/TAES.2005.1561884. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1561884. <39, 45, 46, 61, 68>

[77] B.-N. Vo, B.-T. Vo, and R. P. S. Mahler. A closed form solution to the Probability Hypothesis Density Smoother, 2010. <47>

[78] X. Wang and A. Mccallum. Topics over Time : A Non-Markov Continuous-Time Model of Topical Trends. *Time*, pages 424–433, 2006. doi: 10.1145/1150402.1150450. URL http://portal.acm.org/citation.cfm?id=1150402.1150450. <91>

[79] Z. Wang, E. E. Kuruo, X. Yang, Y. Xu, and T. S. Huang. Time Varying Dynamic Bayesian Network for Nonstationary Events Modeling and Online Inference. *IEEE Transactions on Signal Processing*, 59(4):1553–1568, 2011. ISSN 1053587X. doi: 10.1109/TSP.2010.2103071. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?isnumber=5725269&arnumber=5678659. <2, 11>

[80] K. S. Xu, M. Kliger, and A. O. Hero. Adaptive Evolutionary Clustering. *eprint arXiv11041990v1*, page 29, 2011. URL http://arxiv.org/abs/1104.1990. <2, 91>

[81] T. Xu, Z. M. Zhang, P. S. Yu, and B. Long. Evolutionary Clustering by Hierarchical Dirichlet Process with Hidden Markov State. *2008 Eighth IEEE International Conference on Data Mining*, pages 658–667, 2008. ISSN 15504786. doi: 10.1109/ICDM.2008.24. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4781161. <92>

[82] X. Xuan and K. Murphy. Modeling changing dependency structure in multivariate time series. *Proceedings of the 24th International Conference on Machine Learning (2007)*, 227(m):1055–1062, 2007. doi: 10.1145/1273496.1273629. URL http://portal.acm.org/citation.cfm?doid=1273496.1273629. <2, 92, 93>

[83] K. Yu. Generating Gaussian Mixture Models by Model Selection For Speech Recognition. <63>

[84] J. Zhang, Y. Song, G. Chen, and C. Zhang. On-line Evolutionary Exponential Family Mixture. *21st International Joint Conference on Artificial Intelligence IJCAI09*, pages 1610–1615, 2009. <2, 91>