CLUSTERING, COVERAGE AND AGGREGATION METHODS FOR LARGE NETWORKS

BY

YUNWEN XU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Associate Professor Carolyn Beck, Chair
Associate Professor Srinivasa Salapaka
Professor Rayadurgam Srikant
Associate Professor Prashant Mehta
Assistant Professor Alex Olshevsky

Abstract

The work in this thesis presents methods for clustering and aggregation of large dynamic networked systems, typically consisting of numerous units/subsystems with complex interactions. Networked system models are used in many scientific and real world applications, for example to describe functional relationships among neurons in the brain, to achieve consensus in the design of communication and controller actuation rules in multi-agent systems, and to estimate multiple service demands in web-based software systems in order to enhance service quality by web cluster relocation. The level of complexity in modeling, analysis, and control synthesis for these systems increases combinatorially with the number of constituent elements in the network. This thesis presents methods for obtaining concise aggregated *representations* of such systems, while capturing important and relevant interconnectivity information.

In the first part of this thesis, we develop a theoretical framework for aggregating systems that are represented by *directed weighted* graphs. In this framework, we formulate an optimization problem, with the goal of minimizing a dissimilarity function that captures the distance between the *representative* and the original graphs. We propose a class of *dissimilarity measures* and introduce the notion of *composite graph sets* allowing us to compare directed weighted graphs that contain different numbers of nodes. The dissimilarity measures capture node similarities based on node connectivity, and in the simplest case reduce to metrics previously defined for equal-sized undirected unweighted graphs. The representative graph is determined by systematically identifying and aggregating *similar nodes* of the original graph into *supernodes*, and then determining the *inter supernode connectivities*. The key challenge herein is in overcoming the computational complexity that arises in iteratively solving two combinatorial optimization problems corresponding to evaluating and minimizing the cost function (dissimilarity measure). In our formulation, using the *composite graph sets*, only a single optimization problem is needed in every iteration. Further, we

introduce a maximum entropy based soft aggregation approach for node clustering, and propose a multi-scaled aggregation method whose central part incorporates the Deterministic Annealing algorithm. Specifically, we solve a sequence of relaxed minimization problems by allowing soft cluster associations; as we gradually decrease the level of softness, the solution for the original problem is approached. We discuss graph structures for which this aggregation method is provably effective, and provide comprehensive simulation examples demonstrating this efficacy.

As a special case, we study the reduction of Markov chains. By viewing a Markov chain as a directed weighted graph, the graph aggregation techniques we propose are directly applicable. If we consider nearly completely decomposable Markov chains, that is, chains whose transition matrices $\mathbf{P}$ can be written as a perturbation added to a completely decomposable (reducible) chain $\mathbf{P}^*$, i.e., $\mathbf{P} = \mathbf{P}^* + \epsilon\mathbf{C}$, then we provide sufficient conditions under which our method guarantees recovery of the decomposable sub-chain structure indicated by $\mathbf{P}^*$, thus yielding easily verifiable conditions to corroborate results given by perturbation theory. We then derive upper bound on how close the stationary distribution of aggregated Markov chain is to the aggregated stationary distribution of the original Markov chain. The effectiveness of this method has been demonstrated through numerous examples at a variety of scales. We further apply this aggregation method to reduce systems consisting of interactive stochastic processes that can be represented by graph models. In particular, we study seismic activities at different geological locations, with parametric generative models characterizing the influences among every location. In this scenario, ensemble interactions within the system can be discovered by clustering the underlying graph model.

In the second part of this thesis, we consider *dynamic* clustering and coverage problems. The goal is to use a small number of *resources* to provide *continuous and sufficient coverage* for a large number of *moving objects*. We integrate control-theoretic methods, specifically Lyapunov-based analysis, into our aggregation framework. Specifically, we adaptively compute the actuation rules for all resources to achieve the tracking objective, in addition to a set of optimal resource locations for fixed time-instances. For systems where moving objects are driven by *acceleration fields*, we show that a *dynamic* control is *necessary* to achieve dynamic coverage, and provide closed-form solutions for the resource dynamics. The algorithms we propose guarantee asymptotic tracking of cluster group dynamics, and we further establish continuity and boundedness of the corresponding

control laws. The algorithm has been successfully applied to many systems with a large number of dynamic sites, and we provide examples in this thesis to corroborate the results.

To my parents.

Acknowledgments

I would like to express my deepest gratitude to my advisors, Professor Carolyn Beck and Professor Srinivasa Salapaka, for their insightful coaching, continuous support and warm encouragement during my graduate study and research. I always regard them as my role models, for their passion and dedication to work, and their promotion and patience with students. I feel fortunate to have the opportunities to work with them, they have spent tremendous time and energy to guide me in research, from building a solid foundation, to maintaining integrity and rigorousness, then to pursing inspiration. Their influence is beyond research, they are always open-minded and considerable, and generous to offer advice during my hard times. I could not have imagined having a better advisor or mentor, and I will always cherish my experience as a graduate student under their supervision.

I am deeply grateful to Professor Rayadurgam Srikant, Professor Prashant Mehta and Professor Alex Olshevsky for serving as my committee members and for their support and valuable comments that have greatly improved my work. The six years in the University of Illinois is truly my greatest time. It has been my great honor and pleasure to have the opportunities to interact with a lot of brilliant people and to explore many interesting areas; these have made me a better person.

I would like to thank Dr. Puneet Sharma, for his pioneering work that established a sound foundation for my first research project; his generous suggestions helped me to gain a better understanding and avoid many mistakes in the beginning of my research. I also owe many thanks to to my current and former labmates, Gayathri, Xuemeng, Mashrafi, Mayank, Ram Sai, Pratik, Lucas, Chandra and Dr. Dimitrios Katselis, for all the valuable discussions and joyful casual chats.

Throughout the years in Urbana-Champaign, I have been very fortunate to know many wonderful people, many of them have become my life-long friends. Thank you Qiaomin, Yun, Peibei, Bin, Xiaoyu, Tao, Rui, Xun, Ji, Quan, Jiaming, Lili, James, Ge and Liang, for all the happy times

we have spent together, the “CSL Family” will always be a shining memory for me. My thanks also to Kun, Dayu, Yu, Yangmin and Nanjun, for their constructive suggestions that have helped me out of many predicaments.

Finally, I owe my greatest gratitude to my family. This journey would not have been possible without them. First deepest thanks to my parents, for making many sacrifices and granting me the time and freedom to grow up, for always holding belief in me, and for their unconditional love. Then, to my husband Shan, who has always stood by me, taken care of me and endured my complaints, for his love, support and long-time companionship, I would not have been the person I am without him.

Table of Contents

List of Figures

List of Abbreviations

MEP	Maximum Entropy Principle
DA	Deterministic Annealing
KL	Kullback-Leibler
MC	Markov Chains
CD	Completely Decomposable
NCD	Nearly Completely Decomposable
DME	Dynamic Maximum Entropy
SM	Shi and Malik
FBF	Frame-by-Frame

List of Symbols

$\mathcal{G}_X(\mathcal{V}_x, \mathcal{E}_x, \mathbf{X})$	Graph with node set $\mathcal{V}_x$, edge set $\mathcal{E}_x$, edge weight matrix $\mathbf{X}$, abbr., $\mathcal{G}_X$
$\mathbf{x}(i)$	The i^{th} row of matrix $\mathbf{X}$
$\boldsymbol{\mu}^X$	Node weight vector for $\mathcal{G}_X$
$\boldsymbol{\mu}$	Node weight vector (no confusion)
ϕ	Partition function
$\mathbf{\Phi}$	Aggregation matrix
ψ	Bijective partition function (or permutation function)
$\mathbf{\Psi}$	Permutation matrix
$\mathcal{C}_{XY}^{\phi}(\mathbf{Z})$	Composite graph generated by $\mathcal{G}_X$ and $\mathcal{G}_Y$, specified by partition function ϕ and edge weight matrix $\mathbf{Z}$
$\boldsymbol{\mathcal{C}}_{XY}$	Composite graph set generated by by $\mathcal{G}_X$ and $\mathcal{G}_Y$
$\mathbb{I}$	Indicator function
$\mathbf{1}_n$	All 1 vector in $\mathbb{R}^n$
$\mathbf{I}_n$	Identity matrix of dimension n-by-n
$\mathrm{Tr}(\mathbf{X})$	Trace of matrix $\mathbf{X}$
$\mathrm{diag}(\mathbf{x})$	Diagonal matrix with the elements of vector $\mathbf{x}$ on the main diagonal
$\mathcal{X}$	Stochastic process with values $X(k)$
$\mathbf{P}$	Transition matrix of a Markov chain
$\mathbf{P}^*$	Transition matrix of a completely decomposable Markov chain
$\mathbf{C}$	Perturbation matrix
$\mathbf{P}_{IJ}$	The $(IJ)^{th}$ block of of matrix $\mathbf{P}$
$P_{i_I j_J}$	The ij^{th} entry of $\mathbf{P}_{IJ}$
d°	Unspecified distance function for two arbitrary vectors

$\rho^{\circ}_{\Phi}(\mathcal{G}_X, \mathcal{G}_Y)$	Unspecified $\Phi-$dissimilarity function between $\mathcal{G}_X$ and $\mathcal{G}_Y$
$\nu(\mathcal{G}_X, \mathcal{G}_Y)$	Dissimilarity measure between $\mathcal{G}_X$ and $\mathcal{G}_Y$
$\tilde{\mathbf{\Phi}}$	Soft aggregation matrix, or association weight matrix
$\rho^{\circ}_{\tilde{\Phi}}(\mathcal{G}_X, \mathcal{G}_Y)$	Soft unspecified $\Phi-$dissimilarity
$d_{\phi}(\mathcal{G}_X, \mathcal{C}^{\phi}_{XY}(\mathbf{Z}))$	$\Phi-$distance defined through the composite graph $\mathcal{C}^{\phi}_{XY}(\mathbf{Z})$ for equal-sized graphs
$d_{\phi}(\mathcal{G}_X, \mathcal{C}^{\tilde{\phi}}_{XY}(\mathbf{Z}))$	Soft $\Phi-$distance defined through the composite graph $\mathcal{C}^{\phi}_{XY}(\mathbf{Z})$ for equal-sized graphs
$\rho_{\Phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y)$	$\Phi-$dissimilarity function induced by composite graph $\mathcal{C}^{\phi}_{XY}(\mathbf{Z})$
$\tilde{\rho}_{\tilde{\Phi},\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y)$	Soft $\Phi-$dissimilarity function induced by composite graph $\mathcal{C}^{\tilde{\phi}}_{XY}(\mathbf{Z})$
$\mathbf{D}$	Distance matrix for two set of vectors in the same vectors space
$H(\mathcal{C}^{\tilde{\phi}}_{XY}(\mathbf{Z})\|\mathcal{G}_X)$	Entropy
$F(\mathcal{G}_X, \mathcal{C}^{\tilde{\phi}}_{XY}(\mathbf{Z}))$	Free energy
$\mathbf{R}$	Posterior association weight matrix
$p_{j\|i}$	Association weight between the i^{th} node and the j^{th} supernode $(= [\mathbf{\Phi}]_{ij})$
$q_{i\|j}$	Posterior association weight between the i^{th} node and the j^{th} supernode $(= [\mathbf{R}]_{ij})$
$\mathcal{S}$	Set of shrunk (truncated) regions in decentralized algorithm
$\Delta F, \Delta^2 F$	First and second variation of function F
$\lambda_{\min}(\mathbf{X})$	The minimum eigenvalue of matrix $\mathbf{X}$
δ	Maximum degree of coupling
η	Index of unbalance of a matrix
df	Degree of Freedom
$\chi^2(df)$	χ^2 distribution with degree of freedom equals to df
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and standard deviation σ
$Poi(\lambda)$	Poisson distribution with parameter λ
$\otimes$	Kronecker product
$\boldsymbol{\pi}^T(\mathbf{P})$	Row vector corresponding to the stationary distribution of Markov chain with transition matrix $\mathbf{P}$
$\mathbb{P}$	Probability

$\mathcal{H}_M$	History of a random process up to the M^{th} step
$\boldsymbol{\gamma}$	Acceleration field
$\boldsymbol{\xi}$	Concatenated state vector
$\boldsymbol{\xi}_{cl}$	Augmented closed-loop state vector
$\mathbf{x}_3^c, \mathbf{x}_4^c$	Instantaneous centroid of site locations and velocities

Chapter 1

Introduction

1.1 The Graph Clustering Problem

1.1.1 Historical overview

In studies of large interconnected systems such as economics [11, 62], social networks [21], neuroscience [37, 52], Internet performance [39, 64] and multi-agent systems [15, 45], directed weighted graph models are frequently used to capture complex dynamical interactions. However, both physical modeling and data-based modeling methods typically yield large models with numerous nodes and complex interactions represented by edges; this makes the analysis of fundamental system behavior intractable. In systems research, it has long been a goal to approximate large complicated system models by succinct and more tractable models, while maintaining the underlying *ensemble* or *dominant* behaviors. For graph models in particular, a simple *representative graph* that is *similar* to a given graph under some notion of similarity is pursued. As an important special case of directed weighted graphs, the reduction of large Markov chains by itself arises in numerous applications areas.

A major class of existing graph-simplification approaches have built on the concept of graph partitioning, where subgraphs are identified within a large graph based on how strongly or weakly they are connected to the rest of the graph [57]. Most graph partition algorithms use "cut-based" methods, which typically require computation of eigenvalues and eigenvectors of the large adjacency or Laplacian matrices associated with the graphs [61, 67]. These methods become increasingly intractable as the size of graph grows, since the corresponding spectral decomposition becomes computationally challenging. Moreover, cut-based algorithms usually produce a series of embedded bipartitions which do not always lead to representations of actual clusters. The other effective

while relatively less common approach to simplify graphs is through graph clustering, which aims to aggregate similar nodes of a graph into supernodes [57, 70, 73]. In fact, there are algorithms that convert clustering problems into graph partitioning problems and then apply spectral methods; only a few address the opposite direction [14]. In this thesis, we mainly focus on the clustering based approach for graph aggregation.

In forming the graph clustering problem, that is determining a smaller representative graph for a given large and complicated graph, an optimal node aggregation is sought, where optimality is defined and evaluated based on some distance measure quantifying similarity in node connectivities. There is no universally applied distance function for general directed weighted graphs. Early development of similarity/distance measures between graphs was mainly based on graph isomorphisms, including graph isomorphism identification [9]; the edit distance given by the minimum number of adding/deleting edge operations required to obtain one graph from another [47]; the distance characterized by the maximum common subgraph or the minimum common supergraph [20]. Similarity notions that reflect neighborhood similarity have become popular in the fields of Internet analysis and social networking, and many statistical iterative updating methods have been developed [6, 39]. Another large class of graph similarity notions are defined through graph node matching or subgraph embedding [8, 35]. Nevertheless, these methods either apply to graphs with the same number of nodes (graph isomorphisms), or deal with unweighted, typically undirected graphs (distances based on editing or sub/sup-graphs).

More recently in the domain of control and information theory, specific forms of graph distances have been proposed in the statistical inference, or probabilistic, setting, such as the variation-of-information distance for comparing clustering (aggregation) results on the same dataset [46], the Kullback–Leibler (K-L) divergence rate between two Markov chains [54], and an information-based metric between two probability distributions [65]. Most of these approaches are compatible with undirected unweighted graphs but not typically applicable to weighted and/or directed graphs.

The problem of Markov chain reduction has been solved by specific tools, which can be broadly classified into two categories: balanced truncation [40] or state aggregation [13, 62]. A review on model reduction methods (mainly balanced truncation) for Markov chains can be found in [5]. There exists substantial analogue between the state aggregation approach and general graph clustering.

Possibly the most well-known state aggregation methods for Markov chains are based on singular perturbation and spectral analysis [2, 51, 62]. In [62], the authors proposed an aggregation rule which identifies sub-chains when the large systems are *completely decomposable (CD)* or *nearly completely decomposable (NCD)* Markov chains. A simulation-based aggregation method has been proposed recently which employs information theoretical analysis and adaptively attains better aggregation [13].

In general, the definition and evaluation of graph similarity incur the well known combinatorial issue, which typically yields the mathematical formulations of graph aggregation NP-hard problems [1]. Many formulations pose combinatorial optimization problems, whose cost surfaces typically comprise of many local minima [24]. Amongst the major clustering methods, most are based on k-means type clustering algorithms [43, 57, 73], which usually get trapped in poor local minima quickly, and therefore require many implementation runs to achieve a satisfactory result. Another method is the well-known heuristic algorithm, simulated annealing which avoids poor minima, but typically requires very large computation times [23, 73]. Alternatively, the deterministic annealing (DA) algorithm, which is an iterative method characterized by an annealing parameter, detects underlying clusters hierarchically and finer (possibly multiple) sub-clusters are identified as the algorithm progresses. The main advantage of developing the clustering methods by DA algorithm is two-fold: effectiveness and efficiency. Since the DA builds upon the maximum entropy principle (MEP), there is no need to guess a starting point for the algorithm, and as a result repeated implementations with different initial settings are not necessary. Moreover, the incorporation of the annealing process is designed to avoid poor local minima with fewer iterations by adopting a geometric annealing rule. An inherent phase-transition process induces natural splits of supernodes, which do not have the inconsistencies associated with the embedded bipartitioning aspect of spectral decomposition [55, 60, 72].

1.1.2 Main contributions of Part I

In this thesis, we focus on developing a meaningful similarity measure for general directed weighted graphs, then provide a framework to determine a coarse graph by aggregating nodes that are similar in terms of their connectivity to the rest of the graph. We apply this framework to aggregation of

Markov chains. In Part I, our contributions are:

1) Development of a dissimilarity measure for general directed weighted graphs of (possibly) different sizes. This formulation is motivated by many applications including those listed in the beginning of this chapter, used to systematically evaluate aggregation performance, and is consistent with existing graph distance measures for undirected unweighted graphs.

2) Reformulation of the graph clustering problem in an optimal *resource allocation* framework, which is then addressed by incorporating soft partitioning and the deterministic annealing (DA) algorithm. Presentation of a guided decentralized modification to improve algorithm scalability by gradually shrinking the ranges of calculation.

3) Application of the graph clustering approach to aggregate large Markov chains. Detailed investigation of the nearly decomposable Markov chain, and proposition of sufficient conditions under which the MEP-based state aggregation guarantees to recover the sub-chain decomposition. Determination of bounds on the approximation error between the aggregated stationary distribution of the original Markov chain and the stationary distribution of the aggregated Markov chain.

4) Adaptation of the method to aggregate interconnected systems consisting of a large number of stochastic processes described by parametric models, and identification of functional interconnections in specific.

1.2 Clustering for Dynamic Systems

1.2.1 Historical overview

The study of clustering and coverage control problems with dynamic objects arises in various applications, such as placement of autonomous sensors to perform distributed sensing tasks in a dynamic environment with the goal of achieving and maintaining a required sensor coverage criterion [10, 22, 30]; modeling CPU and database demands and their fluctuations in web-based software engineering [25]; and identifying the centroidal evolution in clusters within massive dynamic datasets containing varying features in database research [74]. These problems can be described in terms of dynamics of multiple objects (or *sites*) in a (general) domain with the main objective being to identify the *group dynamic* properties. Specifically, this goal can be viewed as a two-fold

task: (1) partition the set of sites and place a *resource* in each cell of the partition, such that the averaged distance from a site to the nearest resource is minimized; (2) control the resource dynamics such that they track the corresponding dynamic cells.

The first task requires real-time decision-making and can be viewed as a *static* clustering or facility allocation problem, which has been studied in various contexts, such as coding, vector quantization and statistical learning [17, 24, 55]. In particular, this static clustering problem can be viewed and formulated as a resource allocation problem, and shares a common foundation with the graph node aggregation problem introduced in Section 1.2; thus the same challenges and constraints apply here. The second task adds further computational and design difficulties to the static problem by introducing site dynamics. Specifically, the main design challenges arise from the associated tracking problems of the dynamic clusters and their interactions.

Problems related to dynamic coverage are considered in [10, 22, 30], where the emphasis is on distributed implementations, i.e., under limited information flow between individual elements. Therefore, these algorithms are sensitive to the initial placement of the resources and suffer from drawbacks analogous to those found in Lloyd's algorithm. Also, these problems generally assume the underlying sites are static or fixed, and focus only on the dynamics of the resources; thus the resulting goal is to determine for each resource its optimal location and the path to reach its destination [10, 30]. In contrast to the distributed approach, there is scant research that seeks algorithms of a non-distributed nature that aim simultaneously to attain global solutions and maintain low computational expense in a dynamic environment. In [60], a Dynamic Maximum Entropy (DME) framework is proposed that treats dynamic coverage of mobile sites under given *velocity* fields by designing corresponding velocity fields for the resources.

This part of the work in this thesis has close parallels to [60] in terms of problem formulation and objectives, however, we extend the class of site dynamics and develop altogether new control strategies. To successively overcome the local dependence of many distributed algorithms, we adopt a soft partitioning approach from the DA algorithm and associate each site to multiple resources via nonnegative association weights. The computational challenge induced by the underlying site dynamics is addressed by adopting an *energy function* that approximates the coverage metric as the *control Lyapunov function* of the system. Implementation of the proposed method is shown

to be computationally less expensive than implementing repeated *static*-clustering of the data at fixed time instances.

1.2.2 Main contributions of Part II

For the dynamic clustering problem, we consider systems with general and realistic dynamics, by allowing for acceleration fields and control. Specifically, the trajectories of the resources can be manipulated through design of their acceleration fields. This generalization is motivated mainly by multi-vehicle systems, such as those that may arise in disaster relief, search and rescue, and reconnaissance operations. We propose a new constructive *dynamic* control law that satisfies the tracking and coverage requirements in such systems. Our main contribution in Part II can be summarized as follows:

1) Proof that a dynamic control law is necessary to track the centroidal dynamics of the mobile objects when their motions are driven by acceleration fields.

2) Development of a nonlinear dynamic feedback control law for the resources, under which asymptotic tracking of cluster centers is achieved under mild conditions.

3) Proof that the constructive control law is non-conservative, that is, if asymptotic tracking can be achieved by some Lipschitz control near the instantaneous cluster center, then the proposed control is also Lipschitz and bounded.

1.3 Organization

Main content of this thesis has been separated to Part I and Part II, with emphasize on static and dynamic problems, respectively. The organization is as follows. In Chapter 2, we review the resource allocation problem and overviewed its solution based on the MEP and the DA algorithm, which are the foundation of our solution method. We then introduce the problem of graph clustering via the aggregation, by clarifying clarify the key issues that need to be addressed in formulating and solving the problem, and propose a solution approach in resource allocation framework. In Chapter 3, we formally define a class of dissimilarity measures for directed weighted graphs of possibly different sizes and establish the mathematical rigorousness of the graph clustering formulation. In Chapter 4, we develop a practically efficient and effective solution method based on the MEP for

the graph aggregation problem defined with the proposed graph dissimilarity measures. We discuss the connectivity structures that can be identified using this method. Moreover, we study in parallel the state aggregation of Markov chain aggregations. As special example, in Chapter 5.1, we develop sufficient conditions under which sub-chains of a NCD Markov chain can be successfully identified, and characterize the steady state approximation error of the aggregation result. In Chapter 6, we consider an interesting application of graph clustering by investigating the functional aggregation of a class of parametric models for stochastic processes. Specifically, we propose a mapping that converts the interconnected systems into directed weighted graphs, and a special distance function defined by the parametric models of all stochastic processes, allowing us to cast this into the general graph aggregation setting. The dynamic coverage and clustering problem is presented in Part II. Some useful material in control theory, focusing on Lyapunov stability results, is provided in Appendix A. In Chapter 7, we provide a treatment for the objective of tracking cluster centers in dynamic coverage control problems. We show the necessity of adopting a dynamic control law when the site dynamics are driven by acceleration fields. An explicit form of control law is developed, for which the effeteness and boundless results are established. Finally we conclude the thesis by revisiting some key results. In the end of every chapter, we briefly summarize the highlighted results point-by-point.

Part I

Aggregation of Graph Models

Chapter 2

Overview of Graph Clustering via Aggregation

2.1 Mathematical Preliminaries

We briefly introduce some mathematical notations and facts frequently used in Part I.

2.1.1 Notations

Vectors and matrices

We use boldface uppercase to denote matrices, boldface lowercase to denote vectors, regular font uppercase for random variables, and regular type lowercase for scalars, respectively. Sets and subspaces are represented by calligraphic letters.

- For a finite set $\mathcal{S}$, we denote its cardinality by $|\mathcal{S}|$.
- $\mathbb{R}$ and $\mathbb{R}_+$ denote the set of real numbers and nonnegative real numbers respectively. The vector in $\mathbb{R}^n$ with all entries being 1 is denoted by $\mathbf{1}_n$, an n-by-n identity matrix is denoted by $\mathbf{I}_n$, and $\mathbf{e}_i \in \mathbb{R}^n$ denotes the i^{th} column of $\mathbf{I}_n$ for $1 \leq i \leq n$.
- For a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, we denote the entry at the i^{th} row the j^{th} column by X_{ij}, and the i^{th} row of $\mathbf{X}$ by $\mathbf{x}(i) \in \mathbb{R}^{1 \times n}$.
- For a vector $\mathbf{x} \in \mathbb{R}^n$ and an integer $1 \leq p < \infty$, the vector $p-$norm is given by

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}},$$

and for $p = \infty$,

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

Graphs

– We denote a weighted directed graph by a triple, $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$, in which $\mathcal{V}, \mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ and $\mathbf{W} \in \mathbb{R}_+^{|\mathcal{V}| \times |\mathcal{V}|}$ denote the set of *nodes*, *edges* and the *edge weight matrix*, respectively. We assume that $|W_{ij}| < B < \infty$ for some finite B and for all (i, j) pairs. Define the node weights to be $\{\mu_i\}$ for $i = 1, \ldots, |\mathcal{V}|$, satisfying $\mu_i \geq 0$ with $\sum_i \mu_i = 1$. We define the *outgoing vector* of the i^{th} node, $\mathbf{w}(i)$, by the weights on its outgoing edges, i.e., $\mathbf{w}(i) \stackrel{\triangle}{=} [W_{i1}, W_{i2}, \cdots, W_{i|V|}]$. Here $W_{ij} = 0$ if and only if there is no edge from the i^{th} node to the j^{th} node.

– A graph $\mathcal{G}_X(\mathcal{V}_x, \mathcal{E}_x, \mathbf{X})$ is *isomorphic* to another graph $\mathcal{G}_Y(\mathcal{V}_y, \mathcal{E}_y, \mathbf{Y})$, if $|\mathcal{V}_x| = |\mathcal{V}_y|$ and there exists an edge-weight-preserving bijective mapping $\psi : \mathcal{V}_x \to \mathcal{V}_y$, such that the directed edge $(i, j) \in \mathcal{E}_x$, if and only if $(\psi(i), \psi(j)) \in \mathcal{E}_y$, and the edge weights satisfy $X_{ij} = Y_{\psi(i)\psi(j)}$ for all ordered pairs (i, j); we denote this isomorphism by $\mathcal{G}_X \simeq \mathcal{G}_Y$.

– Let $\mathcal{N} = \{1, 2, \ldots, n\}$ and $\mathcal{M} = \{1, 2, \ldots, m\}$ be two index sets with $|\mathcal{N}| = n$ and $|\mathcal{M}| = m$, assuming $n \geq m \geq 2$. A *(hard) partition function* $\phi : \mathcal{N} \to \mathcal{M}$ is a map from $\mathcal{N}$ *onto* $\mathcal{M}$, such that $\phi^{-1}(\mathcal{M})$ is a partition of $\mathcal{N}$. That is, for any $1 \leq j \neq k \leq m$, $\phi^{-1}(j) \subset \mathcal{N}$ is non-empty, $\phi^{-1}(j) \bigcap \phi^{-1}(k) = \emptyset$ and $\bigcup_{j=1}^{m} \phi^{-1}(j) = \mathcal{N}$.

– Each partition function $\phi : \mathcal{N} \to \mathcal{M}$ defines an *aggregation matrix* $\mathbf{\Phi} \in \{0, 1\}^{n \times m}$ as

$$\Phi_{ij} = [\mathbf{\Phi}]_{i,j} = \begin{cases} 1 & \text{if } \phi(\text{i}) = \text{j}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

Therefore, $\mathbf{\Phi}\mathbf{1}_m = \mathbf{1}_n$, and the k^{th} column of $\mathbf{\Phi}$ equals $\sum_{i \in \phi^{-1}(k)} \mathbf{e}_i$. For a vector $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^T\mathbf{\Phi}$ is a vector in $\mathbb{R}^m$, indicating the aggregation of $\mathbf{x}$ specified by partition ϕ.

– When $m = n$, a partition function becomes an index *relabeling, or permutation.* In this scenario, we denote the *bijective* partition function and the square aggregation (or permutation) matrix by ψ and $\mathbf{\Psi}$. Further, if $\mathcal{G}_Y$ is obtained by relabeling the nodes of $\mathcal{G}_X$, then $\mathcal{G}_X \simeq \mathcal{G}_Y$.

2.1.2 Markov Chains

A *(row) stochastic matrix* is a square matrix of nonnegative real numbers, with each row summing to 1. Similarly, a *column stochastic matrix* is a square matrix of nonnegative real numbers, with each row summing to 1. A *doubly stochastic matrix* is both row-stochastic and column-stochastic.

Let $\mathcal{X} = \{X(0), X(1), \ldots, X(k), \ldots\}$ be a homogeneous discrete Markov chain on a finite state space $\mathcal{N} = \{1, 2, \ldots, n\}$ with *transition matrix* $\mathbf{P} \in \mathbb{R}^{n \times n}$. Then $X(0) = x_0 \in \mathcal{N}$ is the initial state of the Markov process, $X(k)$ is the state value at time step k, and $\mathbf{P}$ is a (row) stochastic matrix whose ij^{th} element denotes the transition probability from the i^{th} state to the j^{th} state, i.e., $P_{ij} = [\mathbf{P}]_{i,j} = \mathbb{P}(X(k+1) = j | X(k) = i)$ for any nonnegative integer k.

We adopt standard notation from the Markov chain literature and the notation from [11,29,62] for *completely decomposable (CD)* and *nearly completely decomposable (NCD) Markov chains*. A CD Markov chain can be aggregated to several non-communicating subchains, in other words, the transition matrix $\mathbf{P}^*$ of a CD Markov chain is block-diagonal stochastic matrix

$$\mathbf{P}^* = \begin{bmatrix} \mathbf{P}_1^* & & & 0 \\ & \ddots & & \\ & & \mathbf{P}_I^* & \\ & & & \ddots \\ 0 & & & \mathbf{P}_N^* \end{bmatrix},$$

with every principal submatrix $\mathbf{P}_I^*$ being a stochastic matrix of dimension $n_I I = 1, \ldots, N$. The transition matrix of an NCD Markov chain can be viewed as a perturbation of a transition matrix of a CD Markov chain $\mathbf{P}^*$, that is, for a NCD Markov chain with $\mathbf{P}$, we have

$$\mathbf{P} = \mathbf{P}^* + \epsilon \mathbf{C}, \tag{2.2}$$

in which $\mathbf{C}$ is the perturbation matrix of the same dimension as $\mathbf{P}^*$, and ϵ is a small positive real number. The block structure of $\mathbf{P}^*$ provides a natural partition on the state space $\mathcal{N}$, where the

states associated to the K^{th} diagonal block are in the K^{th} sub-chain

$$\phi^{-1}(K) = \{\sum_{I=1}^{K-1} n_I + 1, \ldots, \sum_{I=1}^{K} n_I\}, \quad 1 \leq K \leq N.$$

Thus $n = \sum_{I=1}^{N} n_I$. A completely decomposable Markov chain can be aggregated according to these groups, and $\mathbf{P}_I^*$ is the transition matrix for the I^{th} sub-chain, specifically, the block structure of $\mathbf{P}^*$ determines the block structures of both $\mathbf{P}$ and $\mathbf{C}$, and we denote the submatrix of $\mathbf{P}$ consisting of the rows in $\phi^{-1}(I)$ and columns in $\phi^{-1}(J)$ by $\mathbf{P}_{IJ}$. We denote the ab^{th} entry of $\mathbf{P}_{IJ}$ by $P_{a_I b_J}$. To ensure the stochasticity of the resulting $\mathbf{P}$ matrix, entries of $\mathbf{C}$ need to satisfy the following constraints (similar to those defined in [11])

$$\begin{cases} \sum_{k=1}^{n_I} C_{i_I k_I} = -\sum_{J \neq I} \sum_{j=1}^{n_J} C_{i_I j_J}, \\ C_{i_I j_J} \geq 0, \quad 1 \leq i \leq n_I, \forall j, 1 \leq I, J \leq N, I \neq J. \end{cases} \tag{2.3}$$

From these constraints, we see that the diagonal blocks of $\mathbf{C}$ must contain negative entries. Therefore, the size of perturbation ϵ has an upper bound under which the non-negativity of P is guaranteed.

We further assume that each stochastic submatrix $\mathbf{P}_I^*$ corresponds to an *aperiodic* and *irreducible* Markov chain, and thus has a *unique* stationary distribution $\boldsymbol{\pi}^T(\mathbf{P}_\mathbf{I}^*) \in \mathbb{R}_+^{n_I}$. Note that $\boldsymbol{\pi}^T(\mathbf{P}_\mathbf{I}^*)$ is also the left eigenvector of $\mathbf{P}_I^*$ associated to the (unrepeated largest) eigenvalue 1 (by the *Perron-Frobenius Theorem* [29]). Let the stationary distribution of the Markov chain $\mathcal{X}$ be $\boldsymbol{\pi}^T(\mathbf{P}) \in \mathbb{R}_+^n$.

2.2 Resource Allocation Problems

In this thesis, aggregation problems in different spaces are viewed as a specific class of *resource allocation* problems. This viewpoint forms the common theme which facilitates a common approach to solve the different aggregation problems. In this section, we lay down the basic resource allocation problem and the approach that is central to the solution of the aggregation problems in this thesis.

Resource allocation problems arise in a wide variety of fields, such as vector quantization [24], pattern recognition [31], facility allocation [17], marketing analysis [19], and drug discovery [59], just to name a few. In all these applications, we try to partition a large set of n objects (i.e.,

sites or nodes) into $m(< n)$ mutually disjoint subsets, and assign each subset a single *resource* as a representative of the subset of objects. We adopt the concept of *distortion* from the data compression literature to measure instantaneous coverage. Thus the objective is to minimize this distortion function by determining the partitioning and the resources. We provide the mathematical formulation as follows.

Let $\{r\}_j, j = 1, \ldots, m$ and $\{s\}_i, i = 1, \ldots, n$ denote the sets of properties, for example, the positional coordinates, of the m resources and the n objects where $r_i, s_j \in \mathbb{R}^d, \forall i, j$. Let $\{\mu_i\}, i = 1, \ldots, n$ denote the relative weights for the objects (i.e., $\mu_i \geq 0, \forall i$ and $\sum_{i=1}^{n} \mu_i = 1$); this relative weighting may reflect, for example, a probabilistic distribution of the objects. The resource allocation (clustering) problem can be written as

$$\min_{r_j, 1 \leq j \leq m} D(s, r) \quad = \quad \min_{r_j, 1 \leq j \leq m} \sum_{i=1}^{n} \mu_i d(s_i, r(s_i)), \tag{2.4}$$

where the cost function $D(s, r)$ is the distortion to be minimized, $r(s_i)$ is the resource assigned to the i^{th} object, and $d(\cdot, \cdot)$ is a distance measure to evaluate the dissimilarity of a resource and an object; a popular choice is the squared Euclidean distance. Therefore, $D(s, r)$ evaluates the averaged dissimilarity between an object and its representative resource. The above resource allocation problem can be rewritten in terms of the partition function $\phi : \{1, \ldots, n\} \to \{1, \ldots, m\}$ or the aggregation matrix $\mathbf{\Phi} \in \{0, 1\}^{n \times m}$ as

$$\begin{aligned} \min_{r_j, 1 \leq j \leq m} D(s, r) \quad &= \quad \min_{\phi;\, r_j, 1 \leq j \leq m} \sum_{i=1}^{n} \mu_i d(s_i, r_{\phi(i)}) \\ &= \quad \min_{\mathbf{\Phi} \in \{0,1\}^{n \times m};\, r_j, 1 \leq j \leq m} \sum_{i=1}^{n} \mu_i d(s_i, r_j) \Phi_{ij}. \end{aligned} \tag{2.5}$$

The goal of clustering is to find the global optimum or a good approximation to the above problem (2.5). The optimal solution $(\phi^*, \{r_j^*\})$ satisfies two necessary conditions:

(1) *Nearest neighborhood partition:* or Voronoi partition, in which the partition is given by $\phi_i^* = \arg\min_k d(s_i, r_k^*)$, so that the problem satisfies

$$\min_{r_j, 1 \leq j \leq m} D(s, r) \quad = \quad \min_{r_j, 1 \leq j \leq m} \sum_{i=1}^{n} \mu_i \min_{1 \leq k \leq m} d(s_i, r_k). \tag{2.6}$$

(2) *Centroid resource:* every optimal resource r_j^* must be located at the "centroid" of the jth cell of the partition, where the notion of centroid is defined through the pairwise distance $d(s_i, r_k)$.

It is well known that (2.5) is a combinatorial optimization problem, and has NP-hard complexity [1,42,44]. The cost function $D(s,r)$ is non-convex and typically riddled with multiple local minima. Many popular descent algorithms, such as *Lloyd's* or *k-means* [43], focus on iterative optimization between the two necessary conditions. These algorithms typically get trapped in local minima, and are sensitive to the initial placement of resources. The sensitivity to initial resource locations can be explained from the non-convexity of the distortion measure $D(s,r)$ and that in these algorithms changes in object locations typically effect only *nearby* partition cells and have virtually no effect on *distant* cells [55].

In order to address these problems, we allow soft partitioning, or allow each object s_i to associate with all resources r_j with *association weights* $p_{j|i}$ (satisfying $0 \leq p_{j|i} \leq 1$ and $\sum_{j=1}^{M} p_{j|i} = 1$. See Figure 2.1 for illustration). Therefore, we have changed the (hard) distortion function (2.5) to the soft distortion function given by

$$\tilde{D}(s,r) = \sum_{i=1}^{n} \mu_i \sum_{j=1}^{m} p_{j|i} d(s_i, r_j), \tag{2.7}$$

in which the objective becomes minimizing the weighted average distance of an object to all resources where the weights are specified by $p_{j|i}$. If we define a soft aggregation matrix $\tilde{\mathbf{\Phi}} \in [0,1]^{m \times n}$ by $\tilde{\Phi}_{ij} = p_{j|i}$, that is, the i^{th} row of $\tilde{\mathbf{\Phi}}$ indicates the association of object i to every resource respectively, then the soft clustering problem can be expressed as

$$\begin{aligned} \min_{\tilde{\mathbf{\Phi}};\, r_j, 1 \leq j \leq m} \tilde{D}(s,r) \quad &= \quad \min_{\tilde{\mathbf{\Phi}};\, r_j, 1 \leq j \leq m} \min_{\tilde{\mathbf{\Phi}}} \sum_{i=1}^{n} \mu_i d(s_i, r_j) \tilde{\Phi}_{ij} \\ &\text{s.t.} \qquad \tilde{\mathbf{\Phi}} \in [0,1]^{n \times m}, \tilde{\mathbf{\Phi}} \mathbf{1}_n = \mathbf{1}_n \end{aligned} \tag{2.8}$$

The choice of the weights, $p_{j|i}$, is critical in accessing the trade-off between the decreasing local influence and the deviation of the distortion term (2.8) from the original cost function (2.5). We adopt the maximum entropy principle (MEP) as the criterion to determine such weights. Specifically, we use a Shannon entropy term to quantify the uncertainty of all soft associations

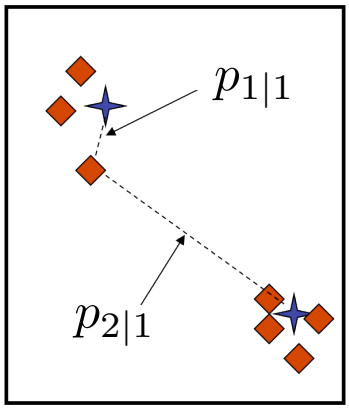


Figure 2.1: Soft association between an object (red diamond) to two resources (blue stars). The association weights $\{p_{j|i}\}$ satisfy $p_{1|1} + p_{2|1} = 1$ and $p_{1|1}, p_{2|1} \geq 0$.

$\{p_{j|i}\}$

$$H(r|s) = -\sum_{i=1}^{n} \mu_i \sum_{j=1}^{m} p_{j|i} \log(p_{j|i}). \tag{2.9}$$

The value of this entropy term lies in $[0, \log m]$, with a large value denoting high uncertainty. In particular, $H(r|s) \approx \log m$ is achieved when every object is equally associated with all resources (i.e., $p_{j|i} = 1/M, \forall i, j$) (the maximum uncertainty), and $H(r|s) \approx 0$ is achieved when every object is solely associated with one resource (thus no uncertainty). In the solution, we minimize the distortion (from soft partitioning) (2.7), while specifying the level of uncertainty by adding a constraint $H = H_0$, with $0 \leq H_0 \leq \log m$. This yields the Lagrangian minimization

$$\min_{\{p_{j|i}\},\, r_j, 1 \leq j \leq m} F(r) = \min_{r_j, 1 \leq j \leq m} \tilde{D}(s, r) - TH(r|s) \;\; (+TH_0), \tag{2.10}$$

where T is a Lagrangian multiplier. We drop the last term in the brackets since it does not depend on r_j. The Lagrangian $F(r)$ is also referred to as the *free energy*, due to a close analogy to similar constructs in statistical physics [38].

Here we provide an alternative interpretation of the Lagrangian minimization problem (2.10) by the *maximum entropy principle (MEP)* [32, 33]. If we define an *annealing parameter* as $\beta = \frac{1}{T}$,

(2.10) is equivalent to

$$\max_{\{p_{j|i}\},r_j,1\le j\le m} L(r) \stackrel{\triangle}{=} \max_{r_j,1\le j\le m} H(r|s) - \beta\tilde{D}(s,r),$$

which is the Lagrangian of the following maximum entropy problem

$$\begin{aligned} \max_{\{p_{j|i}\},r_j,1\le j\le m} \quad H(r|s) &= -\sum_{i=1}^{n}\mu_i\sum_{j=1}^{m} p_{j|i}\log(p_{j|i}) \\ \text{s.t.} \quad \tilde{D}(s,r) &= \sum_{i=1}^{n}\mu_i p_{j|i} d(s_i,r_j) \ \ (=D_0). \end{aligned} \tag{2.11}$$

The merit of the MEP is it provides a systematic method to compute a weighting function that achieves a specific feasible value of distortion, and thereby achieves a prespecified tradeoff in the above context.

In practice, the optimization problem (2.10) is solved by iteratively updating the association weights and the resource locations, or

$$\min_{r_j,1\le j\le m}\min_{\{p_{j|i}\}} F(r) = \min_{r_j,\,1\le j\le m}\left\{\min_{\{p_{j|i}\}}\tilde{D}(s,r) - \frac{1}{\beta}H(r|s)\right\}.$$

The inner minimization is solved for a set of fixed $\{r_j\}$, and this yields the Gibbs distribution

$$p_{j|i} = \frac{\exp\{-\beta d(s_i,r_j)\}}{\sum_{k=1}^{m}\exp\{-\beta d(s_i,r_k)\}}, \tag{2.12}$$

with the distance measure $d(s_i,r_j)$ to be selected. By substituting the association weights (2.12) into the free energy, the problem becomes

$$\min_{r_j,1\le j\le m} F(r) = \min_{r_j,1\le j\le m} -\frac{1}{\beta}\sum_{i=1}^{n}\mu_i\log\sum_{k=1}^{m}\exp\{-\beta d(s_i,r_k)\}, \tag{2.13}$$

the minimizer of which can be solved after selecting the d function. The two steps (2.12) and (2.13) are iteratively minimized until convergence to a solution of (2.10) at a particular β.

To approach a solution of the original problem (2.5), we adopt the deterministic annealing algorithm (DA), which incorporates an annealing process. Note that the Gibbs distribution (2.12)

indicates a one-to-one correspondence between the value of annealing parameter β and the entropy value H_0 as in (2.10). In the DA algorithm, the annealing is driven by a sequence of increasing values of β ($\beta_k < \beta_{k+1}$), which is equivalent to a decreasing value of entropy constraint $\log m > H_1 > H_2 > \cdots > 0$. In the other words, the DA algorithm iteratively determines the optimal association weights $\tilde{\mathbf{\Phi}}$ and the resources $\{r_j\}$ at each H_k value (and hence each β_k value). Both $\tilde{\mathbf{\Phi}}$ and $\{r_j\}$ are parametrized by β, therefore, as we gradually reduce the level of uncertainty in associations, the soft aggregation matrix $\tilde{\mathbf{\Phi}}$ approaches a binary-valued hard aggregation matrix. Meanwhile, we also obtain a sequence of resources $\{r_j\}$, each being local minima of $\tilde{D}(s, r)$. Finally, a desired aggregation matrix $\mathbf{\Phi}$ satisfying (2.1), and a corresponding set of resources are obtained. Detailed discussions for specific distance measures can be found in Chapter 4 and Chapter 7.

2.3 Graph Clustering as Resource Allocation Problems

The emphasis of Part I is to aggregate nodes of a large graph with similar edge connections into supernodes, so that the resulting graph with supernodes and edge connections among them, contains a reduced number of nodes and edges, and thus only retains core or ensemble inter-connections. More specifically, for a general directed weighted graph our goal is to construct a reduced-order representative graph, such that some dissimilarity measure between the two graphs is minimized. This is the first instance of the general clustering problem introduced in Section 2.2.

We define the notion of similarity by comparing node connectivities: we say two nodes in a graph are similar if they have similar edge connections to the rest of the graph. The connectivity information of a node is contained in the vector of edge weights starting from that node, or its *outgoing vector*. So if two nodes have similar outgoing vectors, we say they are similar. Note that this notion of similarity generalizes the concept of "node-matching" previously developed for undirected unweighted graphs in [6, 8, 35]. Following this logic, the objective of node clustering can be converted to clustering a set of outgoing vectors for the nodes of a given graph, which is essentially clustering data points in a vector space. The first obvious question arising in node comparison, is how to handle mismatch in the length of outgoing vectors for two nodes from different size graphs. This issue is addressed in Chapter 3; in this section we provide some some high-level ideas to give an overview of the graph clustering problem.

For a given large graph $\mathcal{G}_X(\mathcal{V}_x, \mathcal{E}_x, \mathbf{X})$ with n nodes, and a small graph $\mathcal{G}_Y(\mathcal{V}_y, \mathcal{E}_y, \mathbf{Y})$ with m nodes (assume $2 \leq m < n$), suppose we want to use $\mathcal{G}_Y$ to represent $\mathcal{G}_X$. Then each node in $\mathcal{G}_X$ is represented by a node in $\mathcal{G}_Y$; we define a partition function $\phi : \mathcal{V}_x \to \mathcal{V}_y$, equivalently an aggregation matrix $\mathbf{\Phi} \in \{0,1\}^{n \times m}$, to describe this relationship. Suppose there is an abstract distance function d°,

$$d^\circ : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+, \tag{2.14}$$

which takes an outgoing vector from $\mathcal{G}_X$ and the other from $\mathcal{G}_Y$, and computes a "distance" between them. We can then express the overall "error" caused from using nodes in $\mathcal{G}_Y$ to represent nodes in $\mathcal{G}_X$, where the nodes' assignment is specified by ϕ (or $\mathbf{\Phi}$), by a distortion function (2.4), given by

$$\rho^\circ_\Phi(\mathcal{G}_X, \mathcal{G}_Y) = \sum_{i=1}^{n} \mu_i \, d^\circ(\mathbf{x}(i), \mathbf{y}(\phi(i))), \tag{2.15}$$

where μ_i represents the relative weight of ith node. Since there are a total number of $\binom{n-1}{m-1} m!$ ways to assign n nodes in $\mathcal{G}_X$ to m nodes in $\mathcal{G}_Y$, it is natural to define the graph dissimilarity as the minimum error over *all node assignments* ϕ. That is, we define the *overall representation error* (or dissimilarity) by the smallest Φ-dissimilarity,

$$\nu(\mathcal{G}_X, \mathcal{G}_Y) = \min_{\Phi} \rho^\circ_\Phi(\mathcal{G}_X, \mathcal{G}_Y). \tag{2.16}$$

Therefore, the objective of finding an optimal representative graph can be posed as a clustering problem in an optimal resource allocation formulation with the decision variable $\mathcal{G}_Y$,

$$\min_{\mathcal{G}_Y} \nu(\mathcal{G}_X, \mathcal{G}_Y). \tag{2.17}$$

The precise development and interpretation are provided in detail in Chapter 3.

2.3.1 Soft Clustering via Maximum Entropy Principle

A challenge that is independent of the selection of d° arises from the integer constraint $\mathbf{\Phi} \in \{0,1\}^{m \times n}$, which incurs the well-known combinatorial computational complexity in solving the problem (2.17). To address this problem, we consider soft clustering (with $\tilde{\mathbf{\Phi}} \in [0,1]^{m \times n}$) and

apply the MEP as a realization of (2.8). In particular, let $\tilde{\nu}(\mathcal{G}_X, \mathcal{G}_Y)$ be the a *soft* dissimilarity function, then the *soft clustering problem* is given by:

$$\begin{aligned} \min_{\mathcal{G}_Y} \tilde{\nu}(\mathcal{G}_X, \mathcal{G}_Y) &= \min_{\mathcal{G}_Y} \min_{\tilde{\Phi}} \tilde{\rho}^{\circ}_{\tilde{\Phi}}(\mathcal{G}_X, \mathcal{G}_Y) \\ \text{where} \quad & \begin{cases} \tilde{\rho}^{\circ}_{\tilde{\mathbf{\Phi}}}(\mathcal{G}_X, \mathcal{G}_Y) = \sum_{i=1}^{n} \mu_i \sum_{j=1}^{m} \tilde{\Phi}_{ij} d^{\circ}(\mathbf{x}(i), \mathbf{y}(j)), \\ \tilde{\mathbf{\Phi}} \in [0,1]^{n \times m}, \tilde{\mathbf{\Phi}} \mathbf{1}_n = \mathbf{1}_n. \end{cases} \end{aligned} \tag{2.18}$$

in which $\tilde{\Phi}_{ij}$ indicates the association weight from the i^{th} node to the j^{th} supernode. After relaxation, the soft version of the overall dissimilarity and the $\mathbf{\Phi}$-dissimilarity become $\tilde{\nu}(\mathcal{G}_X, \mathcal{G}_Y)$ and $\tilde{\rho}^{\circ}_{\tilde{\mathbf{\Phi}}}(\mathcal{G}_X, \mathcal{G}_Y)$ in (2.18).

The solution follows the annealing procedure controlled by increasing values of β. As a result, both $\tilde{\mathbf{\Phi}}$ and weighing matrix $\mathbf{Y}$ can be calculated at any β, which yields a sequence of smaller graphs $\{\mathcal{G}_Y(\mathbf{Y})\}_\beta$, each being local minima of $\tilde{\nu}(\mathcal{G}_X, \mathcal{G}_Y)$, with dimensions less than or equal to n. A desired aggregation matrix $\mathbf{\Phi}$ satisfying (2.1), and an edge weight matrix $\mathbf{Y}$ of $\mathcal{G}_Y$ are obtained at the end when β is sufficiently large. A detailed description and implementation can be found in Chapter 4.

Summary of This Chapter

Our main achievements in this chapter are summarized as follows, we have

1. introduced the clustering problem in a resource allocation setting, and a combinatorial optimization formulation, and reviewed the DA algorithm based on the MEP;

2. interpreted the graph clustering problem as a resource allocation problem.

Chapter 3

A Dissimilarity Measure for Comparing Graphs

In the formulation of graph clustering problems, a small representative graph is obtained by identifying and aggregating similar nodes into supernodes, and then determining the resulting connections amongst these supernodes. This approach can be interpreted from the resource allocation or data compression perspective, where each supernode is a *resource* that is representative of a set of *close* nodes. Nodes that have *similar* connectivities to the rest of the nodes in the graph, which are captured in the edge-weight matrices, are considered *close*. In this chapter we develop a notion of dissimilarity between nodes of a graph, and extend it to characterize distortion between graphs. One of the main challenges in the context of graph aggregation is that we need a dissimilarity notion between graphs that are of different sizes, that is the edge-weight matrices are of different dimensions. Another challenge stems from the combinatorial number of ways in which the nodes of the original graph can be associated with the supernodes; therefore any development of *useful* dissimilarity measure should be computationally viable. As such, we introduce a dissimilarity measure $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ which characterizes the overall representation error (distortion) of using a smaller graph $\mathcal{G}_Y$ to represent a large graph $\mathcal{G}_X$.

In this chapter, we propose a dissimilarity measure $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ for directed weighted graphs (which thus also applies to undirected, or unweighted graphs). In other words, we will provide a precise expression for equations (2.14) to (2.16).

3.1 Comparing Graphs of the Same Dimensions

As an illustrative starting point, we consider the comparison of two graphs with the same number of nodes. In graph theory, a popular *distance measure* between two *connected undirected unweighted* graphs $\mathcal{G}_X(\mathcal{V}_x, \mathcal{E}_x, \mathbf{X})$ and $\mathcal{G}_Y(\mathcal{V}_y, \mathcal{E}_y, \mathbf{Y})$, when $|\mathcal{V}_x| = |\mathcal{V}_y| = n$, is defined through *graph match-*

ing [8], which provides a one-to-one assignment between two node sets of the same cardinality. Specifically, every *bijective* (relabeling) mapping $\psi : \mathcal{V}_x \to \mathcal{V}_y$ defines a ψ-*distance*, given by

$$d_\psi(\mathcal{G}_X, \mathcal{G}_Y) = \sum_{i,j \in \mathcal{V}_x} |d_{\mathcal{G}_X}(i,j) - d_{\mathcal{G}_Y}(\psi(i), \psi(j))|, \tag{3.1}$$

where $d_{\mathcal{G}_X}(i,j)$ and $d_{\mathcal{G}_Y}(\psi(i), \psi(j))$ are the lengths of the *shortest paths* in $\mathcal{G}_X$ and $\mathcal{G}_Y$ that connect the i^{th} to the j^{th} nodes, and the $\psi(i)^{th}$ to the $\psi(j)^{th}$ nodes. Then the graph dissimilarity (or distance) is defined as the minimum $\psi-$distance over all bijections ψ. This provides a motivation for our definition of the similarity measure for directed weighted graphs with different dimensions.

Taking $d_{\mathcal{G}_X}(i,j)$ as the shortest path length is suitable when $\mathcal{G}_X$ is undirected and unweighted. For *directed weighted* graphs, the edge weight specifies the strength of the connection, so we redefine $d_{\mathcal{G}_X}(i,j)$ to be the edge weight from the i^{th} node to the j^{th} node in $\mathcal{G}_X$, i.e.,

$$d_{\mathcal{G}_X}(i,j) := X_{ij},$$

which reflects the *functional influence*. Note that from the definition of the weight matrix, $\mathbf{X}$, the functional influence is 0 if and only if the i^{th} node and the j^{th} node are not directly connected by an edge. Further, we allow for different node weights, which can be interpreted as the "importance" of a node, or may be chosen to encode prior information about the nodes. Thus the ψ-*distance* in (3.1) is extended to

$$d_\psi(\mathcal{G}_X, \mathcal{G}_Y) = \sum_{i,j \in \mathcal{V}_x} \mu_i |X_{ij} - Y_{\psi(i)\psi(j)}| \tag{3.2}$$

to accommodate for directed weighted graphs with node weights $\{\mu_i\}$. Note that every bijective mapping ψ defines an *invertible* permutation matrix $\mathbf{\Psi} \in \{0,1\}^{n \times n}$ as in (2.1), with $\mathbf{\Psi}^{-1} = \mathbf{\Psi}^T$. We introduce a $z-$*transformation* of the weight matrix $\mathbf{Y}$ given by an n-by-n $\mathbf{Z}$ matrix satisfying

$$\mathbf{Z} = \mathbf{Y}\mathbf{\Psi}^T \Leftrightarrow \mathbf{Y} = \mathbf{Z}\mathbf{\Psi}. \tag{3.3}$$

So, row vectors of $\mathbf{Z}$ and $\mathbf{X}$ are of the same length, moreover, $Y_{ij} = \sum_{k=1}^n Z_{ik}\Psi_{kj} = \mathbf{z}(i)\mathbf{e}_{\psi^{-1}(j)} =$

$Z_{i\psi^{-1}(j)}$ for all i, j. Equivalently, $Y_{i\psi(k)} = Z_{ik}$, for all i, k. We can then give a realization of the abstract d° function in (2.14) by adopting the $z-$transformation, and rewriting the ψ-distance (3.2) as

$$d_\psi(\mathcal{G}_X, \mathcal{G}_Y) = \sum_{i=1}^{n} \mu_i \|\mathbf{x}(i) - \mathbf{z}(\psi(i))\|_1. \tag{3.4}$$

Note that this expression only requires the number of columns of $\mathbf{X}$ and $\mathbf{Z}$ matrices to be the same (i.e., there are no restrictions on the number of rows).

The distance between two graphs $\mathcal{G}_X$ and $\mathcal{G}_Y$ is then defined as the *minimum ψ-distance* achieved by the optimal node matching, that is,

$$\nu(\mathcal{G}_X, \mathcal{G}_Y) \stackrel{\triangle}{=} \min_{\psi} d_\psi(\mathcal{G}_X, \mathcal{G}_Y). \tag{3.5}$$

As shown in [8], for *undirected*, *unweighted* graphs with equally weighted nodes, the definition for $\nu(\cdot, \cdot)$ with $d_{\mathcal{G}_X}(i, j)$ chosen by (3.1), is symmetric and satisfies the triangle inequality. Further, $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ is always positive unless $\mathcal{G}_X \simeq \mathcal{G}_Y$, where $\nu(\mathcal{G}_X, \mathcal{G}_Y) = 0$. Therefore, $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ defines a valid pseudo-metric. It is worth noting that, computing this distance requires extensive computation. For example, to compute the distance between two graphs of size n requires minimizing over a set of $n!$ bijective mappings $\psi : \mathcal{V}_x \to \mathcal{V}_y$.

The distance metric defined in (3.4), along with the z-transformation (3.3), provide a basis for the generalization to graphs with different numbers of nodes.

3.2 Comparing Graphs of Different Dimensions

The definition based on graph node matching (3.5) is only applicable to graphs with the same numbers of nodes. When two graphs have different dimensions, we propose a further generalization. Namely, one node in the smaller graph may be matched to a set of nodes in the larger graph, where overlap between different groups is not permitted (that is the nodes of smaller graph partition the nodes of the larger graph). As noted in Section 2.3, the dimensionality difference needs to be addressed, or we need to give vector distance function d° in (2.14) a proper definition. This goal

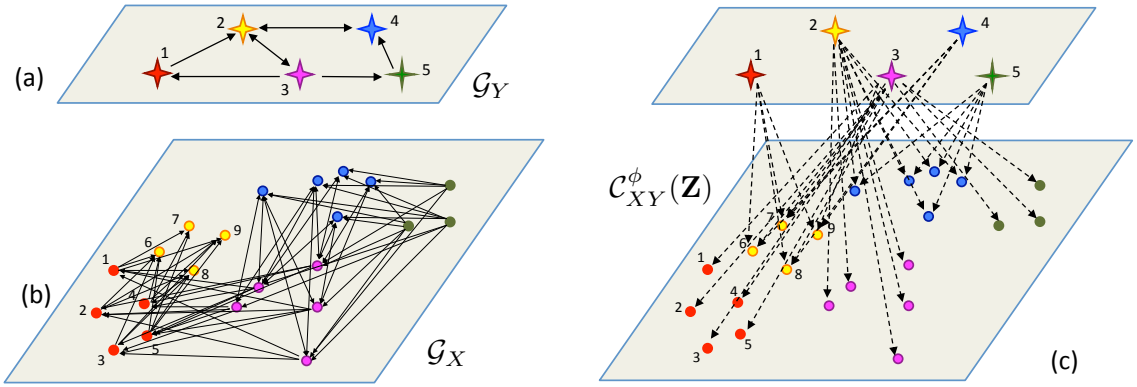


Figure 3.1: The construction of one composite graph $\mathcal{C}_{XY}^{\phi}(\mathbf{Z}) \in \boldsymbol{\mathcal{C}}_{XY}$ (shown in (c)) from two graphs, $\mathcal{G}_X(\mathcal{V}_x, \mathcal{E}_x, \mathbf{X})$ (shown in (b)) and $\mathcal{G}_Y(\mathcal{V}_y, \mathcal{E}_y, \mathbf{Y})$ (shown in (a)). $\mathcal{G}_X$ has 22 nodes, and we assume all its edges have unit weight; for example, the outgoing vector of the first node is $\mathbf{x}(1) = [0, 0, 0, 0, 0, 1, 1, 1, 1, 0, \ldots, 0]$. A representative graph $\mathcal{G}_Y$ has 5 supernodes (denoted by stars), suppose its outgoing vectors have $\mathbf{y}(1) = [0, 4, 0, 0, 0]$ and $\mathbf{y}(2) = [0, 0, 5, 5, 0]$. By Definition 1, $\mathcal{C}_{XY}^{\phi}$ contains all nodes (stars and dots) from both $\mathcal{G}_X$ and $\mathcal{G}_Y$, with edges that initiate from $\mathcal{V}_y$ and terminate at $\mathcal{V}_x$ (dashed arrows). In this example, the partition function ϕ from the mapping from $\mathcal{V}_x$ to $\mathcal{V}_y$ is shown in colors, for example, $\phi^{-1}(1) = \{1, 2, 3, 4, 5\}, \phi^{-1}(2) = \{6, 7, 8, 9\}$, are indicated by red and yellow nodes/stars. The edge weight matrix of $\mathcal{C}_{XY}(\mathbf{Z})$ and the $\mathbf{Y}$ matrix satisfies the z-transformation (3.6), for example, $Z_{16} + Z_{17} + Z_{18} + Z_{19} = 4 = Y_{12}$ and $Z_{11} + Z_{12} + Z_{13} + Z_{14} + Z_{15} = 0 = Y_{11}$. In this case, either $\hat{\mathbf{z}}(1) = [0, \ldots, 0, 1, 1, 1, 1]$ or $\check{\mathbf{z}}(1) = [0, \ldots, 0, 0.5, 1.5, 1, 1]$ is a valid outgoing vector for the red star in $\mathcal{C}_{XY}(\mathbf{Z})$, but setting $\mathbf{z}(1) = \hat{\mathbf{z}}(1)$ yields a smaller value $\rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y)$.

is accomplished by introducing a set of *composite graphs* derived from the two graphs of interest, which defines a set of *extended z-transformations.* We further generalize the $\psi-$distance in (3.4) by allowing a variety of vector distances beyond the l_1 norm.

Definition 1 (Composite graph set [71]) *Given two graphs, $\mathcal{G}_X(\mathcal{V}_x, \mathcal{E}_x, \mathbf{X})$ with n nodes and $\mathcal{G}_Y(\mathcal{V}_y, \mathcal{E}_y, \mathbf{Y})$ with m nodes ($m \leq n$), the* composite graph set *associated with $\mathcal{G}_X$ and $\mathcal{G}_Y$ is defined as $\boldsymbol{\mathcal{C}}_{XY} \triangleq \{\mathcal{C}_{XY}^{\phi}(\mathcal{V}_z, \mathcal{E}_z, \mathbf{Z})\}$, such that each* composite graph $\mathcal{C}_{XY}^{\phi}(\mathcal{V}_z, \mathcal{E}_z, \mathbf{Z}) \in \boldsymbol{\mathcal{C}}_{XY}$ *satisfies the following conditions:*

(i) The node set $\mathcal{V}_z = \mathcal{V}_x \cup \mathcal{V}_y$ is the union of all nodes in $\mathcal{G}_X$ and $\mathcal{G}_Y$. Moreover, for notational simplicity, $\mathcal{V}_z$ is indexed in an order such that the first m nodes are from $\mathcal{G}_Y$ and the remaining n nodes are from $\mathcal{G}_X$.

(ii) The edges in $\mathcal{C}_{XY}^{\phi}(\mathbf{Z})$ start from nodes in $\mathcal{G}_Y$ and end at nodes in $\mathcal{G}_X$, or equivalently, $\mathcal{E}_z \subset \mathcal{V}_y \times \mathcal{V}_x$. Therefore, although every $\mathcal{C}_{XY}^{\phi}(\mathbf{Z})$ has $n + m$ nodes, we represent its edge weight

matrix by $\mathbf{Z} = \left[\mathbf{z}(1)^T\, \mathbf{z}(2)^T \cdots \mathbf{z}(m)^T\right]^T \in \mathbb{R}^{m\times n}$*, with the outgoing vector* $\mathbf{z}(j) \in \mathbb{R}^{1\times n}$*.*

(iii) The partition function $\phi : \mathcal{V}_x \to \mathcal{V}_y$ *provides an edge weight aggregation relation between* $\mathcal{C}^{\phi}_{XY}(\mathbf{Z})$ *and* $\mathcal{G}_Y$*:*

$$Y_{jk} = \sum_{i\in\phi^{-1}(k)} Z_{ji}, \qquad j,k = 1,\cdots,m. \tag{3.6}$$

Using the aggregation matrix $\mathbf{\Phi}$ *defined in* (2.1)*,* (3.6) *can be compactly written as* $\mathbf{Y} = \mathbf{Z}\mathbf{\Phi}$*. This extends the z-transformation in* (3.3) *by substituting the permutation matrix* $\mathbf{\Psi}$ *by an aggregation matrix* $\mathbf{\Phi}$*, and thus allows for comparison of graphs with different sizes.*

By construction, the outgoing vectors of any $\mathcal{C}^{\phi}_{XY}(\mathbf{Z}) \in \boldsymbol{\mathcal{C}}_{XY}$ are of the same length, n, as the outgoing vectors of $\mathcal{G}_X$ (see Figure 3.1 for illustration). Now we are at the stage to formally introduce the dissimilarity measure. Let

$$\begin{aligned} d^{\circ}(\mathbf{x}(i), \mathbf{y}(\phi(i))) &\stackrel{\triangle}{=} d(\mathbf{x}(i), \mathbf{z}(\phi(i))), \\ \text{where} \qquad & \mathbf{Y} = \mathbf{Z}\mathbf{\Phi}. \end{aligned} \tag{3.7}$$

Then we extend the concept of *ψ-distance* in (3.4) to allow node comparison for unequal-sized graphs, and this yields a *ϕ-dissimilarity* between $\mathcal{G}_X$ and $\mathcal{G}_Y$,

$$\begin{aligned} \rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y) &= d_{\phi}(\mathcal{G}_X, \mathcal{C}^{\phi}_{XY}(\mathcal{V}_z, \mathcal{E}_z, \mathbf{Z})) \\ &= \sum_{i=1}^{n} \mu_i\, d(\mathbf{x}(i), \mathbf{z}(\phi(i))), \end{aligned} \tag{3.8}$$

where $\{\mu_i\}$ are node weights, and $d(\cdot,\cdot)$ is a generic vector distance function; reasonable choices include any p-norm for any $p \in [1,\infty]$, or the K-L divergence function. The *dissimilarity* between $\mathcal{G}_X$ and $\mathcal{G}_Y$ is defined as the minimum achievable *ϕ-dissimilarity* over all partitions ϕ and weighting matrices $\mathbf{Z} \in \mathbb{R}^{m\times n}_{+}$. Equivalently, this is the minimum over all $\mathcal{C}^{\phi}_{XY}$ in the composite graph set $\boldsymbol{\mathcal{C}}_{XY}$, given by:

$$\nu(\mathcal{G}_X, \mathcal{G}_Y) \stackrel{\triangle}{=} \min_{\mathcal{C}^{\phi}_{XY}\in\boldsymbol{\mathcal{C}}_{XY}} \rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y). \tag{3.9}$$

Similar to the findings discussed following (3.5), computing $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ requires the solution of a combinatorial optimization problem, which is NP-hard [1].

3.3 Properties of the Distance Measure

In this section we discuss in detail conditions under which $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ is a pseudo-metric, namely when the properties of non-negativity, symmetry and the triangle inequality hold. We begin with some useful definitions.

Definition 2 (Graph dilation) *For a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ with node weights $\{\mu_k\}$, a* dilated graph *$\bar{\mathcal{G}}(\bar{\mathcal{V}}, \bar{\mathcal{E}}, \bar{\mathbf{W}})$ of $\mathcal{G}$ is obtained by* splitting *one node, $v_k \in \mathcal{V}$, into several nodes, $\bar{v}_{k_1}, \bar{v}_{k_2}, \ldots, \bar{v}_{k_{n_k}}$, while maintaining all given edge connections. Specifically, after splitting, the (nonnegative) node weights satisfy $\sum_{i=1}^{n_k} \bar{\mu}_{k_i} = \mu_k$, and $\bar{\mu}_j = \mu_j$, for $j \neq k$; and edge weights satisfy $\bar{W}_{k_i,j} = W_{kj}$, $\bar{W}_{j,k_i} = \frac{\bar{\mu}_{k_i}}{\mu_k} W_{jk}$ and $\bar{W}_{k_i k_l} = 0$, for $1 \leq i \neq l \leq n_k$, and $\bar{W}_{ls} = W_{ls}$ for all $l, s \neq k$. Note that $\bar{\mathbf{W}}$ will contain n_k identical rows.*

Definition 3 (Graph projection) *If $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ contains repeated nodes, $\{v_{k_i}\}$, as indicated by identical outgoing vectors (or identical rows in the weight matrix $\mathbf{W}$), let the node weights be $\{v_{k_i}\}$, then a* projection graph *$\underline{\mathcal{G}}(\underline{\mathcal{V}}, \underline{\mathcal{E}}, \underline{\mathbf{W}})$ is obtained by* collapsing *or* merging *repeated nodes in $\mathcal{G}$, giving $\underline{v}_k$. Specifically, we collapse the duplicated rows (maintaining one copy) in $\mathbf{W}$ and get $\mathbf{W}_1$. The collapse step defines a node partition $\phi : \mathcal{V} \to \underline{\mathcal{V}}$. The weight matrix for the projection graph is given by $\underline{\mathbf{W}} = \mathbf{W}_1 \mathbf{\Phi}$, where $\mathbf{\Phi}$ is the associated aggregation matrix. The node weights are also aggregated as $\underline{\mu}_k = \sum_i \mu_{k_i}$.*

We now develop some useful properties of the graph dissimilarity (or distance) function defined in (3.9), and show that $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ is a pseudo-metric when choosing the vector distance $d(\cdot, \cdot)$ in (3.8) as a p-norm.

Proposition 1 *The dissimilarity function $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ given by* (3.9) *is nonnegative for any two graphs $\mathcal{G}_X$ and $\mathcal{G}_Y$.*

This is a straightforward result of the definitions (3.8) and (3.9).

Proposition 2 *The dissimilarity between isomorphic graphs is zero, that is, if* $\mathcal{G}_X \simeq \mathcal{G}_Y$*, then* $\nu(\mathcal{G}_X, \mathcal{G}_Y) = 0$.

Proof: If $\mathcal{G}_X \simeq \mathcal{G}_Y$, by definition $|\mathcal{V}_x| = |\mathcal{V}_y| = n$, and there exists a bijective mapping $\check{\psi} : \mathcal{V}_x \to \mathcal{V}_y$, such that $(i,j) \in \mathcal{E}_x$ if and only if $(\check{\psi}(i), \check{\psi}(j)) \in \mathcal{E}_y$, and $X_{ij} = Y_{\check{\psi}(i)\check{\psi}(j)}$. Let $\check{\mathbf{\Psi}}$ be the permutation matrix induced from $\check{\psi}$ according to mapping (2.1). We have $\check{\mathbf{\Psi}}^T\check{\mathbf{\Psi}} = \mathbf{I}_n$. Select the composite graph $\mathcal{C}_{XY}^{\check{\psi}}(\check{\mathbf{Z}})$ specified by $\check{\psi}$ and $\check{\mathbf{Z}} \stackrel{\triangle}{=} \mathbf{Y}\check{\mathbf{\Psi}}^T$. Then conditions (i), (ii) and the edge weight constraints (3.6) in (iii) of *Definition 1* are satisfied, so $\rho_{\check{\psi},\check{\mathbf{Z}}}(\mathcal{G}_X, \mathcal{G}_Y) = d_{\check{\psi}}(\mathcal{G}_X, \mathcal{C}_{XY}^{\check{\psi}}(\mathbf{Z})) = \sum_i \mu_i d(\mathbf{x}(i), \check{\mathbf{z}}(\check{\psi}(i)))$.To verify the equality $\nu(\mathcal{G}_X, \mathcal{G}_Y) = 0$, we note that $\mathbf{y}(i) = \check{\mathbf{z}}(i)\check{\mathbf{\Psi}} = [\check{Z}_{i1}, \check{Z}_{i2}, \ldots, \check{Z}_{in}][\mathbf{e}_{\check{\psi}^{-1}(1)}, \mathbf{e}_{\check{\psi}^{-1}(2)}, \ldots, \mathbf{e}_{\check{\psi}^{-1}(n)}] = [\check{Z}_{i\check{\psi}^{-1}(1)}, \check{Z}_{i\check{\psi}^{-1}(2)}, \ldots, \check{Z}_{i\check{\psi}^{-1}(n)}]$
$= [Y_{i1}, Y_{i2}, \ldots, Y_{in}]$, which implies $\check{Z}_{i\check{\psi}^{-1}(k)} = Y_{ik}$ for all i, k, or $X_{ij} = Y_{\check{\psi}(i)\check{\psi}(j)} = \check{Z}_{\check{\psi}(i)j}$, for all i, j. Therefore, $\mathbf{x}(i) = \check{\mathbf{z}}(\check{\psi}(i))$ for all i, and hence $0 \le \nu(\mathcal{G}_X, \mathcal{G}_Y) \le \rho_{\check{\psi},\check{\mathbf{Z}}}(\mathcal{G}_X, \mathcal{G}_Y) = 0$. □

The following example demonstrates *Proposition 2*.

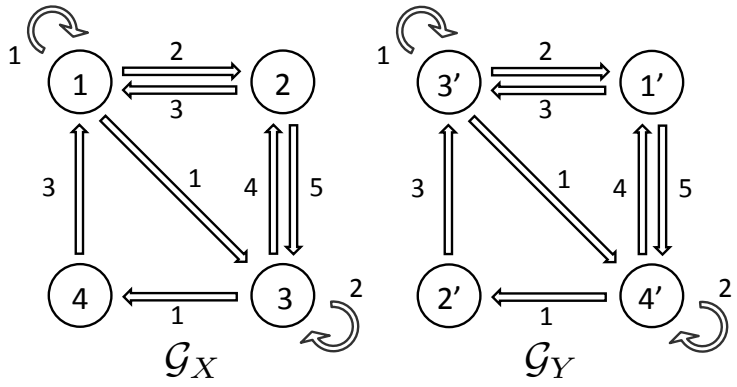


Figure 3.2: $\nu(\mathcal{G}_X, \mathcal{G}_Y) = 0$ if $\mathcal{G}_Y \simeq \mathcal{G}_X$.

Example Consider graphs $\mathcal{G}_X$ and $\mathcal{G}_Y$ depicted in Figure 3.2, with $\mathcal{V}_x = \{1,2,3,4\}$, $\mathcal{V}_y = \{1',2',3',4'\}$, and suppose all nodes haves the same weight, with edge weight matrices given by

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 3 & 0 & 5 & 0 \\ 0 & 4 & 2 & 1 \\ 3 & 0 & 0 & 0 \end{bmatrix}, \text{ and } \mathbf{Y} = \begin{bmatrix} 0 & 0 & 3 & 5 \\ 0 & 0 & 3 & 0 \\ 2 & 0 & 1 & 1 \\ 4 & 1 & 0 & 2 \end{bmatrix}.$$

It is easy to see that $\mathcal{G}_X \simeq \mathcal{G}_Y$ through node relabeling: the permutation is given by $\check{\psi} : \mathcal{V}_x \mapsto \mathcal{V}_y$ as $\check{\psi}(1) = 3', \check{\psi}(2) = 1', \check{\psi}(3) = 4', \check{\psi}(4) = 2$. Therefore,

$$\check{\mathbf{\Psi}} = [\mathbf{e}_2, \mathbf{e}_4, \mathbf{e}_1, \mathbf{e}_3], \check{\mathbf{Z}} = \mathbf{Y}\check{\mathbf{\Psi}}^T = \begin{bmatrix} 3 & 0 & 5 & 0 \\ 3 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 4 & 2 & 1 \end{bmatrix}.$$

Then, $\rho_{\check{\psi},\check{\mathbf{Z}}}(\mathcal{G}_X, \mathcal{G}_Y) = \frac{1}{4}\{d(\mathbf{x}(1), \mathbf{z}(3)) + d(\mathbf{x}(2), \mathbf{z}(1)) + d(\mathbf{x}(3), \mathbf{z}(4)) + d(\mathbf{x}(4), \mathbf{z}(2))\} = 0$ $\geq \nu(\mathcal{G}_X, \mathcal{G}_Y) \geq 0$, for all vector distance measures d, and thus $\nu(\mathcal{G}_X, \mathcal{G}_Y) = 0$.

Proposition 3 *For a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}\ \mathbf{W})$, if $\bar{\mathcal{G}}(\bar{\mathcal{V}}, \bar{\mathcal{E}}, \bar{\mathbf{W}})$ and $\underline{\mathcal{G}}(\underline{\mathcal{V}}, \underline{\mathcal{E}}, \underline{\mathbf{W}})$ are a* dilation *and a* projection *of $\mathcal{G}$, respectively, then $\nu(\mathcal{G}, \bar{\mathcal{G}}) = \nu(\mathcal{G}, \underline{\mathcal{G}}) = 0$.*

Proof: We show $\nu(\mathcal{G}, \bar{\mathcal{G}}) = 0$ in the dilation case; the projection case is similar. Assume $|\mathcal{V}| = n$. Since multiple node repetitions can be viewed as a sequence of single-node-repetitions, it is enough to consider the single-node-repetition case. We assume $\bar{\mathcal{G}}$ is obtained from $\mathcal{G}$ by a single-node-repetition, say $v_i \to \{\bar{v}_{i_1}, \ldots, \bar{v}_{i_k}\}$, then $|\bar{\mathcal{V}}| = n + k - 1$. From the definition of dilation, the node weights satisfy the relation $\mu_i = \sum_{s=1}^{k} \bar{\mu}_{i_s}$ and $\mu_j = \bar{\mu}_j$ for $j \neq i$.

Now we show $\nu(\mathcal{G}, \bar{\mathcal{G}}) = 0$ by defining a composite graph $\mathcal{C}^{\phi}_{W\bar{W}}(\mathbf{Z})$ with a partition function $\phi : \bar{\mathcal{V}} \to \mathcal{V}$ and weighting matrix $\mathbf{Z} \in \mathbb{R}^{n \times (n+k-1)}$ that satisfies *Definition 1*, such that $\rho_{\phi,\mathbf{Z}}(\mathcal{G}, \bar{\mathcal{G}}) = 0$. Let $\phi^{-1}(v_j) = \bar{v}_j$, if $j \neq i$, and $\phi^{-1}(v_i) = \{\bar{v}_{i_1}, \ldots, \bar{v}_{i_k}\}$, and select the rows of $\mathbf{Z}$ as $\mathbf{z}(j) := \bar{\mathbf{w}}(j)$, for $j \neq i$, and $\mathbf{z}(i) := \bar{\mathbf{w}}(i_1)$ (note that the $\bar{\mathbf{w}}(i_s)$'s are identical rows for $1 \leq s \leq k$). With this choice, $\rho_{\phi,\mathbf{Z}}(\mathcal{G}, \bar{\mathcal{G}}) = \sum_{j \neq i} \bar{\mu}_j d(\bar{\mathbf{w}}(j), \mathbf{z}(j)) + \sum_{s=1}^{k} \bar{\mu}_{i_s} d(\bar{\mathbf{w}}(i_s), \mathbf{z}(i)) = 0$, and hence $\nu(\mathcal{G}, \bar{\mathcal{G}}) = 0$. □

The following example illustrates the dilation/projection proposition.

Example Consider graphs $\mathcal{G}_X$ and $\mathcal{G}_Y$ depicted in Figure 3.3, with $\mathcal{V}_x = \{1, 2\}$, $\mathcal{V}_y = \{1', 2', 3', 4'\}$. Let $\mu_1 = p$, $\mu_2 = 1 - p$ and the weighting matrices be

$$\mathbf{X} = \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix}, \text{ and } \mathbf{Y} = \begin{bmatrix} 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 \\ 0.5 & 1.5 & 1 & 0 \end{bmatrix}.$$

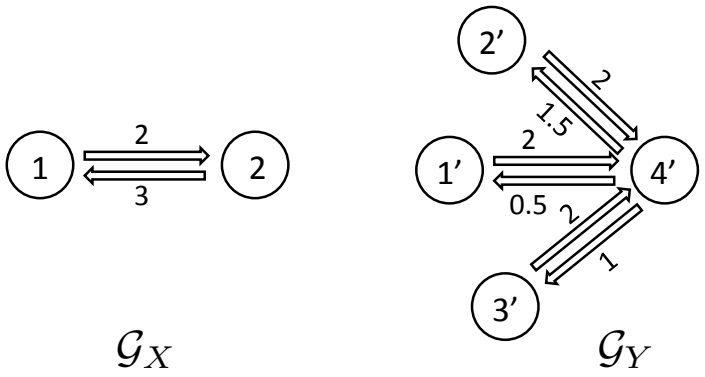


Figure 3.3: Case when $\mathcal{G}_Y$ is a dilation of $\mathcal{G}_X$.

Note that $\mathbf{Y}$ contains duplicated rows, which indicates $\{1', 2', 3'\}$ can be collapsed without inducing a positive dissimilarity. We choose $\phi : \{1', 2', 3', 4'\} \to \{1, 2\}$ as $\phi^{-1}(1) = \{1', 2', 3'\}$ and $\phi^{-1}(2) = \{4'\}$, then the corresponding $\mathbf{\Phi} = [\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_4] \in \{0, 1\}^{4\times 2}$. Now we construct $\mathbf{Z} \in \mathbb{R}^{2\times 4}$ by deleting duplicated rows in $\mathbf{Y}$, that is $\mathbf{z}(1) := \mathbf{y}(1), \mathbf{z}(2) := \mathbf{y}(2)$, so $\mathbf{Z\Phi} = \mathbf{Y}$ is satisfied, and $\mathcal{C}^{\phi}_{YX}(\mathbf{Z}) \in \boldsymbol{\mathcal{C}}_{YX}$. Thus, $\rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y) = \mu_{1'} d(\mathbf{y}(1), \mathbf{z}(1)) + \mu_{2'} d(\mathbf{y}(2), \mathbf{z}(1)) + \mu_{3'} d(\mathbf{y}(3), \mathbf{z}(1)) + \mu_{4'} d(\mathbf{y}(4), \mathbf{z}(2)) = \mu_1 d(\mathbf{y}(1), \mathbf{z}(1)) + \mu_2 d(\mathbf{y}(4), \mathbf{z}(2)) = 0$, and this gives $\nu(\mathcal{G}_X, \mathcal{G}_Y) = 0$. Alternatively, in this example $\mathcal{G}_X$ can be viewed as a projection of $\mathcal{G}_Y$.

Lemma 1 *For directed weighted graphs $\mathcal{G}_X(\mathcal{V}_x, \mathcal{E}_x, \mathbf{X})$ with n nodes and $\mathcal{G}_Y(\mathcal{V}_y, \mathcal{E}_y, \mathbf{Y})$ with m nodes, $\nu(\mathcal{G}_X, \mathcal{G}_Y) = 0$ if and only if $\mathcal{G}_X \simeq \mathcal{G}_{Y'}$ where $\mathcal{G}_{Y'}$ is either $\mathcal{G}_Y$ or a dilation/projection of $\mathcal{G}_Y$.*

Proof: The "if" part has been shown in *Proposition 2* and *Proposition 3*. For the "only if" part, assume that $n > m$. Let $\nu(\mathcal{G}_X, \mathcal{G}_Y) = \min_{\mathcal{C}^{\phi}_{XY} \in \boldsymbol{\mathcal{C}}_{XY}} \rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y) = 0$, then there exist $\check{\phi} : \mathcal{V}_x \to \mathcal{V}_y$ and $\check{\mathbf{Z}}$ that define a composite graph $\mathcal{C}^{\check{\phi}}_{XY}(\mathbf{Z})$ attaining $\rho_{\check{\phi},\check{\mathbf{Z}}}(\mathcal{G}_X, \mathcal{G}_Y) = \sum_{i=1}^{n} \mu_i d(\mathbf{x}(i), \check{\mathbf{z}}(\check{\phi}(i))) = 0$. Since $\mu_i \geq \delta > 0$, the zero dissimilarity is only achieved when $d(\mathbf{x}(i), \check{\mathbf{z}}(\check{\phi}(i))) = d(\mathbf{x}(i), \check{\mathbf{z}}(\check{\phi}(i))) = 0$ for all i. This implies $\check{\mathbf{z}}(k) = \mathbf{x}(j)$ for all $j \in \check{\phi}^{-1}(k)$, and $k = 1, \ldots, m$. In other words, all outgoing vectors $\mathbf{x}(i)$ are identical (and equal to $\check{\mathbf{z}}(\check{\phi}(i))$ within each cell of the partition). This can happen in two cases: (i) the set $\check{\phi}^{-1}(k)$ only contains a single element for all k, that is, all partitions are one-to-one, which implies $\mathcal{G}_X$ and $\mathcal{G}_Y$ are permutations of each other, thus $\mathcal{G}_X \simeq \mathcal{G}_Y$; (ii) some partitions consist of nodes with repeated outgoing vectors; this is case when the larger graph is a dilation of the smaller graph. In both cases, using the notation $\mathcal{G}_{Y'}$, we have $\mathcal{G}_{Y'} \simeq \mathcal{G}_X$. □

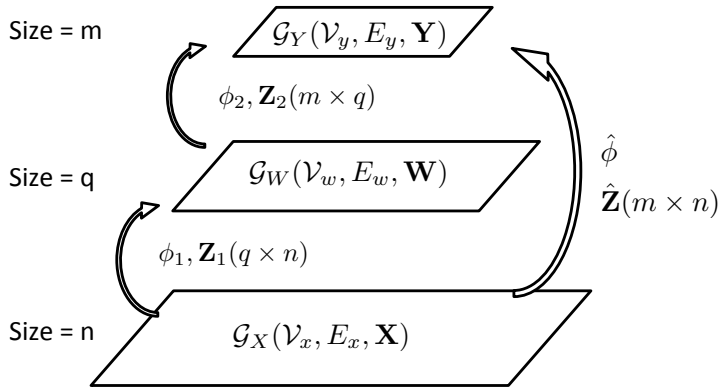


Figure 3.4: $\nu(\mathcal{G}_X, \mathcal{G}_Y) \leq \nu(\mathcal{G}_X, \mathcal{G}_W) + \nu(\mathcal{G}_W, \mathcal{G}_Y)$.

Lemma 2 *Let $\mathcal{G}_X(\mathcal{V}_x, \mathcal{E}_x, \mathbf{X})$, $\mathcal{G}_Y(\mathcal{V}_y, \mathcal{E}_y, \mathbf{Y})$ and $\mathcal{G}_W(\mathcal{V}_w, \mathcal{E}_w, \mathbf{W})$ be three weighted directed graphs, then the following inequality holds for the dissimilarity function defined in* (3.9)*, with $d(\cdot, \cdot)$ being the p-norm for $1 \leq p \leq \infty$:*

$$\nu(\mathcal{G}_X, \mathcal{G}_Y) \leq \nu(\mathcal{G}_X, \mathcal{G}_W) + \nu(\mathcal{G}_W, \mathcal{G}_Y). \tag{3.10}$$

Proof: Assume $|\mathcal{V}_X| = n$, $|\mathcal{V}_W| = q$, $|\mathcal{V}_Y| = m$ and $n \geq q \geq m$, and let μ_i^X, μ_i^Y, and μ_i^W be the corresponding node weights. Let $\mathcal{C}_{XW}^{\phi_1}(\mathbf{Z}_1) = \arg\min_{\mathcal{C}_{XW}^{\phi} \in \boldsymbol{\mathcal{C}}_{XW}} \rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_W)$ and $\mathcal{C}_{WY}^{\phi_2}(\mathbf{Z}_2) = \arg\min_{\mathcal{C}_{WY}^{\phi} \in \boldsymbol{\mathcal{C}}_{WY}} \rho_{\phi,\mathbf{Z}}(\mathcal{G}_W, \mathcal{G}_Y)$, that is, $\nu(\mathcal{G}_X, \mathcal{G}_W) = \rho_{\phi_1,\mathbf{Z}_1}(\mathcal{G}_X, \mathcal{G}_W)$ and $\nu(\mathcal{G}_W, \mathcal{G}_Y) = \rho_{\phi_2,\mathbf{Z}_2}(\mathcal{G}_W, \mathcal{G}_Y)$.

We define a partition function $\hat{\phi} : \mathcal{V}_x \to \mathcal{V}_z$ by $\hat{\phi}(i) = \phi_2(\phi_1(i))$ for $1 \leq i \leq n$, and a weight matrix $\hat{\mathbf{Z}} \in \mathbb{R}^{m \times n}$ as

$$\hat{Z}_{kl} = \gamma_{kl}[\mathbf{Z}_1]_{il} \tag{3.11}$$

for $1 \leq k \leq m, 1 \leq l \leq n$ and $\phi_2(i) = k$, $\gamma_{kl} \stackrel{\triangle}{=} \frac{[\mathbf{Z}_2]_{k\phi_1(l)}}{W_{i\phi_1(l)}}$, ($\gamma_{kl} \stackrel{\triangle}{=} 0$ when both numerator and denominator are 0). The pair $\hat{\phi}$ and $\hat{\mathbf{Z}}$ satisfies the z-transformation (3.6), since $\sum_{l \in \hat{\phi}^{-1}(j)} \hat{Z}_{kl} = \sum_{r \in \phi_2^{-1}(j)} \left[\sum_{l \in \phi_1^{-1}(r)} \hat{Z}_{kl} \right] = \sum_{r \in \phi_2^{-1}(j)} [\mathbf{Z}_2]_{kr} = Y_{kj}$, and they define a valid (although not neces-

sarily optimal) compositie graph $\mathcal{C}_{XY}^{\hat{\phi}}(\hat{\mathbf{Z}})$. Then we have

$$\begin{aligned} & \nu(\mathcal{G}_X, \mathcal{G}_Y) \leq \rho_{\hat{\phi},\hat{\mathbf{Z}}}(\mathcal{G}_X, \mathcal{G}_Y) \\ = & \sum_{i=1}^{n} \mu_i^X d(\mathbf{x}(i), \hat{\mathbf{z}}(\hat{\phi}(i))) \\ \overset{(a)}{\leq} & \sum_{i=1}^{n} \mu_i^X \underbrace{\left[d(\mathbf{x}(i), \mathbf{z}_1(\phi_1(i))) + d(\mathbf{z}_1(\phi_1(i)), \hat{\mathbf{z}}(\hat{\phi}(i))) \right]}_{(I)}. \end{aligned} \tag{3.12}$$

in which we apply the triangle inequality for p-norms in (a). Also by assumption,

$$\begin{aligned} & \nu(\mathcal{G}_X, \mathcal{G}_W) + \nu(\mathcal{G}_W, \mathcal{G}_Y) \\ = & \sum_{i=1}^{n} \mu_i^X d(\mathbf{x}(i), \mathbf{z}_1(\phi_1(i))) + \sum_{j=1}^{q} \mu_j^W d(\mathbf{w}(j), \mathbf{z}_2(\phi_2(j))) \\ \overset{(b)}{=} & \sum_{i=1}^{n} \mu_i^X d(\mathbf{x}(i), \mathbf{z}_1(\phi_1(i))) \\ & + \sum_{j=1}^{q} \left[\sum_{i \in \phi_1^{-1}(j)} \mu_i^X d(\mathbf{w}(\phi_1(i)), \mathbf{z}_2(\phi_2\phi_1(i))) \right] \\ = & \sum_{i=1}^{n} \underbrace{\mu_i^X \left[d(\mathbf{x}(i), \mathbf{z}_1(\phi_1(i))) + d(\mathbf{w}(\phi_1(i)), \mathbf{z}_2(\phi_2\phi_1(i))) \right]}_{(II)}. \end{aligned} \tag{3.13}$$

Equality (b) follows from the fact that the summation over the index j (representing the nodes in $\mathcal{G}_W$) can be replaced by the summation over the index i (representing the nodes in $\mathcal{G}_X$) by using the partition function $\phi_1 : \mathcal{V}_x \to \mathcal{V}_w$.

Let $d(\cdot, \cdot)$ be the p-norm for $1 \leq p < \infty$. We also denote $j := \phi_1(i)$, $k = \phi_2(j)$ to simplify the indices. To prove (3.10), we need to show that the summation (I) is less than summation (II) and therefore it is sufficient to show the individual terms in (I) are less than the corresponding terms in (II). More specifically, it is sufficient to show the following inequality

$$\|\mathbf{z}_1(j) - \hat{\mathbf{z}}(k)\|_p^p \leq \|\mathbf{w}(j) - \mathbf{z}_2(k)\|_p^p \tag{3.14}$$

holds for each $1 \leq j \leq q$. Then an upper bound of $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ as given in (3.12) will be smaller than (3.13), which is the right hand side of (3.10). For a finite p, the right hand side of (3.14) is

given by

$$\begin{aligned} RHS &= \sum_{h=1}^{q} |W_{jh} - [\mathbf{Z}_2]_{kh}|^p \\ &\overset{(c)}{=} \sum_{h=1}^{q} \left| \sum_{l \in \phi_1^{-1}(h)} ([\mathbf{Z}_1]_{jl} - \hat{Z}_{kl}) \right|^p \\ &= \sum_{h=1}^{q} \left| \sum_{l \in \phi_1^{-1}(h)} (1 - \gamma_{kl})([\mathbf{Z}_1]_{jl}) \right|^p \\ &\overset{(d)}{\geq} \sum_{h=1}^{q} \left[\sum_{l \in \phi_1^{-1}(h)} |(1 - \gamma_{kl})[\mathbf{Z}_1]_{jl}|^p \right] \\ &= \sum_{h=1}^{q} \left[\sum_{l \in \phi_1^{-1}(h)} |[\mathbf{Z}_1]_{jl} - \hat{Z}_{kl}|^p \right] = LHS \end{aligned}$$

where (c) uses the definition for $\hat{\mathbf{Z}}$ in (3.11), and (d) holds since for all $l \in \phi_1^{-1}(h)$, γ_{kl} is a constant and $\mathbf{Z}_1 \geq 0$, and thus all terms $(1 - \gamma_{k,l})([\mathbf{Z}_1]_{jl})$ are of the same sign. The inequality follows since $|a + b|^p \geq |a|^p + |b|^p$, for $a, b \in \mathbb{R}, ab > 0$ and $1 \leq p < \infty$.

When $p = \infty$, the right hand side of (3.14) is given by

$$\begin{aligned} RHS &= \max_{1 \leq h \leq q} |W_{jh} - [\mathbf{Z}_2]_{kh}| \\ &= \max_{1 \leq h \leq q} \left| \sum_{l \in \phi_1^{-1}(h)} (1 - \gamma_{kl})[\mathbf{Z}_1]_{jl} \right| \\ &\overset{(e)}{\geq} \max_{1 \leq h \leq q} \left\{ \max_{l \in \phi_1^{-1}(h)} |(1 - \gamma_{kl})[\mathbf{Z}_1]_{j,l}| \right\} \\ &= \max_{1 \leq h \leq q} \left\{ \max_{l \in \phi_1^{-1}(h)} |[\mathbf{Z}_1]_{jl} - \hat{Z}_{kl}| \right\} = LHS. \end{aligned}$$

The inequality in (e) holds for the same reason as that in (d). We have proved (3.14) for $1 \leq p \leq \infty$, which implies $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ satisfies the triangle inequality when d is a p-norm.

Remark: The case for $p = \infty$ can also be seen by the equivalence of p-norms, for $1 \leq p \leq \infty$. On the other hand, the proof given is for case $|\mathcal{V}_x| \geq |\mathcal{V}_w| \geq |\mathcal{V}_y|$, but can be easily adapted to cases when $|\mathcal{V}_w| > |\mathcal{V}_x|$ or $|\mathcal{V}_w| < |\mathcal{V}_y|$ by imposing dilation/projection. For example, when

$|\mathcal{V}_w| \geq |\mathcal{V}_x| \geq |\mathcal{V}_y|$, we can first dilate $\mathcal{G}_X$ to the same size as $\mathcal{G}_W$ such that $\nu(\mathcal{G}_X, \mathcal{G}_{X'}) = 0$, then $\mathcal{G}_{X'}, \mathcal{G}_W$ and $\mathcal{G}_Y$ are in the same case in our proof. □

We have shown that $\nu(\mathcal{G}_X, \mathcal{G}_Y) \geq 0$ for all $\mathcal{G}_X$, $\mathcal{G}_Y$, and 0 is achieved if and only if $\mathcal{G}_X \simeq \mathcal{G}_{Y'}$. Note that, $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ in (3.9) is proved for the case when $|\mathcal{V}_x| \geq |\mathcal{V}_y|$. The triangle inequality also holds when $|\mathcal{V}_x| < |\mathcal{V}_y|$ by symmetry. Therefore, we have the following result.

Theorem 1 *If $d(\cdot, \cdot)$ in (3.8) is p-norm ($1 \leq p \leq \infty$), The dissimilarity function defined in (3.9) is a pseudo-metric between two directed weighted graphs.*

3.4 Extension and Discussion

3.4.1 Other choices of vector distance function $d(\cdot, \cdot)$

We have shown that when $d(\cdot, \cdot)$ is a p-norm ($1 \leq p \leq \infty$) on $\mathbb{R}^n$, the dissimilarity function $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ defined in (3.9) satisfies the conditions for being a pseudo-metric. The flexibility provided by using p-norms fulfills a range of practical needs. For instance, in an Internet graph, the l_1 distance between two nodes quantifies the total number of uncommon hyperlinks directed to two web pages [14]; in transportation networks, the l_2 (or the Euclidean) distance measures geological distance. In these examples, the dissimilarity/distance $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ defined in our framework provides a pseudo-metric. However, satisfying the conditions of a metric is not always necessary (nor possible) for practical problems. Nevertheless, for problems with the goal of minimizing distance/maximizing dissimilarity, alternative meaningful measures of similarity can be used as utility or cost functions.

One example is the *squared* Euclidean distance, instead of the l_2 distance, which provides three advantages: first, the squared Euclidean distance has a nice differential property, which is often preferred in algorithm design; second, the squared distance yields a larger penalty for large errors than the l_2 distance; and third, in many applications, the squared Euclidean distance is a natural choice for representing energy based costs. When $\mathcal{G}_Y$ closely approximates $\mathcal{G}_X$, these distance measures yield a similar error. Another example arises in state aggregation of Markov chains. Although the K-L divergence does not satisfy the symmetry property and is not a metric, it provides useful interpretations for problems related to probability distributions. The usage of the K-L divergence rate as a measure of distance between two Markov chains has been widely

accepted [13,54]. Therefore, although taking $d(\cdot,\cdot)$ as the K-L divergence in (3.4) does not provide a metric, some notion of similarity can still be inferred.

3.4.2 Connection with graph clustering formulation

Computing the distance $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ for given $\mathcal{G}_X$ and $\mathcal{G}_Y$ requires optimizing over a combinatorial number of partition functions. This problem becomes even more computationally intense when searching for a graph $\mathcal{G}_Y$ that minimizes $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ for a given $\mathcal{G}_X$. Most existing algorithms propose iterative solution candidates $\mathcal{G}_Y$ and compute $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ at each iteration. Our formulation of composite graphs makes it possible to combine the optimization problems of computing $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ and minimizing over possible $\mathcal{G}_Y$. This perspective gives a reinterpretation of the graph aggregation method proposed in [71], where a single optimization problem

$$\phi^*, \mathbf{Z}^* = \arg\min_{\phi,\mathbf{Z}} \rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y), \text{ for } \mathcal{C}_{XY}^{\phi} \in \bigcup_{\mathcal{G}_Y:|\mathcal{V}_y|=m} \boldsymbol{\mathcal{C}}_{XY}.$$

is posed and solved, approximately. This single optimization leads to an optimal composite graph $\mathcal{C}_{XY}^{\phi^*}(\mathbf{Z}^*) \in \{\bigcup_{\mathcal{G}_Y:|\mathcal{V}_y|=m} \boldsymbol{\mathcal{C}}_{XY}\}$, for all $\mathcal{G}_Y$. Moreover, the optimal aggregated graph $\mathcal{G}_Y^*$ may be determined from ϕ^* and $\mathbf{Z}^*$. A detailed solution is presented in the next chapter.

Summary of This Chapter

Our main achievements in this chapter are summarized as follows.

1. Proposition of the concept of composite graphs (*Definition 1*), and development of a dissimilarity measure to compare two directed weighted graphs with possibly different number of nodes (3.9).

2. Verification of the ability to identify graph isomorphisms, dilations and projections (*Proposition 3*). Proof that the dissimilarity measure is a pseudo-metric if the node comparison is conducted with the vector $p-$norms (*Theorem 1*).

3. Presentation of a single combinatorial optimization formulation for the graph aggregation problem, by application of the composite graph set (Section 3.4.2).

Chapter 4

Graph Aggregation using Maximum Entropy Principle

The derivation of the graph dissimilarity measures in Chapter 3 requires solving a combinatorial optimization problem; this doubles the difficulty in the determination of a small representative graph. However, in studies of neuroscience, social networks, Internet performance and multi-agent systems, where graph models are frequently used [15, 37, 39, 45, 73], graphs with large dimensions typically result from physical modeling or data based methods. Therefore, we look for some manageable practical methods to solve the problem aggregation problems.

In this chapter, we propose a method that directly solves for an optimal representative graph (of a given dimension), i.e., avoids iterations between evaluation of $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ and changing $\mathcal{G}_Y$, by reformulating the graph aggregation problem as a data clustering problem and working with a set of composite graphs $\boldsymbol{\mathcal{C}}_{XY}$. We then adapt the deterministic annealing algorithm in the solution process and obtain an efficient algorithmic framework that

1) is insensitive to implementation initialization

2) provides a good approximation for the optimal solution of the NP-hard data clustering problem.

Specifically, we consider aggregation of similar nodes of the original graph into supernodes, and then determine the edge weights between supernodes by minimizing the dissimilarity function between the original and the aggregated graphs. We will study three realizations of this framework for general directed weighted graphs, the Markov chains, and systems with interactive stochastic processes described by parametric models.

4.1 Aggregation Framework for General Directed Weighted Graphs

4.1.1 Data-clustering formulation

Consider the aggregation of a large graph $\mathcal{G}_X(\mathcal{V}_x, \mathcal{E}_x, \mathbf{X})$ with n nodes into $\mathcal{G}_Y(\mathcal{V}_y, \mathcal{E}_y, \mathbf{Y})$ with m nodes, where the representativeness is quantified by the dissimilarity function $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ defined in Section 3.2. The central tool in eliminating the iterative combinatorial optimizations (as discussed in Section 3.4.2) is the set of composite graphs (*Definition 1*).

By construction, the outgoing vectors of any composite graph $\mathcal{C}_{XY}^{\phi}(\mathbf{Z}) \in \boldsymbol{\mathcal{C}}_{XY}$ have the same dimension as those of $\mathcal{G}_X$ (see Figure 3.1). Thus we can define a distance matrix $\mathbf{D}(\mathbf{X}, \mathbf{Z}) \in \mathbb{R}^{n \times m}$ as

$$D_{ij} = [\mathbf{D}]_{i,j} = d(\mathbf{x}(i), \mathbf{z}(j)),$$

with $d(\cdot, \cdot)$ being a convex function that measures the vector distance. The form of $d(\cdot, \cdot)$ is case-specific, for example, the $l_1$norm, the squared Euclidean distance and the K-L divergence, respectively, can be choices for the Internet graphs, geographical graphs and Markov chains, respectively. Then the $\phi-$dissimilarity of $\mathcal{G}_X$ and any composite graph $\mathcal{C}_{XY}^{\phi}(\mathbf{Z})$, given by $\rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y)$ as in (3.8) shows the weighted average distance between the corresponding outgoing vectors assigned by partition ϕ, that is:

$$\rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y) = d_{\phi}(\mathcal{G}_X, \mathcal{C}_{XY}^{\phi}(\mathcal{V}_z, \mathcal{E}_z, \mathbf{Z})) \stackrel{\triangle}{=} \sum_{i=1}^{N} \mu_i \, d(\mathbf{x}(i), \mathbf{z}(\phi(i))) = \mathrm{Tr}(\mathbf{U}\mathbf{D}\boldsymbol{\Phi}^T), \tag{4.1}$$

in which $\mathbf{U} = \mathrm{diag}(\boldsymbol{\mu}) \in \mathbb{R}^{\mathrm{N} \times \mathrm{N}}$ denote the node weight. We will see later that this formulation gives a novel resource allocation perspective to the graph aggregation problem. The *dissimilarity* between $\mathcal{G}_X$ and $\mathcal{G}_Y$ is given by the minimum $\phi-$dissimilarity achieved over all composite graphs, that is

$$\nu(\mathcal{G}_X, \mathcal{G}_Y) \stackrel{\triangle}{=} \min_{\mathcal{C}_{XY}^{\phi} \in \boldsymbol{\mathcal{C}}_{XY}} \rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y). \tag{4.2}$$

in particular, this is the minimum over all aggregation matrices $\boldsymbol{\Phi}$ and weight matrices $\mathbf{Z}$. This

minimum exists since the number of possible partitions ϕ is finite (although combinatorial, and specifically $\binom{n}{m} m!$), and $d_\phi(\mathcal{G}_X, \mathcal{C}^\phi_{XY}(\mathbf{Z}))$ is convex in the rows of $\mathbf{Z}$ for each ϕ.

With the $\nu-$dissimilarity measure, our objective of finding a m-node aggregated graph for $\mathcal{G}_X$ becomes solving the following optimization problem:

$$\underset{\mathbf{\Phi},\mathbf{Y}:|V_y|=m}{\arg\min} \ \nu(\mathcal{G}_X, \mathcal{G}_Y), \tag{4.3}$$

$$\text{s.t.} \quad \mathbf{\Phi} \in \{0,1\}^{n\times m}, \ \mathbf{\Phi}\mathbf{1}_M = \mathbf{1}_N, \ \mathbf{Y} = \mathbf{Z}\mathbf{\Phi}. \tag{4.4}$$

This problem is NP-hard [1], partially resulting from the integer constraint $\mathbf{\Phi}$ belongs to $\{0,1\}^{N\times M}$; thus we aim to approximate the optimal $\mathbf{\Phi}^*$ and $\mathbf{Y}^*$.

4.1.2 Solution based on Deterministic Annealing Algorithm

We decompose the optimization problem (4.3) into two stages:

(I) *Node clustering:* Solve $\arg\min_{\mathbf{Z},\mathbf{\Phi}} \rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y)$, where $\mathcal{C}^\phi_{XY}(\mathbf{Z}) \in \{\boldsymbol{\mathcal{C}}_{XY} : |V_y| = m\}$.

(II) *Edge aggregation:* Obtain $\mathbf{Y}^*$ (and therefore $\mathcal{G}^*_Y$) from (4.4) using $\mathbf{Z}^*$ and $\mathbf{\Phi}^*$ from step (I).

The optimization problem in step (I) can be viewed as a resource allocation problem in which the set of N nodes in the given graph $\mathcal{G}_X$ is partitioned into M cells; to each cell a representative supernode is to be allocated such that the averaged pairwise distance between a node and its representative supernode (4.1) is minimized. Equivalently we want to partition the set of n outgoing vectors $\{\mathbf{x}(i)\}_{i=1}^n$ into m cells and to each cell allocate a representative outgoing vector $\mathbf{z}(j)$ (the jth row of the weight matrix $\mathbf{Z}$) such that the total representation error $\nu(\mathcal{G}_X, \mathcal{G}_Y) = \min \rho_{\phi,\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y) = \min d_\phi(\mathcal{G}_X, \mathcal{C}^\phi_{XY}(\mathbf{Z}))$ is minimized. As indicated, we seek an approximation of the optimal $\mathbf{\Phi}^*$ and $\mathbf{Z}^*$ by considering this problem with a relaxed version of the integer constraint (4.4). Specifically, we adapt the DA algorithm [55, 60] to address step (I).

The main idea of the DA algorithm is to incorporate *soft partitioning*, that is, instead of using a partition function ϕ that defines a binary aggregation matrix $\mathbf{\Phi} \in \{0,1\}^{N\times M}$, each node i $(1 \le i \le n)$ is associated with *all* supernodes j $(1 \le j \le m)$ via *nonnegative association weights* $p_{j|i} = \tilde{\Phi}_{ij}$. We choose $\{p_{j|i}\}$ to satisfy $\sum_{j=1}^m p_{j|i} = 1$ for all i, and define a *soft aggregation matrix*

$\tilde{\mathbf{\Phi}} \in [0,1]^{N\times M}$, with $\tilde{\Phi}_{ij} = p_{j|i}, i = 1,\ldots,n, j = 1,\ldots,m$. Then, we modify the dissimilarity functions in (4.1) and (4.2) into the following *soft dissiimlarities*,

$$\begin{aligned}\tilde{\rho}_{\tilde{\phi},\tilde{\mathbf{Z}}}(\mathcal{G}_X,\mathcal{G}_Y) &\triangleq d_{\tilde{\phi}}(\mathcal{G}_X,\mathcal{C}^{\tilde{\phi}}_{XY}(\mathbf{Z})) \\ &= \sum_{i=1}^{n}\mu_i\sum_{j=1}^{m}p_{j|i}d(\mathbf{x}(i),\mathbf{z}(j)) = \mathrm{Tr}(\mathbf{U}\mathbf{D}\tilde{\mathbf{\Phi}}^T), \qquad (4.5)\\ \text{and } \tilde{\nu}(\mathcal{G}_X,\mathcal{G}_Y) &\triangleq \min\{\tilde{\rho}_{\tilde{\phi},\tilde{\mathbf{Z}}}(\mathcal{G}_X,\mathcal{G}_Y)|\mathbf{Y}=\mathbf{Z}\tilde{\mathbf{\Phi}},\tilde{\mathbf{\Phi}}\in[0,1]^{n\times m}\}. \qquad (4.6)\end{aligned}$$

The allowance of soft partitioning introduce uncertainty in associating a node to any supernodes; this uncertainty is quantified by a Shannon entropy term as

$$H(\mathcal{G}_Y|\mathcal{G}_X) = H(\mathcal{C}^{\tilde{\phi}}_{XY}(\mathbf{Z})|\mathcal{G}_X) = -\sum_{i=1}^{n}\mu_i\sum_{j=1}^{m}\tilde{p}_{j|i}\log\tilde{p}_{j|i}. \qquad (4.7)$$

The association weights $\{p_{j|i}\}$ are determined by the Maximum Entropy Principle (MEP) introduce by E.T. Jayes [33]. Specifically, we minimize $\tilde{\rho}_{\tilde{\phi},\tilde{\mathbf{Z}}}(\mathcal{G}_X,\mathcal{G}_Y)$ under the following entropy constraint

$$H(\mathcal{C}^{\tilde{\phi}}_{XY}(\mathbf{Z})|\mathcal{G}_X) = \boldsymbol{\mu}^T\mathbf{h} = \sum_{i=1}^{n}\mu_i h_i = H_0, \qquad (4.8)$$

where $\mathbf{h} = [h_1,\ldots,h_N] \in \mathbb{R}^n_+$ is defined by

$$h_i = H(\tilde{\boldsymbol{\phi}}(j)) = -\sum_{k=1}^{m}p_{k|i}\log p_{k|i},$$

with $H(\cdot)$ being the Shannon entropy of a probability vector, and $\tilde{\boldsymbol{\phi}}(j) = [p_{1|j},\ldots,p_{m|j}]$ being the jth row of the soft aggregation matrix $\tilde{\mathbf{\Phi}}$. In short, we modify step (I) to solve a continuous relaxation of (4.3):

$$\begin{aligned}&\underset{\tilde{\mathbf{\Phi}},\mathbf{Z}:|V_y|=m}{\arg\min}\ \tilde{\rho}_{\tilde{\phi},\tilde{\mathbf{Z}}}(\mathcal{G}_X,\mathcal{G}_Y) = \mathrm{Tr}(\mathbf{U}\mathbf{D}\tilde{\mathbf{\Phi}}^T), \qquad (4.9)\\ &\text{s.t. } \tilde{\mathbf{\Phi}}\in[0,1]^{n\times m},\ \tilde{\mathbf{\Phi}}\mathbf{1}_m = \mathbf{1}_n,\ \mathbf{Y}=\mathbf{Z}\tilde{\mathbf{\Phi}},\ H(\mathcal{C}^{\tilde{\phi}}_{XY}(\mathbf{Z})|\mathcal{G}_X) = H_0, \qquad (4.10)\end{aligned}$$

for a feasible value of H_0 ($0 \leq H_0 \leq \log m$). This is solved by minimizing the Lagrangian $F(\mathcal{G}_X, \mathcal{C}_{XY}^{\tilde{\phi}}(\mathbf{Z})) \stackrel{\triangle}{=} \tilde{\rho}_{\tilde{\phi},\mathbf{Z}}(\mathcal{G}_X, \mathcal{G}_Y) - \frac{1}{\beta} H(\mathcal{C}_{XY}^{\tilde{\phi}}(\mathbf{Z})|\mathcal{G}_X)$ with respect to $\{p_{j|i}\}$, where $\frac{1}{\beta}$ is a Lagrange multiplier (and β is the annealing parameter). This yields a *Gibbs distribution*

$$p_{j|i} = \frac{\exp\{-\beta D_{ij}\}}{\sum_{k=1}^{m} \exp\{-\beta D_{ik}\}}. \tag{4.11}$$

Remark: Note that each value of the annealing parameter β corresponds to a value of H_0 (obtained by substituting (4.11) in (4.8)). In [32, 55], it is shown that the larger the value β, the smaller the value of the corresponding H_0. In the DA algorithm, the relaxed problem (4.9) is repeatedly solved with increasing values of parameter β, i.e., decreasing (yet feasible) values of H_0.

Substituting the association weights (7.5) into (4.5) and (4.8), the Lagrangian F becomes

$$F^*(\mathcal{G}_X, \mathcal{C}_{XY}^{\tilde{\phi}}(\mathbf{Z})) = -\frac{1}{\beta} \sum_{i=1}^{n} \mu_i \log \sum_{k=1}^{m} \exp\{-\beta D_{ik}\}. \tag{4.12}$$

The DA algorithm incorporates an annealing process characterized by the annealing parameter β. At each iteration of the DA algorithm, indicated by a fixed β, a local minimum of (4.12) is computed. That is, the representative outgoing vectors $\mathbf{z}^*(j)$ are computed using the following implicit equation,

$$0 = \nabla_{\mathbf{z}(j)} F^*(\mathcal{G}_X, \mathcal{C}_{XY}^{\tilde{\phi}}(\mathbf{Z})) = \sum_{i=1}^{n} \mu_i p_{j|i} \nabla_{\mathbf{z}(j)} D_{ij}. \tag{4.13}$$

Equation (4.13) is solved using gradient descent methods where the solutions from the previous iteration are used as starting values in the current iteration. These computations are repeated as the parameter β is increased in the annealing process.

The rationale behind this annealing is as follows. For $\beta \approx 0$, minimizing the cost function F is approximately the same as minimizing $-H$, which is convex and has a global minimum. In fact in this case, the association weights given by (7.5) are approximately uniform (i.e., $p_{j|i} \approx \frac{1}{m}, \forall i, j$), so all outgoing vectors $\mathbf{z}^*(j)$ are coincident, thus there is a single distinct supernode. As β is increased, the soft aggregation matrix $\tilde{\mathbf{\Phi}}$ becomes more and more binary, moreover, the annealing process exhibits a series of *phase transitions* as shown in [55, 60], where the solutions $\{\mathbf{z}^*(j)\}$ are

insensitive to changes in β except at critical values β_c; the number of *distinct* outgoing vectors in the composite graph increases at these critical values. When β is very large, $F \approx \tilde{\rho}_{\tilde{\phi},\mathbf{Z}} \approx \rho_{\phi,\mathbf{Z}}$ (since $p_{j|i} \approx 1$ if $j = \arg\min_k D_{ik}$ and otherwise is ≈ 0), and thus we recover a hard partition and the original dissimilarity function. The underlying heuristic of the DA algorithm is that it finds the global minimum of F at very small β and tracks the minimum as β is increased.

After obtaining the weighting matrix $\mathbf{Z}^*$ from step (I) as above, we can determine the weighting matrix $\mathbf{Y}^*$ by soft edge aggregation as in step (II): $\mathbf{Y}^* = \mathbf{Z}^*\tilde{\mathbf{\Phi}}$.

The key aspects of the graph aggregation process described in the preceding and the DA algorithm are summarized in *Lemma 3* below. For clearer exposition, we choose the vector distance d in (4.6) to be the squared Euclidean distance function.

Lemma 3 *Let* $d(\mathbf{u},\mathbf{v}) \stackrel{\triangle}{=} \|\mathbf{u}-\mathbf{v}\|_2^2$, *then: (i) the weighting matrix* $\mathbf{Z}^*$ *of* $\mathcal{C}_{XY}^{\tilde{\phi}}(\mathbf{Z}^*)$ *that satisfies the first-order optimality condition in* (4.13) *for a given* β *and the corresponding weighting matrix* $\mathbf{Y}^*$ *for* $\mathcal{G}_Y^*$ *are given by:*

$$\mathbf{Z}^* = \mathbf{R}^T\mathbf{X} \text{ and } \mathbf{Y}^* = \mathbf{R}^T\mathbf{X}\tilde{\mathbf{\Phi}}, \tag{4.14}$$

where $\tilde{\mathbf{\Phi}}, \mathbf{R} \in [0,1]^{n\times m}$,

$$\begin{aligned} \tilde{\Phi}_{ij} &= p_{j|i} = \frac{e^{-\beta d(\mathbf{x}(i),\mathbf{z}^*(j))}}{\sum_k e^{-\beta d(\mathbf{x}(i),\mathbf{z}^*(k))}} \\ \text{and } R_{ij} &= q_{i|j} = \frac{\mu_i p_{j|i}}{\sum_r \mu_r p_{j|r}}, i = 1,\ldots,n,\ j = 1,\ldots,m; \end{aligned} \tag{4.15}$$

(ii) the number of distinct outgoing vectors increases when β *surpasses a critical value* β_c, *at which the determinant of the Hessian,* $\det\left(\nabla^2_{z^*(j_0)} F^*(\mathcal{G}_X, \mathcal{C}_{XY}^{\tilde{\phi}}(\mathbf{Z}))\right) = 0$ *for some* $1 \le j_0 \le m$. *Moreover,* β_c^{-1} *is given by twice the maximum eigenvalue of the matrix given by* $\sum_{i=1}^n q_{i|j}(\mathbf{x}(i)-\mathbf{z}^*(j_0))(\mathbf{x}(i)-\mathbf{z}^*(j_0))^T$.

Remark: *Lemma 3* is from [55]. Part (i) is obtained by solving (4.13) with the squared Euclidean distance. Part (ii) is a direct consequence of the MEP-based algorithm and its properties (see [55,60] for details). It should be noted that the solution $\mathbf{Z}^*$ in (4.14) is insensitive to changes in β between two successive critical values of β_c (see [60] for quantitative details). As β is increased beyond a

critical β_c, the number of *distinct* solutions $\mathbf{z}^*(j)$ to (4.13) increases. Thus for graph aggregation, we update $\tilde{\mathbf{\Phi}}$, $\mathbf{R}$, $\mathbf{Z}^*$ and $\mathbf{Y}^*$ by (4.14) for each β value as β is increased, and stop when the number of *distinct* outgoing vectors $\mathbf{z}^*(j)$ (or the rows of $\mathbf{Z}^*$) equals m.

4.1.3 A Guided Decentralized Implementation

By viewing the graph aggregation problem from a resource allocation perspective, we can apply the principles of the MEP-based algorithm to identify an appropriate aggregated graph size, use the flexibility in choosing a distance measure to define a dissimilarity function, impose case specific constraints, and avoid poor local minima [55, 59, 60]. Further, the computational cost of solving large eigenvalue/eigenvector problems in spectral-decomposition methods is avoided. Unlike many clustering based methods that conduct repeated bipartitioning (such as the min-flow cut and the simulation based aggregations [13]), hierarchical multiple-partitions result from splitting. In fact, as β tends to infinity, the proposed algorithm mimics Lloyd's algorithm, but with a carefully selected initial guess for the representative points (obtained from the previous annealing steps).

Since the aggregation result is insensitive to the value of β (see [55, 60, 72]), our algorithm uses a geometric annealing schedule (e.g., $\beta_{k+1} = \alpha\beta_k$ for $\alpha > 1$) and thus requires far fewer iterations than popular annealing algorithms such as simulated annealing (which typically has a logarithmic cooling law [23]).

On the other hand, the MEP-based algorithm requires *centralized* computations to overcome convergence to local minima, i.e., it uses information from all $\mathbf{x}(i)$'s to compute each $\mathbf{z}^*(j)$; in this sense, the computational effort at each iteration is high. However, as the annealing progresses (as β increases), the association weights $p_{j|i}$ given by (4.11) tend to either 0 for distant node-supernode pairs or 1 for nearby pairs, i.e.,

$$p_{j|i} = \frac{\exp\{-\beta[d(\mathbf{x}(i), \mathbf{z}(j)) - d_{\min}(i)]\}}{1 + \sum_{k \neq j_0} \exp\{-\beta[d(\mathbf{x}(i), \mathbf{z}(k)) - d_{\min}(i)]\}}.$$

where $d_{\min}(i) \stackrel{\triangle}{=} \min_j d(\mathbf{x}(i), \mathbf{z}(j))$ and $j_0 \stackrel{\triangle}{=} \arg\min_j d(\mathbf{x}(i), \mathbf{z}(j_0))$. This decentralization can be exploited by replacing the most expensive computation $\mathbf{z}^*(j) = \sum_{i=1}^{n} q_{i|j}\mathbf{x}(i)$ in (4.14) by $\hat{\mathbf{z}}^*(j) = \sum_{i \in \mathcal{S}_j} q_{i|j}\mathbf{x}(i)$ for an appropriate subset of nodes close to the j^{th} supernode, $\mathcal{S}_j$. For instance, we

can choose $\mathcal{S}_j$ by the Voronoi cells centered at the outgoing vectors $\mathbf{z}(j)$, and have the following result.

Proposition 4 *The centroid outgoing vectors $\mathbf{z}(j)'$ calculated by using only the outgoing vectors of nodes inside the Voronoi cells,*

$$\mathcal{S}_j \stackrel{\triangle}{=} \{\mathbf{x}(i) : \min_k d(\mathbf{x}(i), \mathbf{z}(k)) = d(\mathbf{x}(i), \mathbf{z}(j))\},\ 1 \le j \le m,$$

is close to the results from centralized calculation $\mathbf{z}(j)$ when β is large. Specifically, in order to guarantee $\|\mathbf{z}^*(l) - \hat{\mathbf{z}}^*(l)\|_\infty < \tau$ for some positive number τ, it is sufficient to conduct restrictive calculation only when the following condition is true for all cells $\{\mathcal{S}_j\}$:

$$\varepsilon_{ij} \stackrel{\triangle}{=} \sum_{l \in \mathcal{S}_i} \mu_l p_{j|l} < \varepsilon_0 \stackrel{\triangle}{=} \frac{\tau \min_{1 \le j \le m} \sum_{i=1}^n \mu_i p_{j|i}}{(m-1) \max_{1 \le t \le n} \sum_{i=1}^n |X_{it}|},\ \forall i \ne j, \tag{4.16}$$

with some positive numbers ε_0.

Proof: We first expand the optimal outgoing vectors of the centroid calculation (4.14) by plug in the $p_{i|j}$ and $q_{j|i}$ in (4.15), which gives

$$\mathbf{z}(l) = \frac{\sum_{i=1}^n \mu_i p_{l|i} \mathbf{x}(i)}{\sum_{i=1}^n \mu_r p_{l|i}}. \tag{4.17}$$

We define the following terms to simplify the expression:

- $A_l(\mathcal{S}) \stackrel{\triangle}{=} \sum_{i=1}^n \mu_i p_{l|i} \mathbf{x}(i) \in \mathbb{R}^{1 \times n}$, and $B_l(\mathcal{S}) \stackrel{\triangle}{=} \sum_{i=1}^n \mu_i p_{l|i} \in \mathbb{R}$, which are the numerator and the denominator of (4.17);
- $S(\mathcal{S}) \stackrel{\triangle}{=} \sum_{i=1}^n |\mathbf{x}(i)| \in \mathbb{R}^{1 \times n}$, is the vector sum of the absolute values of $\mathbf{x}(i)$'s.

Similarly, when these quantities are restricted to Voronoi cell $\mathcal{S}_l$ containing $\mathbf{z}(l)$, we have:

- $A_l(\mathcal{S}_l) \stackrel{\triangle}{=} \sum_{\mathbf{x}(i) \in \mathcal{S}_l} \mu_i p_{l|i} \mathbf{x}(i)$, $B_l(\mathcal{S}_l) \stackrel{\triangle}{=} \sum_{\mathbf{x}(i) \in \mathcal{S}_l} \mu_i p_{l|i}$;
- $S(\mathcal{S}_l) \stackrel{\triangle}{=} \sum_{\mathbf{x}(i) \in \mathcal{S}_l} |\mathbf{x}(i)|$.

Therefore, the outgoing vector in centralized computation given by (4.17) can be written as

$$\mathbf{z}(l) \quad = \quad \frac{A_l(\mathcal{S})}{B_l(\mathcal{S})}.$$

After eliminating the nodes outside the cell $\mathcal{S}_l$, (only use nodes within $\mathcal{S}_l$ to compute $\mathbf{z}(l)$ for each l), the decentralized calculation gives $\mathbf{z}(l)'$ as

$$\mathbf{z}(l)' \quad = \quad \frac{\sum_{\mathbf{x}(i)\in\mathcal{S}_l} \mu_i p_{l|i} \mathbf{x}(i)}{\sum_{\mathbf{x}(i)\in\mathcal{S}_l} \mu_i p_{l|i}} = \frac{A_l(\mathcal{S}_l)}{B_l(\mathcal{S}_l)}.$$

The error from decentralization, $\mathbf{z}(l) - \mathbf{z}(l)'$, is given by

$$\begin{aligned} \mathbf{z}(l) - \mathbf{z}(l)' &= \frac{A_l(\mathcal{S})B_l(\mathcal{S}_l) - A_l(\mathcal{S}_l)B_l(\mathcal{S}_l)}{B_l(\mathcal{S}_l)B_l(\mathcal{S})} \\ &= \frac{A_l(\mathcal{S}_l^c)B_l(\mathcal{S}_l) - A_l(\mathcal{S}_l)B_l(\mathcal{S}_l^c)}{B_l(\mathcal{S}_l)B_l(\mathcal{S})}, \end{aligned}$$

where $\mathcal{S}_l^c = \mathcal{S} \setminus \mathcal{S}_l$. Without loss of generality, consider the supremum norm of above error vector:

$$\begin{aligned} \|\mathbf{z}(l) - \mathbf{z}(l)'\|_\infty &= \max_{1\le j\le n} |Z_{lj} - Z'_{lj}| \\ &= \frac{1}{B_l(\mathcal{S}_l)B_l(\mathcal{S})} \max_{1\le j\le n} |[B_l(\mathcal{S}_l)A_l(\mathcal{S}_l^c)]_j - [B_l(\mathcal{S}_l^c)A_l(\mathcal{S}_l)]_j| \\ &\le \frac{1}{B_l(\mathcal{S}_l)B_l(\mathcal{S})} \max_{1\le j\le n} \left\{|[B_l(\mathcal{S}_l)A_l(\mathcal{S}_l^c)]_j| \vee |[B_l(\mathcal{S}_l^c)A_l(\mathcal{S}_l)]_j|\right\}. \end{aligned}$$

Meanwhile, we note that $|A_l(\mathcal{S}_l)| = |\sum_{\mathbf{x}(i)\in\mathcal{S}_l} \mathbf{x}(i)\mu_i p_{l|i}| \le \left(\sum_{\mathbf{x}(i)\in\mathcal{S}_l} |\mathbf{x}(i)|\right) B_l(\mathcal{S}_l) = S(\mathcal{S}_l)B_l(\mathcal{S}_l)$, and $|A_l(\mathcal{S}_l^c)| \le S(\mathcal{S}_l^c)B_l(\mathcal{S}_l^c)$. Thus the worst case error of all $\mathbf{z}(l)$'s becomes

$$\begin{aligned} \|\mathbf{z}(l) - \mathbf{z}(l)'\|_\infty &\le \frac{1}{B_l(\mathcal{S}_l)B_l(\mathcal{S})} \max_{1\le j\le n} \left\{|[B_l(\mathcal{S}_l)B_l(\mathcal{S}_l^c)S(\mathcal{S}_l^c)]_j| \vee |[B_l(\mathcal{S}_l)B_l(\mathcal{S}_l^c)S(\mathcal{S}_l)]_j|\right\} \\ &\le \frac{B_l(\mathcal{S}_l^c)}{B_l(\mathcal{S})} \max_{1\le j\le n} \left\{|[S(\mathcal{S}_l^c)]_j| \vee |[S(\mathcal{S}_l)]_j|\right\}. \end{aligned}$$

For the Voronoi set, $\mathcal{S}_l$'s are disjoint, thus we have

$$\max_{1\le j\le n} \left\{|[S(\mathcal{S}_l^c)]_j| \vee |[S(\mathcal{S}_l)]_j|\right\} \quad \le \quad \max_{1\le j\le n} \left\{|[S(\mathcal{S})]_j|\right\} = \|S(\mathcal{S})\|_\infty,$$

which yields $\|\mathbf{z}(l) - \mathbf{z}(l)'\|_\infty \leq \|S(\mathcal{S})\|_\infty \frac{B_l(\mathcal{S}_l^c)}{B_l(\mathcal{S})}$. If we use the definition of ε_{ij} and condition $\varepsilon_{ij} \leq \varepsilon_0$ for all $i \neq j$ as required in (4.16), we have $B_l(\mathcal{S}_l^c) \leq (m-1)\varepsilon_0$. Moreover, $\|S(\mathcal{S})\|_\infty = \max_{1\leq t\leq n} \sum_{i=1}^n |X_{it}|$. Therefore, one sufficient condition to guarantee $\|\mathbf{z}(l) - \mathbf{z}(l)'\|_\infty < \tau$ for all l is to select $\varepsilon_0 \leq \frac{\tau B_l(\mathcal{S})}{(m-1)\max_{1\leq t\leq n} \sum_{i=1}^n |X_{it}|}$, which is guaranteed by (4.16).

Finally, condition (4.16) is always satisfied when β is large, since in this case, $p_{j|l} \approx 0$ for all $l \in \mathcal{S}_i, i \neq j$. □

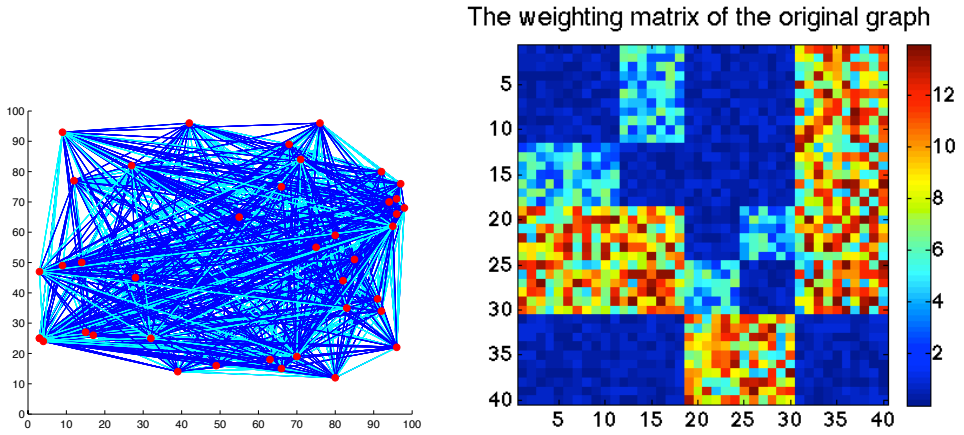
Simulation results demonstrate the potential to improve scalability (Table I). It is seen through this and other examples that adopting decentralized calculation in the solution process provides a significant improvement in computational times, while the sacrifice in computational accuracy compared to the original DA procedure is controllable. Alternative choices for the cells $\mathcal{S}_j$ can be found in [41, 59], in which the authors discussed the approximation could be improved when implementing the localization in an adaptive way.

4.2 Simulation

4.2.1 A general graph aggregation

Figure 4.1 and 4.2 present results for an aggregation of a graph with 40 nodes into a graph with 5 supernodes. This example is motivated by neuroscience studies, and the graph (plot (a)) shows the information flows among 40 networked neurons. Specifically, each node stands for a neuron and an arrow indicates the directed information between two neurons. The weighting matrix is shown in plot (b), in which the color of the i^{th} row and j^{th} column represents the edge weight x_{ij} (indicated by the color bar). We select a uniform node weight $\mu_i = 1/40, \forall i$, and adopt the squared Euclidean distance for $d(\cdot, \cdot)$ in (4.1) to compare the functional dissimilarities between neurons. We apply the two-step aggregation method, where for small $\beta = 0.01$, all supernodes are coincident. As β is increased, the number of distinct supernodes increases from 1 to 15 through splitting, and the dissimilarity achieved by the aggregated graph decreases (plot (c)). From the dissimilarity curve, we estimate the natural size of the aggregated graph to be $m = 5$ clusters. The weighting matrix for the resulting 5-supernode aggregated graph is given in plot (d), which is consistent with the way that this test data was constructed.

To demonstrate the decentralization method of Section 4.1.3, we form the Voronoi cells for each supernode at each β, and for those cells that satisfy the threshold criterion, the computations are restricted to nodes within the cells as discussed. Plots (e) and (f) show 3 and 5 supernodes at different stages of the annealing process. The running times and dissimilarity, achieved by the original and the modified algorithms to obtain a 5-supernode graph are presented in Table 4.1, which shows a significant reduction in computation time.



(a) (b)

Figure 4.1: Multi-scaled aggregation results with squared Euclidean distance for a graph of 40 nodes. Plot (a) shows the original directed graph and the locations of all nodes, cyan arrows indicate weak connections (with small edge weights) and blue arrows indicate strong connections. Plot (b) shows the weighting matrix of the original graph.

We finally compare our algorithm with a standard Normalized Graph Cut algorithm proposed by Shi and Malik [61]. As the size of original graph is increased from 50-node to 10000-node, the times needed to implement spectral clustering (t_{SM}) and our aggregation method (t_{DA}), and the (hard) dissimilarities $\nu(\mathcal{G}_X, \mathcal{G}_Y)$ (as in (4.3)) achieved by both methods are shown in Table 4.2. Note that the times and ratios in Table 4.2 represent the average of 5 implementations on the same datasets. Therefore, it is evident that the aggregated graph obtained using the MEP-based algorithm consistently provides a better representativeness ($\frac{\nu_{DA}}{\nu_{SM}} \approx 0.08$) than Normalized Graph Cut. Also, for very large graphs (roughly more than 1000 nodes), the MEP-based algorithm is also more efficient.

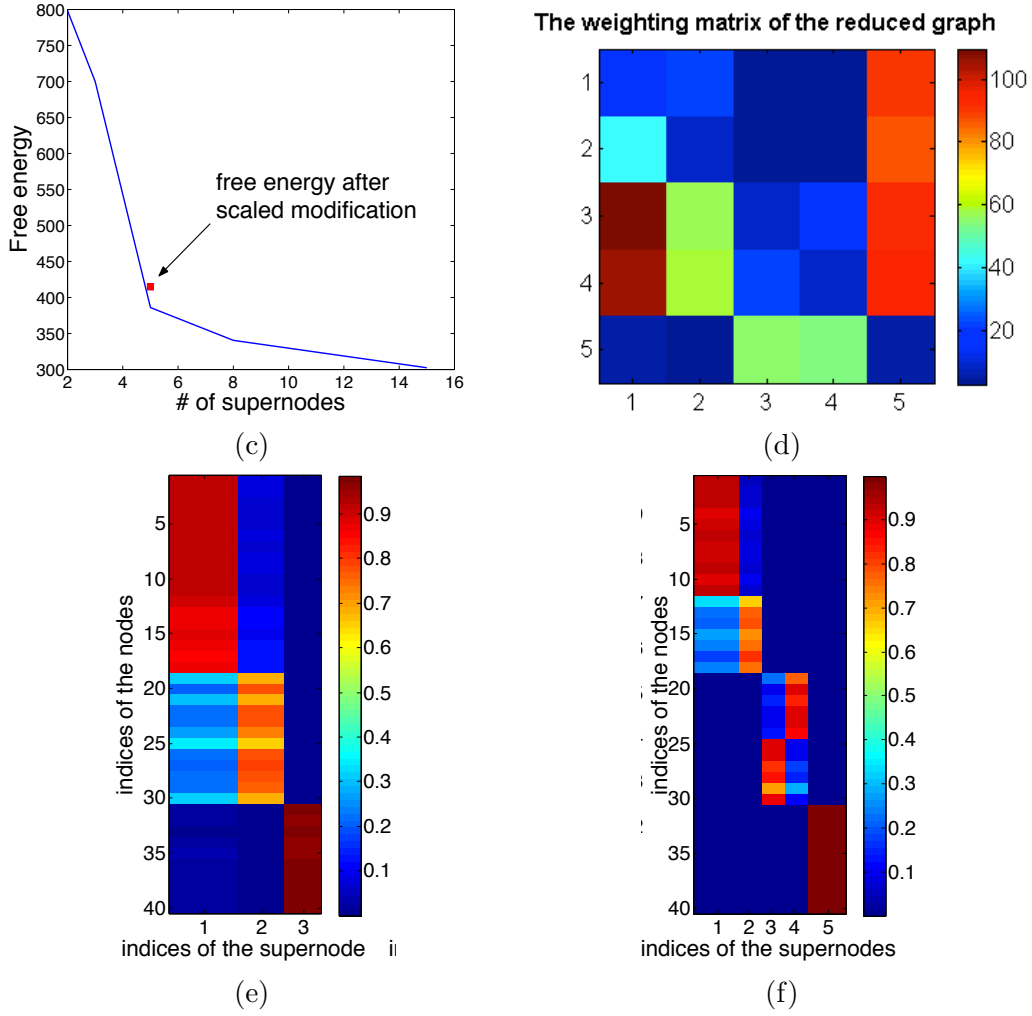


Figure 4.2: Multi-scaled aggregation results with squared Euclidean distance for a graph of 40 nodes. Plot (c) depicts the dissimilarity value versus the number of supernodes, the red dot marks the dissimilarity achieved by the 5-supernode aggregated graph after applying decentralization. Plot (d) provides the weighting matrix of the 5-supernode aggregated graph. Plot (e) shows the association weights when there are 3 supernodes, from which we see the influence from remote nodes in determining the $\mathbf{z}(1)$ is very small, so we omit the nodes outside the Voronoi cells for supernodes $\mathbf{z}_1$ and $\mathbf{z}_2$ in subsequent calculations. Plot (f) shows the association weights from each node to each supernode when the algorithm terminates as 5 supernodes have been identified.

Table 4.1: Decentralized modification vs. the original algorithm

40-node graph	Running time (sec)	Free energy value
Centralized computation	10.6366	386.0528
Decentralized modification	5.3401	415.3161

Table 4.2: Comparison with graph spectral clustering

n	100	1000	5000	7500	10000
t_{SM} (sec)	0.0401	0.9250	57.1976	178.5284	400.4039
t_{DA} (sec)	0.0521	0.8808	23.038	52.8473	76.344
$(\nu_{DA}/\nu_{SM})\%$	8.14	8.11	9.38	7.86	7.87

Summary of This Chapter

Our main achievements in this chapter are summarized as follows.

1. Adaptation of the resource allocation framework to aggregation of large interconnected systems, in particular, the directed weighted graphs and Markov chains.

2. Proposition of a two-step aggregation procedure for graph aggregation, based on the composite graphs set (Section 4.1.2).

3. Presentation of a guided decentralized implementation for the algorithm to improve scalability (Section 4.1.3).

Chapter 5

Aggregation of Markov Chains

5.1 Aggregation of Markov Chains

Based on the obvious correspondence between graph models and Markov chains, we formulate a special constrained optimization problem for Markov chain reduction, and develop an algorithmic implementation based on the MEP-based algorithm.

The aggregation problem for the Markov chains can be stated as: For a given a Markov chain $\mathcal{X}$ with n states, whose transition probability matrix is $\mathbf{P}$, find a low-order chain $\mathcal{Y}$ with m states and transition matrix $\mathbf{Q}$ such that the *dissimilarity* between $\mathcal{X}$ and $\mathcal{Y}$ is minimized.

A discrete Markov chain $\mathcal{X} = \{X(1), X(2), \cdots\}$ with finite state space and transition matrix $\mathbf{P}$ (with the i^{th} row $\mathbf{p}(i)$ and $[\mathbf{P}]_{i,j} = P_{ij}$) can be represented by a graph $\mathcal{G}_P(\mathcal{V}_p, \mathcal{E}_p, \mathbf{P})$. When the state space $|\mathcal{X}| = n$ is large, we aim to find a low-order Markov chain $\mathcal{Y}$ with M states and transition matrix $\mathbf{Q}$ (with the jth row being $\mathbf{q}(j)$ and $[\mathbf{Q}]_{i,j} = Q_{ij}$) to represent $\mathcal{X}$, where representativeness is defined through their graphs. That is, our two-step framework for graph aggregation applies here by aggregating similar states, such that the dissimilarity between their corresponding graphs $\mathcal{G}_P$ and $\mathcal{G}_Q(V_q, E_q, \mathbf{Q})$ given by $\nu(\mathcal{G}_P, \mathcal{G}_Q)$ in (4.3), is minimized. After applying soft aggregation, the optimization problem associated with the first step is given by (4.9) with constraints

$$Z \geq 0,\ Z\mathbf{1}_N = \mathbf{1}_M \tag{5.1}$$

in addition to (4.10). In this setting, we choose the K-L divergence for $d(\cdot,\cdot)$, that is $D_{ij} = d(\mathbf{p}(i), \mathbf{z}(j)) = \sum_{k=1}^{N} P_{ik} \log \frac{P_{ik}}{Z_{jk}}$, which is convex, and we assume that the support set of $\mathbf{p}(i)$ is contained in the support set of $\mathbf{z}(j)$ for all i and j (i.e., $Z_{jk} = 0 \Rightarrow P_{ik} = 0, \forall i, j, k$), and $H_0 \in [0, \log M]$. The constraint (5.1) defines transition probabilities from a *superstate* in $\mathcal{Y}$ to the

original states in $\mathcal{X}$, and also guarantees that $\mathbf{Q}$ computed by $\mathbf{Z}\tilde{\mathbf{\Phi}}$ is a valid Markov transition matrix.

Remark: If the original Markov chain $\mathcal{X}$ has a limiting distribution $\boldsymbol{\pi} \in \mathbb{R}_+^{1\times n}$ satisfying $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ and $\sum_{i=1}^n \pi_i = 1$ (e.g., is an irreducible and aperiodic Markov chain), a natural choice of the node weights is $\mu_i \stackrel{\triangle}{=} \pi_i$ for all i. In this case the resulting dissimilarity function $\tilde{\nu}(\mathcal{G}_P, \mathcal{G}_Q) \stackrel{\triangle}{=} \min_{\mathbf{Q}:|\mathcal{V}_q|=m} \tilde{\rho}_{\tilde{\phi},\mathbf{Z}}(\mathcal{G}_P, \mathcal{C}_{PQ}^{\tilde{\phi}}(\mathbf{Z}))$ provides a soft version of the *lifted* K-L divergence rate metric between two Markov chains proposed in [54].

The Lagrangian, after accounting for the Markov constraints (5.1), becomes

$$F(\mathcal{G}_P, \mathcal{C}_{PQ}^{\tilde{\phi}}(\mathbf{Z})) = \tilde{\rho}_{\tilde{\phi},\mathbf{Z}}(\mathcal{G}_P, \mathcal{C}_{PQ}^{\tilde{\phi}}(\mathbf{Z})) - \frac{1}{\beta} H(\mathcal{C}_{PQ}^{\tilde{\phi}}(\mathbf{Z})|\mathcal{G}_P) - \sum_{j=1}^m \eta_j \sum_{k=1}^n (Z_{jk} - 1),$$

with η_j's being the Lagrange multipliers associated with (5.1). We follow the general two step framework, which leads to results stated in *Theorem 2* and *Theorem 3*.

Theorem 2 *Given a Markov chain with n states and transition probability matrix* $\mathbf{P}$*, the transition matrix* $\mathbf{Q}^*$ *of the low-order Markov chain calculated from the two step aggregation with* $m < n$ *superstates is given by* $\mathbf{Q}^* = \mathbf{Z}^*\tilde{\mathbf{\Phi}}$*, where* $\mathbf{Z}^* = \mathbf{R}^T\mathbf{P}$*,* $\tilde{\Phi}_{il} = p_{l|i} = \frac{e^{-\beta d(\mathbf{p}(i), \mathbf{z}^*(l))}}{\sum_{t=1}^m e^{-\beta d(\mathbf{p}(i), \mathbf{z}^*(t))}}, l = 1, \ldots, m, i = 1, \ldots, n$*, with* $d(\cdot,\cdot)$ *being the K-L divergence, and* $R_{sk} = q_{s|k} = \frac{\mu_s p_{k|s}}{\sum_t \mu_t p_{k|t}}, k = 1, \ldots, m, s = 1, \ldots, n.$

Proof: Since the edge weight constraints in (5.1) do not depend on the $p_{l|i}$ values, taking $\frac{\partial \tilde{F}}{\partial p_{l|i}} = 0$ yields the same Gibbs distribution as in (4.11).

Substituting $p_{l|i}$ into the Lagrangian $\tilde{F}$ to obtain $\tilde{F}^*$, and setting $\frac{\partial \tilde{F}^*}{\partial Z_{ki}} = 0$ for each k and i, we have $\nu_k Z_{ki}^* = \sum_s \mu_s p_{k|s} P_{si}$. Considering constraints $\sum_{i=1}^n Z_{ki}^* = \sum_{i=1}^n P_{si} = 1, \forall 1 \le k \le m$ and $1 \le s \le n$, we have $Z_{ki}^* = \sum_{s=1}^n q_{s|k} P_{si}$. Note that all resulting entries of $\mathbf{Z}^*$ are nonnegative, every entry of $\mathbf{Q}^*$ is a convex combination of the corresponding column in $\mathbf{Z}^*$, and the resulting $\mathbf{Q}^*$ is a nonnegative stochastic matrix. □

The critical value β_c that leads to phase transition is given by β for which the *second variation of* $\tilde{F}^*$ at $\mathbf{Z}^*$ (see *Theorem* 2) defined by $\Delta^2 \tilde{F}^*(\beta, \mathbf{C}) \stackrel{\triangle}{=} \frac{\mathrm{d}^2}{\mathrm{d}\epsilon^2} \tilde{F}^*(\mathbf{Z}^*(\beta) + \epsilon\mathbf{C})\Big|_{\epsilon=0}$ becomes nonpositive for some $\mathbf{C} \in \mathcal{A}_C$, where $\mathcal{A}_C \stackrel{\triangle}{=} \{\mathbf{C} \in \mathbb{R}^{m\times n} : \sum_k \mathbf{w}(k)^T\mathbf{w}(k) = 1, \sum_{j=1}^n C_{ij} = 0, \text{for all } i\}$ denotes an *admissible perturbation* set. This ensures that when the perturbation $\mathbf{C} \neq 0$, the perturbed

weighting matrix $\hat{\mathbf{Z}}^* = \mathbf{Z}^* + \epsilon\mathbf{C}$ satisfies (5.1), and the second variation is independent of the size of $\mathbf{C}$, yielding the following result.

Theorem 3 *Suppose for some β_0, the matrix $\mathbf{Z}^*(\beta_0)$ in* Theorem 2 *satisfies $\Delta^2\tilde{F}^*(\beta_0, \mathbf{C}) > 0$ for all $\mathbf{C} \in \mathcal{A}_C$ and the number of distinct outgoing vectors $m_0 < m$. The critical $\beta \stackrel{\triangle}{=} \min\{\beta > \beta_0 : \Delta^2\tilde{F}^*(\beta, \mathbf{C}) \not> 0$ for some $\mathbf{C} \in \mathcal{A}_C\}$ satisfies the condition $\min_k \lambda_{min}(\mathbf{\Gamma}_\beta(k)) = 0$, where $\mathbf{\Gamma}_\beta(k)$ is defined in* (5.2).

Proof: The second variation of $\tilde{F}^*$ at $\mathbf{Z}^*$ is given by $\Delta^2\tilde{F}^*(\beta, \mathbf{C}) = \gamma_1(\beta, \mathbf{C}) + \gamma_2^2(\beta, \mathbf{C})$ for $\mathbf{C} \in \mathcal{A}_C$, where $\gamma_1(\beta, \mathbf{C}) = \sum_{k=1}^{m_0} q_k \mathbf{w}(k)^T \Gamma_\beta(k)\mathbf{w}(k) \in \mathbb{R}$, $q_k = \sum_{i=1}^n \mu_i p_{k|i}$, $\mathbf{w}(k)$ is the kth row of $\mathbf{C}$,

$$\begin{aligned} \mathbf{\Gamma}_\beta(k) &= \mathbf{\Lambda}_\beta(k) - \beta\mathbf{C}_\beta(k), \qquad (5.2)\\ \text{where } \mathbf{\Lambda}_\beta(k) &= \text{diag}\left\{\left[\sum_{i=1}^n q_{i|k}\mathbf{p}(i)\right] ./ \left(z^*(k).^2\right)\right\},\\ \text{and } \mathbf{C}_\beta(k) &= \left\{\sum_{i=1}^N q_{i|k}[\mathbf{p}(i)./\mathbf{z}^*(k)][\mathbf{p}(i)./\mathbf{z}^*(k)]^T\right\}. \end{aligned}$$

$\gamma_2^2(\beta, \mathbf{C}) = \beta\sum_{i=1}^n \left(\sum_{k=1}^{m_0} p_{k|i}[\mathbf{p}(i)./\mathbf{z}^*(k)]^T\mathbf{w}(k)\right)^2$ and $[\mathbf{p}(i)./\mathbf{z}(k)]$ denotes the element-wise division (we have assumed $Z_{jk} = 0 \Rightarrow P_{ik} = 0, \forall i, j, k$).

Since $\Delta^2\tilde{F}^*(\beta_0, \mathbf{C}) > 0$, we have

$$\begin{aligned} \Delta^2\tilde{F}^*(\beta, \mathbf{C}) &\geq \gamma_2^2(\beta, \mathbf{C}) + \min_k \{\lambda_{min}(\mathbf{\Gamma}_\beta(k))\}\\ &\geq \min_k \{\lambda_{min}(\mathbf{\Gamma}_\beta(k))\} \geq 0 \end{aligned}$$

for $\beta_0 \leq \beta \leq \beta_c$, $\mathbf{C} \in \mathcal{A}_C$, where we have used the fact that $\Delta^2\tilde{F}^*(\beta, \mathbf{C})$ is continuous in its arguments, and have applied the Rayleigh-Ritz inequality $\mathbf{x}^T\mathbf{\Gamma}_\beta(k)\mathbf{x} \geq \lambda_{\min}(\mathbf{\Gamma}_\beta(k))\|\mathbf{x}\|^2$ for symmetric matrices $\mathbf{\Gamma}_\beta(k)$. Therefore $\Delta^2\tilde{F}^*(\beta_c, \mathbf{C}) = 0$ for some $\mathbf{C}$ only if $\min_k \lambda_{\min}(\mathbf{\Gamma}_\beta(k)) = 0$. Moreover, since $\mathbf{C}_\beta(k)$ and $\mathbf{\Lambda}_\beta(k)$ are insensitive to β, β_c can be approximated by $\lambda_{\min}(\mathbf{C}(k)^{-1}\mathbf{\Lambda}(k))$ whenever $\mathbf{C}_\beta(k)$ is invertible. □

Remark: The perturbation matrix $\mathbf{C}, s.t., \Delta^2\tilde{F}^*(\beta_c, \mathbf{C}) = 0$ is given by $\left[0, \ldots, 0, \mathbf{w}(k_0)^T, -\mathbf{w}(k_0)^T, 0, \ldots, 0\right]^T$, where $\lambda_{\min}(\mathbf{\Gamma}_\beta(k_0)) = \min_k \lambda_{\min}(\mathbf{\Gamma}_\beta(k))$, and $\mathbf{w}(k_0)$ is the

corresponding eigenvector of $\mathbf{\Gamma}_\beta(k_0)$. This choice is possible since $m_0 < m$ and there are at least 2 rows of matrix $\mathbf{Z}^*$ which are equal to $\mathbf{z}^*(k_0)$. The index k_0 identifies the outgoing vector that is replaced by multiple distinct vectors when (4.13) is solved for β that surpasses β_c during the annealing process. Thus there is an increase in the number of distinct outgoing vectors; this process repeats at subsequent critical values as β is increased in the MEP-based algorithm.

5.2 The Analysis of the MEP-Based Algorithm for NCD Markov Chains

A basic case for which any graph clustering approach should work for is a large disconnected graph comprising multiple connected components (henceforth referred to as subgraphs) where there exists no path connecting any two nodes from two subgraphs. Extensive simulations show that our MEP-based algorithm identifies subgraphs. In fact, our approach identifies even weakly interconnected subgraphs in simulations; the edge weights of a weakly interconnected graph are perturbations of a disconnected graph. These studies are particularly relevant to the Markov chains; in identifying decomposable sub-chains from large nearly completely decomposable (NCD) Markov chain (2.2). Note that the Markov chain aggregation method is developed from a data-centric point of view, the formulation of the optimization problem does not reflect dynamic aspects of Markov chains. In this section, we investigate the dynamic properties of the resulting reduced-order Markov chain by providing a comparison between the stationary distribution of the aggregated chain and the aggregated stationary distribution of the original chain.

5.2.1 Sub-chain identification

First we restate the results of the MEP-based clustering for the Markov chains (*Theorem 2*), when the transition matrix $\mathbf{P}$ of the original Markov chain has the NCD structure.

Corollary 1 *For a $n-$state NCD Markov chain consisting with N sub-chains, whose transition matrix $\mathbf{P} = \mathbf{P}^* + \epsilon\mathbf{C}$ is in the form of* (2.2)*, the soft aggregated transition matrix obtained at any*

annealing stage characterized by β is given by

$$\tilde{\mathbf{Q}} = \mathbf{Z}\tilde{\mathbf{\Phi}} \text{ and } \mathbf{Z} = \tilde{\mathbf{R}}^T\mathbf{P}, \tag{5.3}$$

where $\mathbf{Z}$ and $\tilde{\mathbf{R}} \in \mathbb{R}^{N\times n}$ satisfy the implicit equation

$$\mathbf{z}(J) = \sum_{I=1}^{N}\sum_{i=1}^{n_I} \tilde{R}_{i_I J}\mathbf{p}(i_I)$$

with

$$\tilde{R}_{i_I J} = \frac{\tilde{\Phi}_{i_I J}\mu_{i_I}}{\sum_{K=1}^{N}\sum_{k=1}^{n_K}\tilde{\Phi}_{k_K J}\mu_{k_K}} \tag{5.4}$$

and

$$\tilde{\Phi}_{i_I J} = \frac{e^{-\beta\|\mathbf{p}(i_I)-\mathbf{z}(J)\|_2^2}}{\sum_{K=1}^{N}\sum_{k=1}^{n_K} e^{-\beta\|\mathbf{p}(k_K)-\mathbf{z}(J)\|_2^2}}. \tag{5.5}$$

Remark: The elements of matrices $\tilde{\mathbf{\Phi}}$ and $\tilde{\mathbf{R}}$ are respectively given by the association weights $\{p_{J|i_I}\}$ and the posterior association weights $\{q_{i_I|J}\}$ used in *Theorem 2*. We refer to $\tilde{\mathbf{R}}$ as the soft *posterior weight matrix* in the sequel.

Below in Theorem 4 we show that as $\beta \to \infty$ the MEP-based algorithm identifies the blocks of $\mathbf{P}^*$ for NCD Markov chains with $\mathbf{P} = \mathbf{P}^* + \epsilon\mathbf{C}$, when the perturbation matrix $\mathbf{C}$ satisfies *Assumption 1*. We also introduce the notions of *maximum degree of coupling* and the *index of unbalance* for the $\mathbf{C}$ and $\mathbf{P}^*$, and the concept of *separability* for further quantitative derivation.

Assumption 1 *All entries of $\mathbf{C}$ in the diagonal blocks are non-positive, i.e., $C_{i_I k_I} \le 0, \forall i, k, I$.*

Definition 4 (Maximum degree of coupling) *The maximum degree of coupling (see [11]) between any two partitions is given by*

$$\delta \stackrel{\triangle}{=} \max_I \left(\sum_{J\neq I}\sum_{j=1}^{n_J} C_{i_I j_J} \right). \tag{5.6}$$

Definition 5 (Index of unbalance) *For an NCD Markov chain in the form of* (2.2) *with com-*

pletely decomposable component $\mathbf{P}^* = \text{diag}\{\mathbf{P}_I^*\}$*, we define its index of unbalance as*

$$\eta \stackrel{\triangle}{=} \max_I \max_{i,j} \|\mathbf{p}^*(i_I) - \mathbf{p}^*(j_I)\|_1. \tag{5.7}$$

Definition 6 (Separability between subchains) *For an NCD Markov chain with states* $\mathcal{X}$ *and the transition matrix* $\mathbf{P}$ *in the form of* (2.2) *with completely decomposable component* $\mathbf{P}^* = \text{diag}\{\mathbf{P}_I^*\}$*. Let* $\mathcal{X}_I$ *be the states associated with the* I^{th} *diagonal block of* $\mathbf{P}^*$*. We say the* I^{th} *subchain and the* J^{th} *subchain are* separable*, if there exists a separating hyperplane, such that outgoing vectors from every state in* $\mathcal{X}_I$ *are on one side of the plane, and outgoing vectors from every state in* $\mathcal{X}_J$ *are on the other side. (See Figure 5.1 for Illustration.)*

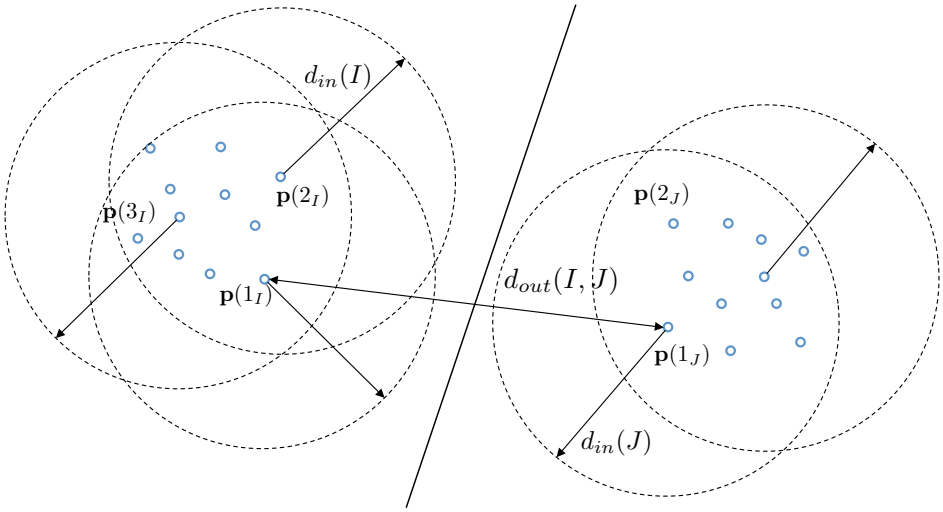


Figure 5.1: Illustration of the notion of separability. Blue circles denote the outgoing vectors of states in $\mathcal{X}_I$ and $\mathcal{X}_J$, $d_{in}(I) = \max_{i,i' \in I} \|\mathbf{p}(i_I) - \mathbf{p}(i'_I)\|_1$ denotes the "radius" of the states in I^{th} subchain. Similarly, $d_{out}(I,J) = \min_{i \in I,\, j \in J} \|\mathbf{p}(i_I) - \mathbf{p}(j_J)\|_1$ represents the closest distance between the sets of outgoing vectors from the I^{th} and J^{th} subchains. In this figure I^{th} and J^{th} subchains are separable, since $\min_{I \neq J} d_{out}(I,J) = d_{out} > 2d_{in} \geq 2\max\{d_{in}(I), d_{in}(J)\}$.

Below we state a result on identification of subchains where we use the concept of separability between subchains.

Theorem 4 *For an NCD Markov chain* $\mathbf{P} = \mathbf{P}^* + \epsilon\mathbf{C}$ *in the form* (2.2)*, with CD nominal transition matrix* $\mathbf{P}^*$*, and perturbation matrix* $\mathbf{C}$ *satisfying* Assumption 1*. If its index of unbalance (defined*

in (5.7)*) for each sub-chain* $\mathbf{P}_I^*$ *of* $\mathbf{P}^*$ *is less than* η*, then the separability between any two subchains is guaranteed when the perturbation* $\epsilon < \frac{1-\eta}{6\delta}$*.*

Proof: We first consider pairwise distances of outgoing vectors from a same sub-chain. For any two states i_I, j_I both associated with the I^{th} sub-chain, the l_1 distance between their outgoing vectors has the following upper bound

$$\begin{aligned} d_{in} &\leq \|\mathbf{p}(i_I) - \mathbf{p}(j_I)\|_1 \\ &\leq \|\mathbf{p}^*(i_I) - \mathbf{p}^*(j_I)\|_1 + \epsilon\|\mathbf{c}(i_I) - \mathbf{c}(j_I)\|_1 \\ &\leq \eta + \epsilon\left[\sum_{k=1}^{n_I}|C_{i_Ik_I} - C_{j_Ik_I}| + \sum_{J\neq I}\sum_{k=1}^{n_J}|C_{i_Ik_J} - C_{j_Ik_J}|\right] \\ &\leq \eta + \sum_{J=1}^{N}\sum_{k=1}^{n_J}\left[|C_{i_Ik_J}| + |C_{j_Ik_J}|\right] \leq \eta + 4\epsilon\delta. \end{aligned}$$

Since by assumption all entries in the diagonal blocks of perturbation matrix $\mathbf{C}$ are non-positive, therefore $\sum_{k=1}^{n_I}|C_{i_Ik_I}| = \sum_{J\neq I}\sum_{j=1}^{n_J} C_{i_Ij_J} \leq \delta$. Similarly, for two states i_I and j_J from different sub-chains, we have

$$\begin{aligned} &\|\mathbf{p}(i_I) - \mathbf{p}(j_J)\|_1 = \|\mathbf{p}^*(i_I) - \mathbf{p}^*(j_J) + \epsilon\mathbf{c}(i_I) - \epsilon\mathbf{c}(j_J)\|_1 \\ = &\underbrace{\sum_{k=1}^{n_I}|P^*_{i_Ik_I} + \epsilon(C_{i_Ik_I} - C_{j_Jk_I})|}_{(a)} + \underbrace{\sum_{k=1}^{n_J}|P^*_{j_Jk_J} + \epsilon(C_{j_Jk_J} - C_{i_Ik_J})|}_{(b)} \\ &+ \underbrace{\epsilon\sum_{N\neq J,I}\sum_{k=1}^{n_N}|C_{i_Ik_N} - C_{j_Jk_N}|}_{(c)}. \end{aligned} \tag{5.8}$$

The term $(a) \geq 1 - 2\delta\epsilon$, since

$$(a) \geq \sum_{k=1}^{n_I}\left(|P^*_{i_Ik_I}| - \epsilon|C_{i_Ik_I} - C_{j_Jk_I}|\right) = 1 - \epsilon\sum_{k=1}^{n_I}\left(|C_{i_Ik_I}| + |C_{j_Jk_I}|\right).$$

Similarly, term $(b) \geq 1 - 2\delta\epsilon$, and term $(c) \geq 0$. Therefore, we conclude the following upper and lower bounds for within, and between partition distances,

$$\begin{aligned} d_{in} &\triangleq \max_{1\leq I\leq N} \max_{1\leq i,j\leq n_I} \|\mathbf{p}(i_I) - \mathbf{p}(j_I)\|_1 \leq 4\delta\epsilon, \text{ and} \\ d_{out} &\triangleq \min_{1\leq I,J\leq N} \min_{1\leq i\leq n_I, 1\leq j\leq n_J} \|\mathbf{p}(i_I) - \mathbf{p}(j_J)\|_1 \geq 2 - 4\delta\epsilon. \end{aligned}$$

The distance of any outgoing vector from subchain I from any outgoing vector from subchain J is at least $2 - 4\delta\epsilon$. Meanwhile, all outgoing vectors from the I^{th} sub-chain are located within $4\delta\epsilon$ distance from each other. The condition that the distance between any two states from the *same* sub-chain is less than the distance between any two *different* sub-chains is satisfied (sufficient condition) when $d_{out} > 2 - 4\delta\epsilon > 4\delta\epsilon + 4\delta\epsilon > 2d_{in}$; that is when $\epsilon < \frac{1-\eta}{6\delta}$. □

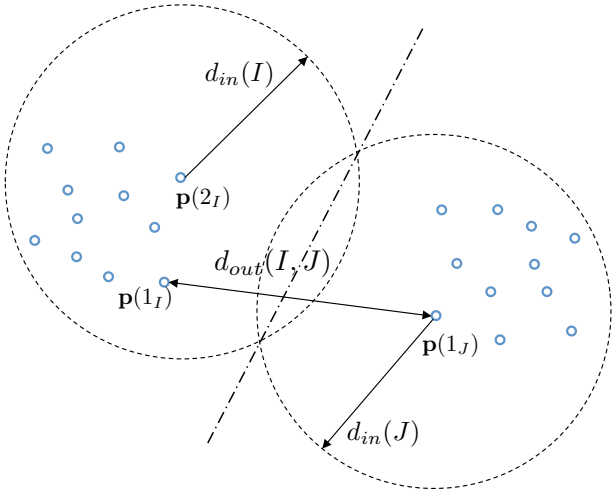


Figure 5.2: The states may still be separable when the strong separable condition $d_{out} = \min_{I,J} d_{out}(I,J) > 2d_{in} = \max_I d_{in}(I)$ fails.

The threshold of perturbation size ϵ given in terms of δ and η provides a condition that guarantees the worst-case separability, since $d_{out} > 2d_{in}$ is a conservative condition for being separable. This becomes clear in Figure 5.2. Next, we discuss some less conservative results by relaxing this condition. Specifically, we characterize the probability of correct classifications; for which, we make the following assumption regarding the distribution of the entries in the perturbation matrix $\mathbf{C}$.

Assumption 2 *All entries in off-diagonal blocks of the perturbation matrix* $\mathbf{C}$ *follow independent,*

identical half-normal distributions *with standard deviation equal to* 1*, that is,* $C_{i_I j_J} = |\hat{C}_{i_I j_J}|, \forall i, j, I \neq J$*, where* $\hat{C}_{i_I j_J} \sim \mathcal{N}(0,1)$*.*

Theorem 5 *Let* $\mathcal{X}_I$ *and* $\mathcal{X}_J$ *be two noncommunicating subchains of a CD Markov chain* $\mathcal{X}$ *with transition matrix* $\mathbf{P}^*$*. The separability between subchains* $\mathcal{X}_I$ *and* $\mathcal{X}_J$ *is bounded below by*

$$\mathbb{P}\left\{(n-n_I)\epsilon^2 Y_1 + \sqrt{2}\epsilon Y_2 + n\epsilon^2\left(\frac{1}{n_I}+\frac{n}{n_J}\right)Y_3 - \sqrt{2}\epsilon Y_4 < \frac{1}{n_I}+\frac{1}{n_J}-\eta^2\right\}, \tag{5.9}$$

in which Y_i *are Chi-squared random variables* $Y_i \sim \chi^2(df_i), i = 1,2,3,4$*, with degree of freedom* $df_1 = df_2 = 2(n-n_J), df_3 = n$ *and* $df_4 = n - n_I - n_J$*.*

Proof: We use the l_2 norm for vector distance in this proof. The probability of disitinguishing identifying the I^{th} and the J^{th} subchains is given by

$$\prod_{i,i' \in I,\ j \in J} \mathbb{P}\{\|\mathbf{p}(i_I) - \mathbf{p}(i'_I)\|_2^2 < \|\mathbf{p}(j_J) - \mathbf{p}(i_I)\|_2^2\}, \tag{5.10}$$

since the row vectors of $\mathbf{C}$ (and thus $\mathbf{P}$) are independent by *Assumption 2*. Let us consider

$$\begin{aligned}
& \|\mathbf{p}(i_I) - \mathbf{p}(i'_I)\|_2^2 \\
= \ & \|\mathbf{p}^*(i_I) + \epsilon\mathbf{c}(i_I) - \mathbf{p}^*(i'_I) - \epsilon\mathbf{c}(i'_I)\|_2^2 \\
= \ & \sum_{k=1}^{n_I} |P^*_{i_I k_I} - P^*_{i'_I k_I} + \epsilon C_{i_I k_I} - \epsilon C_{i'_I k_I}|^2 + \epsilon \sum_{J \neq I} \sum_{k=1}^{n_J} |C_{i_I k_J} - C_{i'_I k_J}|^2 \\
\leq \ & \underbrace{2\sum_{k=1}^{n_I} |P^*_{i_I k_I} - P^*_{i'_I k_I}|^2}_{(a)} + \underbrace{2\epsilon^2 \sum_{k=1}^{n_I} |C_{i_I k_I} - C_{i'_I k_I}|^2}_{(b)} + \underbrace{\epsilon \sum_{J \neq I} \sum_{k=1}^{n_J} |\hat{C}_{i_I k_J} - \hat{C}_{i'_I k_J}|^2}_{(c)},
\end{aligned}$$

where we used the inequality $|a+b|^2 \leq 2|a|^2 + 2|b|^2$ in the last step. The nominal transition matrix $\mathbf{P}^*$ is constant valued, and the entries of $\mathbf{C}$ are random variables. We develop the bounds for individual terms in the summation. Term$(a) = \|\mathbf{p}^*(i_I) - \mathbf{p}^*(i'_I)\|_1^2 \leq 2\eta^2$ since

$$\|\mathbf{a}\|_2^2 \leq \|\mathbf{a}\|_1^2 \leq n\|\mathbf{a}\|_2^2, \quad \forall \mathbf{a} \in \mathbb{R}^n. \tag{5.11}$$

Use (5.11) and note that the entries of $\mathbf{C}$ in the diagonal blocks are assumed to be negative, we have $(b) \leq \sum_{k=1}^{n_I} |C_{i_I k_I}|^2 + \sum_{k=1}^{n_I} |C_{i'_I k_I}|^2 \leq \left[\sum_{k=1}^{n_I} |C_{i_I k_I}|\right]^2 + \left[\sum_{k=1}^{n_I} |C_{i'_I k_I}|\right]^2 = \left[\sum_{J \neq I} \sum_{k=1}^{n_J} |C_{i_I k_I}|\right]^2 + \left[\sum_{J \neq I} \sum_{k=1}^{n_J} |C_{i'_I k_I}|\right]^2 \leq (n - n_I) \left[\sum_{J \neq I} \sum_{k=1}^{n_J} (|\hat{C}_{i_I k_I}|^2 + |\hat{C}_{i'_I k_I}|^2)\right] = (n - n_I) Y_1$, if we define a random variable $Y_1 := \left[\sum_{J \neq I} \sum_{k=1}^{n_J} (|\hat{C}_{i_I k_I}|^2 + |\hat{C}_{i'_I k_I}|^2)\right]$. Since Y_1 is a summation of multiple $(= df_1 = 2(n - n_J))$ independent random variable with standard normal distribution, Y_1 follows a Chi-squared distribution with degree of freedom equal to df_1, or $Y_1 \sim \chi^2(df_1)$.

For the term (c), we first note that $|C_{i_I k_J} - C_{i'_I k_J}| \leq |\hat{C}_{i_I k_J} - \hat{C}_{i'_I k_J}|$ since $C_{i_I k_J} = |\hat{C}_{i_I k_J}|$ for all entries in off-diagonal blocks. Then $\hat{C}_{i_I k_J} - \hat{C}_{i'_I k_J} \sim \mathcal{N}(0, 2)$ being a summation of two independent standard normal random variables. And if we define $Y_2 \stackrel{\triangle}{=} (c)/\sqrt{2}$, we have $Y_2 \sim \chi^2(df_2)$, where $df_2 = n - n_J$.

In summary, we have that $\|\mathbf{p}(i_I) - \mathbf{p}(i'_I)\|_2^2 \leq 2\eta^2 + (n - n_I)\epsilon Y_1 + \sqrt{2}\epsilon^2 Y_2$, where $Y_1 \sim \chi^2(df_1)$ and $Y_2 \sim \chi^2(df_2)$, with $df_1 = 2df_2 = 2(n - n_J)$.

The other norm term in expression (5.10) can be expanded as

$$
\begin{aligned}
& \|\mathbf{p}(i_I) - \mathbf{p}(j_I)\|_2^2 \\
= \; & \|\mathbf{p}^*(i_I) + \epsilon \mathbf{c}(i_I) - \mathbf{p}^*(j_J) - \epsilon \mathbf{c}(j_J)\|_2^2 \\
= \; & \underbrace{\sum_{k=1}^{n_I} |P^*_{i_I k_I} + \epsilon(C_{i_I k_I} - C_{j_J k_I})|^2}_{(e)} + \underbrace{\sum_{k=1}^{n_J} |P^*_{j_J k_J} + \epsilon(C_{j_J k_J} - C_{i_I k_J})|^2}_{(f)} \\
& \underbrace{+\epsilon \sum_{N \neq I,J} \sum_{k=1}^{n_N} |C_{i_I k_N} - C_{j_J k_N}|^2}_{(g)}.
\end{aligned}
$$

We still focus on each term to find their lower bound as follows

$$\begin{aligned}
(e) &\geq \frac{1}{n_I}\left[\sum_{k=1}^{n_I}|P^*_{i_Ik_I}+\epsilon(C_{i_Ik_I}-C_{j_Jk_I})|\right]^2 \\
&\geq \frac{1}{n_I}\left[\sum_{k=1}^{n_I}|P^*_{i_Ik_I}|-\epsilon|C_{i_Ik_I}-C_{j_Jk_I}|\right]^2 \\
&= \frac{1}{n_I}\left[1-\epsilon\sum_{k=1}^{n_I}|C_{i_Ik_I}-C_{j_Jk_I}|\right]^2 \\
&\geq \frac{1}{n_I}\left\{1-\epsilon^2\left[\sum_{k=1}^{n_I}|C_{i_Ik_I}-C_{j_Jk_I}|\right]^2\right\} \\
&= \frac{1}{n_I}-\frac{\epsilon^2}{n_I}\left[\sum_{k=1}^{n_I}|C_{i_Ik_I}|+|C_{j_Jk_I}|\right]^2 \\
&= \frac{1}{n_I}-\frac{\epsilon^2}{n_I}\left[\sum_{J\neq I}\sum_{k=1}^{n_J}|C_{i_Ik_J}|+\sum_{k=1}^{n_I}|C_{j_Jk_I}|]\right]^2 \\
&= \frac{1}{n_I}-\frac{\epsilon^2}{n_I}\left[\sum_{J\neq I}\sum_{k=1}^{n_J}|\hat{C}_{i_Ik_J}|+\sum_{k=1}^{n_I}|\hat{C}_{j_Jk_I}|]\right]^2 \\
&\geq \frac{1}{n_I}-\frac{\epsilon^2}{n_I}n\left[\sum_{J\neq I}\sum_{k=1}^{n_J}|\hat{C}_{i_Ik_J}|^2+\sum_{k=1}^{n_I}|\hat{C}_{j_Jk_I}|^2\right]
\end{aligned}$$

This implies that $(e)\geq\frac{1}{n_I}-\epsilon^2\frac{n}{n_I}Y_3$, where $Y_3\sim\chi^2(df_3)$ with $df_3=n$. (f) has the same distribution as (e) due to symmetry. Finally, the term (g) equals to $\sqrt{2}\epsilon Y_4$, where $Y_4\sim\chi^2(df_4)$ where $df_4=n-n_I-n_J$.

Therefore we have

$$\begin{aligned}
d^2_{in}(I) &\leq 2\eta^2+(n-n_I)\epsilon^2Y_1+\sqrt{2}\epsilon Y_2, \\
d^2_{out}(I,J) &\geq \frac{1}{n_I}+\frac{1}{n_J}-n\epsilon^2\left(\frac{1}{n_I}+\frac{n}{n_J}\right)Y_3+\sqrt{2}\epsilon Y_4),
\end{aligned}$$

where $Y_i\sim\chi^2(df_i)$ are all Chi-squared random variables. So the probability of misclassification between sub-chains I and J can be characterized by the probability of random variables with χ^2 distribution, as (5.9).

Finally, due to the independence assumptions of constructing $\mathbf{P}$, we have: $\mathbb{P}$*(correctly identifying sub-chain I)* $= \prod_{J \neq I} \mathbb{P}$*(correctly identifying sub-chain I from sub-chain J)* $= \prod_{J \neq I} \mathbb{P}(d^2_{out}(I,J) > d^2_{in}(I))$, being the sum of multiple terms that can be estimated. □

Remark: Here we provide a supplemental observation of *Theorem 4* and *Theorem 5*. As annealing progresses, the soft aggregation matrix $\tilde{\mathbf{\Phi}}$ is increasingly close to binary-valued. The aggregation decision in terms of the $\mathbf{\Phi} \in \{0,1\}^{n \times N}$, is obtained from $\tilde{\mathbf{\Phi}}$ at high β limit, given by

$$\Phi_{i_I J} = 1 \iff \lim_{\beta \to \infty} \tilde{\Phi}_{i_I J} > \tilde{\Phi}_{i_I K}, \forall K \neq J.$$

In addition, each of these conditions ($\tilde{\Phi}_{i_I J} > \tilde{\Phi}_{i_I K}$) provides a discriminative boundary that favors associating the ${i_I}^{th}$ state to the J^{th} superstate, rather than the K^{th}. So the states being aggregated into a particular superstate J, is an intersection of all individual discriminative areas.

5.2.2 Properties of long-run dynamics of the aggregated Markov chain

So far we have been mainly focused on formulating and elaborating the problem of finding a simple description for complicated graph models. The aggregation algorithms that we proposed in Section 4.1 and Section 5.1 are data centric and did not consider any dynamical aspects of Markov chains. In this section, we study certain dynamic properties of Markov chains that are preserved by these algorithms. Also for better illustration and putting our algorithms in context, we first discuss a popular aggregation for nearly completely decomposable Markov chains proposed and studied in [62] and [2].

The first result is the aggregation of the NCD Markov chains, proposed by Simon and Ando [62]. In particular, it presents a formulation to calculate the transition matrix of the aggregated chain, which characterizes how the weak interactions among subchains push the Markov process with NCD transition matrix $\mathbf{P}$ towards the steady state. An improved approximation error in the steady state distribution based Simon's aggregation has been proved [11].

Theorem 6 (Variable Aggregation for NCD Markov Chain [11]) *Let* $\mathbf{P}$ *be a transition matrix for a NCD Markov chain with n states; that is, there exists a completely decomposable stochastic matrix* $\mathbf{P}^*$*, such that* $\mathbf{P} = \mathbf{P}^* + \epsilon\mathbf{C}$ *for some perturbation matrix* $\mathbf{C}$ *satisfying constraints* (2.3)*.*

Suppose $\mathbf{P}^* = \text{diag}\{\mathbf{P}_I^*\}_{I=1}^N$ *and the order of* $\mathbf{P}_I^*$ *equal to* n_I *for* $1 \leq I \leq N$. *Let* $\boldsymbol{\pi}_I^*$ *be the stationary distribution vector of* $\mathbf{P}_I^*$, *then the matrix* $\mathbf{Q}$ *with*

$$Q_{IJ}^{\circ} = \sum_{i=1}^{n_I} \pi_{i_I}^* \sum_{j=1}^{n_J} P_{i_I j_J}, \tag{5.12}$$

is a stochastic matrix that defines another Markov chain with N *superstates. Moreover, the stationary distribution of* $\mathbf{Q}^{\circ}$ *satisfies*

$$\|\boldsymbol{\pi}(\mathbf{Q}^{\circ}) - \boldsymbol{\pi}(\mathbf{P})\boldsymbol{\Phi}\|_1 \sim O(\epsilon^2), \tag{5.13}$$

where $\boldsymbol{\Phi}$ *is the aggregation matrix corresponding to* $\mathbf{P}^*$.

Proof: The proof can be found in the reference [11]. Specifically, it has been shown that every entry of the vector $\boldsymbol{\pi}(\mathbf{Q}^{\circ}) - \boldsymbol{\pi}(\mathbf{P})\boldsymbol{\Phi}$ is in the order of ϵ^2. □

Remark: Despite providing a good approximation of the stationary distribution of a large Markov chain, the aggregation (5.12) has an obvious limitation. In order to implement this state aggregation and get $\mathbf{Q}^{\circ}$, it is necessary to know the correct state partition for sub-chains, and the stationary distribution of each $\mathbf{P}_I^*$, both of these are not always available. In fact, most modeling methods yield the $\mathbf{P}$ matrix, and recovering the underlying decomposable subchains is one of the objectives.

In using the MEP-based algorithm to aggregate a large Markov chain, we need to specify a weight vector that denotes the state weights. These weights influence the aggregation results through setting the dissimilarity function $\nu(\mathcal{G}_P, \mathcal{G}_Q)$ between $\mathbf{P}$ and $\mathbf{Q}$ (3.9), (see the following *Lemma 4*). It is desired that, the choice of weights does not require much information of the system. In the following, we study the influence of the state weight vector. Specifically, we estimate an error bound of using the stationary distribution of $\mathbf{Q}$ to approximate the probabilities of the original Markov process stays in each sub-chain at steady-state. In particular, we develop a such bound for the choice of uniform state weights.

Lemma 4 *Consider a NCD Markov chain in* (2.2), *if the MEP based aggregation correctly identifies the underlying sub-chains structure, the transition matrix* $\mathbf{Q} \in \mathbb{R}^{N \times N}$ *of the aggregated chain*

obtained from the MEP-based aggregation algorithm is given by

$$\mathbf{Q} = \mathbf{R}^T \mathbf{P} \mathbf{\Phi},$$

where $\mathbf{\Phi}$ *is the aggregation matrix and* $\mathbf{R} = \mathrm{diag}(\boldsymbol{\mu})\mathbf{\Phi}\left(\mathbf{\Phi}^T \mathrm{diag}(\boldsymbol{\mu})\mathbf{\Phi}\right)^{-1} \in \mathbb{R}_+^{n\times N}$. *Equivalently,*

$$Q_{IJ} = \sum_{i=1}^{n_I} \frac{\mu_{i_I}}{\mu_I} \sum_{j=1}^{n_J} P_{i_I j_J}, \tag{5.14}$$

where $\mu_I = \sum_{i=1}^{n_I} \mu_{i_I}$ *is the weight of the* I^{th} *group of states.*

Proof: Since the resulting transition matrix $\mathbf{Q}$ is obtained when terminating the annealing process when β is large enough, it is equivalent to study the limit of $\tilde{\mathbf{Q}}$ as $\beta \to \infty$. The element-wise limits of $\tilde{\mathbf{\Phi}}$ (5.5) and $\tilde{\mathbf{R}}$ (5.4) are well defined (exist and finite):

$$\lim_{\beta\to\infty} \tilde{\Phi}_{i_I J} \quad = \quad \frac{e^{-\beta\|\mathbf{p}(i_I)-\mathbf{z}(z)\|_2^2 - \|\mathbf{p}(i_I)-\mathbf{z}(J_0)\|_2^2}}{\sum_{K=1}^{N} e^{-\beta\|\mathbf{p}(i_I)-\mathbf{z}(K)\|_2^2 - \|\mathbf{p}(i)-\mathbf{z}(J_0)\|_2^2}}$$

where $\|\mathbf{p}(i_I) - \mathbf{z}(J_0)\|_2 = \min_K \|\mathbf{p}(i_I) - \mathbf{z}(K)\|_2$, and

$$\lim_{\beta\to\infty} \tilde{R}_{i_I J} = \frac{\mu_{i_I} \mathbb{I}_{\|\mathbf{p}(i_I)-\mathbf{z}(J)\|_2 = \min_L \|\mathbf{p}(i_I)-\mathbf{z}(L)\|_2}}{\sum_{K,k:\|\mathbf{p}(k_K)-\mathbf{z}(J)\|_2 = \min_L \|\mathbf{p}(k_K)-\mathbf{z}(L)\|_2} \mu_{k_K}},$$

where $\mathbb{I}$ is the indicator function. On account of the NCD structure, and the identifiability assumption, $\|\mathbf{p}(i_I) - \mathbf{z}(I)\|_2 = \min_K \|\mathbf{p}(i_I) - \mathbf{z}(K)\|_2$ for all I, we can simplify the limits as

$$\lim_{\beta\to\infty} \tilde{\Phi}_{i_I J} \quad = \quad \Phi_{i_I J} = \begin{cases} 1 & \text{if } J = I \\ 0 & \text{else,} \end{cases}$$

$$\text{and } \lim_{\beta\to\infty} \tilde{R}_{i_I J} \quad = \quad R_{i_I J} = \begin{cases} \frac{\mu_{i_I}}{\mu_I} & \text{if } J = I \\ 0 & \text{else.} \end{cases}$$

Also since $\tilde{\mathbf{Q}}$ is continuous in the entries of $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{\Phi}}$, we can conclude that $\lim_{\beta\to\infty} \tilde{\mathbf{Q}} = (\lim_{\beta\to\infty} \tilde{\mathbf{R}}^T)\mathbf{P}(\lim_{\beta\to\infty} \tilde{\mathbf{\Phi}}) = \mathbf{Q}$, or (5.14). □

When $\tilde{\mathbf{\Phi}}$ approaches to the correct subchain partition (when perturbation is small; from extensive simulations and *Theorem 4*), our method uses the initial state weights (a prescribed param-

eter), which eliminates the need for $\mathbf{P}^*$ information. In the sequel, we compare our aggregation formulation (5.14) with $\mathbf{\Phi}$ also resulting from the MEP-based algorithm to the Simon and Ando's aggregation (5.12). In particular, since the aggregated chain reflects the *long-term dynamics* of the Markov chain, analogous to [11], we find an upper bound for $\|\boldsymbol{\pi}(\mathbf{Q}) - \boldsymbol{\pi}(\mathbf{P})\mathbf{\Phi}\|_1$, where $\mathbf{Q}$ is resulting from our method. The result is given in *Theorem 7*. We break up the development of the proof into 5 steps. Development for this theorem requires matrix perturbation theory, the essentials of which we provide in below.

First we start with some notations and required ingredients for the proof.

Suppose a NCD Markov chain is in the form of (2.2). Let $\{\lambda^*(i_I), \boldsymbol{\pi}^*(i_I)^T, \boldsymbol{\upsilon}^*(i_I)\}$ be the i^{th} eigenvalue, left and right eigenvectors of $\mathbf{P}_I^*$. Similarly, let $\{\lambda(l_L), \boldsymbol{\pi}^T(l_L), \boldsymbol{\upsilon}(l_L)\}$ be the ones associated with $\mathbf{P}$, with l_L is an ordering that is compatible with the block structure of $\mathbf{P}^*$. Assume all $\mathbf{P}_I^*$ and $\mathbf{P}$ are diagonalizable, the eigenvalues are in descending order, and all eigenvectors are normalized as

$$\|\boldsymbol{\pi}^*(i_I)\|_1 = \|\boldsymbol{\upsilon}^*(i_I)\|_1 = 1, \ \forall i, I.$$

Since for Markov chains, the largest (and unrepeated) eigenvalue is 1, therefore for each I, $\lambda^*(1_I) = 1, \boldsymbol{\upsilon}^*(1_I) = \frac{1}{n_I}\mathbf{1}_{n_I}$ and $\boldsymbol{\pi}^*(1_I)^T$ is the stationary distribution of the I^{th} sub-chain, satisfying $\boldsymbol{\pi}^*(1_I)^T\mathbf{P}_I^* = \boldsymbol{\pi}^*(1_I)^T$. Similar relations hold for $\lambda(1_1), \boldsymbol{\upsilon}(1_1)$ and $\boldsymbol{\pi}(1_1)$. For the aggregated chain $\mathbf{Q} \in \mathbb{R}^{N\times N}$, we denote its eigenvalue and left eigenvectors by $\{\kappa(L), \boldsymbol{\alpha}^*(L)^T\}$, for $L = 1, \ldots, N$.

Lemma 5 *Let $\{\lambda(l_L), \boldsymbol{\theta}(l_L)^T\}$ be the pair of eigenvalue and left eigenvector of the matrix $\mathbf{G}^*\mathbf{P}(\mathbf{G}^*)^{-1}$, where $\mathbf{G}^* = \mathrm{diag}(\mathbf{G}_I^*)$, and $\mathbf{G}_I^* = \begin{bmatrix} \boldsymbol{\pi}^*(1_I)^T \\ \vdots \\ \boldsymbol{\pi}^*(n_{II})^T \end{bmatrix}$. Then the vector $\boldsymbol{\theta}_1(l_L)^T \stackrel{\triangle}{=} [\theta_{1_1}(l_L), \ldots, \theta_{1_N}(l_L)]$ satisfies $\boldsymbol{\theta}_1(l_L)^T = \boldsymbol{\pi}^T(l_L)\mathbf{\Phi}$. In particular, when $L = 1$ we have*

$$\boldsymbol{\theta}_1(1_1)^T = \boldsymbol{\pi}(\mathbf{P})^T\mathbf{\Phi}, \tag{5.15}$$

where $\boldsymbol{\pi}(\mathbf{P})^T$ is the stationary distribution of $\mathbf{P}$.

Proof: By definition, $\mathbf{G}_I^*$ consists of the left eigenvectors of $\mathbf{P}_I^*$, and the corresponding eigenvalues are $\lambda^*(i_I)$, Therefore $\mathbf{G}_I\mathbf{P}_I^*\mathbf{G}_I^{-1} = \mathbf{\Lambda}_I^*$. Note that the eigenvectors associated with different eigen-

values are orthogonal, i.e., $\boldsymbol{\pi}^*(i_I)^T\boldsymbol{\upsilon}^*(i'_I) = 0, \forall i \neq i'$. If we further define $s^*(i_I) = \boldsymbol{\pi}^*(i_I)^T\boldsymbol{\upsilon}^*(i_I)$, then $(\mathbf{G}^*)^{-1}$ can be written as

$$(\mathbf{G}_I^*)^{-1} = \left[\frac{\boldsymbol{\upsilon}^*(1_I)}{s^*(1_I)}, \ldots, \frac{\boldsymbol{\upsilon}^*(n_{II})}{s^*(n_{II})}\right].$$

We then consider the relation between the eigenvectors of $\mathbf{Q}$ and the aggregated stationary distribution $\boldsymbol{\pi}(\mathbf{P})^T\boldsymbol{\Phi}$. Apply coordinate transformation $\mathbf{G}$ (invertible) on the transition matrix $\mathbf{P}$,

$$\mathbf{G}^*\mathbf{P}(\mathbf{G}^*)^{-1} = \mathbf{G}^*(\mathbf{P}^* + \epsilon\mathbf{C})(\mathbf{G}^*)^{-1} = \boldsymbol{\Lambda}^* + \epsilon\mathbf{G}^*\mathbf{C}(\mathbf{G}^*)^{-1}. \tag{5.16}$$

When ϵ is small, the second term in the right hand side is small, which implies $\mathbf{G}^*$ (consisting of the left eigenvectors of $\mathbf{P}^*$) is close to the matrix consisting of left eigenvectors of $\mathbf{P}$, and this further implies the closeness of their stationary distributions $\boldsymbol{\pi}^*(1_1)^T$ and $\boldsymbol{\pi}(1_1)^T$.

Since $\{\lambda(l_L), \boldsymbol{\theta}(l_L)^T\}$ is the pair of eigenvalue and left eigenvector of $\mathbf{G}^*\mathbf{P}(\mathbf{G}^*)^{-1}$, and $\boldsymbol{\pi}(l_L)$ is the left eigenvector of $\mathbf{P}$, according to similarity transformation, we have

$$\boldsymbol{\theta}(l_L)^T = \boldsymbol{\pi}(l_L)^T(\mathbf{G}^*)^{-1}. \tag{5.17}$$

Consider the block diagonal structure of $\mathbf{G}^*$. The first entry of each block I in $\boldsymbol{\theta}(l_L)$ is given by

$$\theta_{1_I}(l_L) = \frac{1}{s^*(1_I)}\boldsymbol{\pi}_I(l_L)^T\boldsymbol{\upsilon}^*(1_I) = \frac{1}{s^*(1_I)n_I}\boldsymbol{\pi}_I(l_L)^T\mathbf{1}_{n_I} = \sum_{i=1}^{n_I}\pi_{i_I}(l_L), \tag{5.18}$$

where $\boldsymbol{\pi}_I(1_I)$ is the sub-vector of $\boldsymbol{\pi}(1_I)$ with entries in the I^{th} subset. In the sequel, we sometimes drop the brackets (1_1) when referring to the eigenvectors associated with the largest eigenvalue. The relation in (5.18) can be rewritten as

$$\boldsymbol{\theta}_1^T(l_L) \stackrel{\triangle}{=} [\theta_{1_1}(l_L), \ldots, \theta_{1_N}(l_L)] = \boldsymbol{\pi}(l_L)^T\boldsymbol{\Phi}.$$

By setting $l = L = 1$ we have the relation (5.15). The term on the right hand side is the aggregated stationary distribution of $\mathbf{P}$, i.e., grouping $\boldsymbol{\pi}(\mathbf{P})^T$ according to the true sub-chain partition. This is the quantity we want to approximate by using the aggregated chain $\mathbf{Q}$ from the MEP-based

method. □

The additive perturbation $\mathbf{C}$ causes the transitions from a state in one sub-chain to another state in a different sub-chain, and this makes the stationary distribution of $\mathbf{P}$ deviate from that of $\mathbf{P}^*$. We now look at influence of the second term of (5.16).

Lemma 6 *The vector $\boldsymbol{\theta}(l_L)^T$ defined in* Lemma 5 *satisfies linear equation $\mathbf{A}\boldsymbol{\theta}(l_L) = \epsilon\mathbf{b}$, where*

$$A_{IJ} = \begin{cases} 1-\lambda(l_L) & \text{if } J = I \\ \epsilon n_I \sigma_{1_I 1_J} & \text{else} \end{cases}, \quad \text{and } b_I = -n_I \sum_{J=1}^{N}\sum_{j=2}^{n_J} \theta_{j_J}(l_L)\sigma_{i_J i_I}$$

and

$$\sigma_{i_I j_J} \stackrel{\triangle}{=} \boldsymbol{\pi}^*(i_I)^T \mathbf{C}_{IJ} \boldsymbol{\upsilon}^*(j_J),$$

in which $\mathbf{C}_{IJ}$ is the IJ^{th} block of matrix $\mathbf{C}$.

Proof: Since $\boldsymbol{\theta}(l_L)^T$ and $\lambda(l_L)$ satisfy

$$\begin{aligned} \lambda(l_L)\boldsymbol{\theta}(l_L)^T &= \boldsymbol{\theta}(l_L)^T \mathbf{G}^*(\mathbf{P}^* + \epsilon\mathbf{C})(\mathbf{G}^*)^{-1} \\ &= \boldsymbol{\theta}(l_L)^T \boldsymbol{\Lambda}^* + \epsilon\boldsymbol{\theta}(l_L)^T \boldsymbol{\Sigma}\, \mathrm{diag}(\frac{1}{s^*(i_I)}), \end{aligned}$$

where $\boldsymbol{\Sigma} = [\sigma_{i_I j_J}]$. The $i_I{}^{th}$ element of vector $\lambda(l_L)\boldsymbol{\theta}(l_L)^T$ is

$$\lambda(l_L)\theta_{i_I}(l_L) = \lambda^*(i_I)\theta_{i_I}(l_L) + \epsilon \sum_{J=1}^{N}\sum_{j=1}^{n_J} \theta_{j_J}(l_L)\sigma_{i_J i_I}\frac{1}{s^*(i_I)},$$

which gives

$$[\lambda(l_L) - \lambda^*(i_I)]\,\theta_{i_I}(l_L) = \epsilon \sum_{J=1}^{N}\sum_{j=1}^{n_J} \theta_{j_J}(l_L)\sigma_{i_J i_I}\frac{1}{s^*(i_I)}.$$

Note that the largest eigenvalue of each $\mathbf{P}_I^*$ equals to 1. Fix $i = 1$ in above equation, we have $\lambda^*(1_I) = 1$ for all I, $s^*(1_I) = \frac{1}{n_I}$, for all I, and

$$[1-\lambda(l_L)]\,\theta_{1_I}(l_L) + \epsilon n_I \sum_{J=1}^{N} \theta_{1_J}(l_L)\sigma_{1_J 1_I} = -\epsilon n_I \sum_{J=1}^{N}\sum_{j=2}^{n_J} \theta_{j_J}(l_L)\sigma_{i_J i_I}. \tag{5.19}$$

Note that for any $j \neq 1$, $\theta_{j_J}(l_L)$ is in the order of ϵ (from matrix perturbation theory, [11, 68]), so

each term of the summation in the right hand side of (5.19) is of order ϵ^2. Let I change from 1 to N, we have a set of linear equations for entries of vector $\boldsymbol{\theta}_1(l_L)$, with $\{\sigma_{1_J 1_I}\}$'s being coefficients, and the matrix form expression is given by $\mathbf{A}\boldsymbol{\theta}(l_L) = \epsilon\mathbf{b}$, as required. Moreover, every entry of the vector $\mathbf{b}$ is in the order of ϵ. □

Since we are interested in the situation when lacking detailed system information to guide the selection of state weights, we try to develop some error bounds for a general weight vector, for example, the uniform weights $\mu_{i_I} = \frac{1}{n}$ for all i, I.

Lemma 7 *Every element of the vector* $\boldsymbol{\theta}_1(1_L)^T\mathbf{Q} - \lambda(1_L)\boldsymbol{\theta}_1(1_L)^T \sim O(\epsilon)$ *for all* $1 \le L \le N$, *in which* $\mathbf{Q}$ *is the aggregated transition matrix resulting from* (5.14) *with uniform state weights.*

Proof: We consider the linear equations (5.19) in *Lemma 7* for $l = 1$, the expression for these coefficients are given by

$$\begin{aligned}
\sigma_{1_J 1_I} &= \boldsymbol{\pi}^*(1_J)^T \mathbf{C}_{JI} \boldsymbol{\upsilon}^*(1_I) \\
&= \frac{1}{n_I} \sum_{j=1}^{n_J} \pi^*_{j_J}(1_J) \sum_{i=1}^{n_I} C_{j_J i_I} \\
&= \frac{1}{\epsilon n_I} \sum_{j=1}^{n_J} \pi^*_{j_J}(1_J) \sum_{i=1}^{n_I} P_{j_J i_I} \\
&= \frac{1}{\epsilon n_I} \sum_{j=1}^{n_J} \left[\frac{\mu_{j_J}}{\mu_J} + \pi^*_{j_J}(1_J) - \frac{\mu_{j_J}}{\mu_J} \right] \sum_{i=1}^{n_I} P_{j_J i_I} \\
&= \frac{1}{\epsilon n_I} Q_{JI} + \frac{1}{\epsilon n_I} \sum_{j=1}^{n_J} \left[\pi^*_{j_J}(1_J) - \frac{\mu_{j_J}}{\mu_J} \right] \sum_{i=1}^{n_I} P_{j_J i_I},
\end{aligned} \tag{5.20}$$

by noting $\mathbf{C}_{JI} = \frac{1}{\epsilon}\mathbf{P}_{JI}$ for $J \neq I$. Also since for all j_J,

$$\sum_{I=1}^{N} n_I \sigma_{1_J 1_I} = \boldsymbol{\pi}^*(1_J)^T \begin{bmatrix} \sum_{i=1}^{n_I} C_{1_J i_I} \\ \vdots \\ \sum_{i=1}^{n_J} C_{n_J J i_I} \end{bmatrix},$$

we have

$$\sigma_{1_J 1_J} = -\frac{1}{\epsilon n_J} \sum_{J \neq I} \left[Q_{JI} + \sum_{j=1}^{n_J} \left[\pi^*_{j_J}(1_J) - \frac{\mu_{j_J}}{\mu_J} \right] \sum_{i=1}^{n_I} P_{j_J i_I} \right]. \tag{5.21}$$

Substitute (5.20) and (5.21) into (5.19), express $[1-\lambda(1_L)]\theta_{1_I}(1_L)$ using the remaining terms. The I^{th} element of $\boldsymbol{\theta}_1(1_L)^T[\mathbf{Q}-\lambda(1_L)\mathbf{I}_N]$ is given by

$$\begin{aligned}
&\sum_{j=1}^{N}\theta_{1_I}(1_L)Q_{JI}-\lambda(l_L)\theta_{1_I}(1_L) \qquad (5.22)\\
=\;& [1-\lambda(1_L)]\theta_{1_I}(1_L)-\sum_{J\neq I}Q_{IJ}\theta_{1_I}(1_L)+\sum_{J\neq I}Q_{JI}\theta_{1_J}(1_L)\\
=\;& \underbrace{\theta_{1_I}(1_L)\sum_{J\neq I}\left[\sum_{i=1}^{n_I}\left[\pi^*_{i_I}(1_L)-\frac{\mu_{i_I}}{\mu_I}\right]\sum_{j=1}^{n_I}P_{i_Ij_J}\right]}_{(a_I)}\\
&+\underbrace{\sum_{J\neq I}\theta_{1_J}(1_L)\left[\sum_{j=1}^{n_J}\left[\pi^*_{j_J}(1_L)-\frac{\mu_{j_J}}{\mu_J}\right]\sum_{i=1}^{n_I}P_{j_Ji_I}\right]}_{(b_I)}-\underbrace{\epsilon n_I\sum_{J=1}^{N}\sum_{j=2}^{n_J}\theta_{j_J}(1_L)\sigma_{j_Ji_I}}_{(c_I)}.
\end{aligned}$$

Term (c_I) is at the order of ϵ^2 since $\theta_{j_J}(l_L)$'s are all in ϵ for $j\neq 1$. We will next show that the summation of the first two terms dominates the error, in general, at the order of ϵ.

We expand each term in (5.22) when the state weights are uniform, i.e., $\frac{\mu_{j_J}}{\mu_J}=\frac{1}{n_J}$.

$$\begin{aligned}
|a_I| &\leq \theta_{1_I}\sum_{J\neq I}\sum_{i=1}^{n_I}\left[|\pi^*_{i_I}-\frac{1}{n_I}|\sum_{j=1}^{n_J}P_{i_Ij_J}\right]\\
&= \theta_{1_I}\sum_{i=1}^{n_I}\left[|\pi^*_{i_I}-\frac{1}{n_I}|\sum_{J\neq I}\sum_{j=1}^{n_J}P_{i_Ij_J}\right]\\
&\overset{(i)}{\leq} \theta_{1_I}\sum_{i=1}^{n_I}|\pi^*_{i_I}-\frac{1}{n_I}|\epsilon\\
&\overset{(ii)}{\leq} 2\epsilon\theta_{1_I}\left[1-\frac{1}{n_I}\right].
\end{aligned}$$

The inequality (i) is from the maximum degree of coupling (5.6), and (ii) is shown in *Lemma* 10. We also have

$$\sum_{I=1}^{n}|a_I|\leq 2\epsilon,$$

since $\|\boldsymbol{\theta}_1\|_1<1$, due to the eigenvector normalization. We also bound term $\sum_{I=1}^{N}(b_I)$ by 2ϵ as

below

$$\begin{aligned}
\sum_{I=1}^{N}|b_I| &\leq \sum_{I=1}^{N}\sum_{J\neq I}\theta_{1_J}\sum_{j=1}^{n_J}\left[(\pi^*_{j_J}-\frac{1}{n_J})\sum_{i=1}^{n_I}P_{j_J i_I}\right] \\
&= \sum_{J\neq I}\theta_{1_J}\sum_{j=1}^{n_J}\left[(\pi^*_{j_J}-\frac{1}{n_J})\sum_{I=1}^{N}\sum_{i=1}^{n_I}P_{j_J i_I}\right] \\
&\leq \sum_{J\neq I}\theta_{1_J}\sum_{j=1}^{n_J}\epsilon|\pi^*_{j_J}-\frac{1}{n_J}| \\
&\leq 2\epsilon\sum_{J\neq I}\theta_{1_J}\left[1-\frac{1}{n_J}\right]\leq 2\epsilon.
\end{aligned}$$

Overall, the l_1 norm of vector $\boldsymbol{\theta}_1(1_L)^T[\mathbf{Q}-\lambda(1_L)\mathbf{I}_N]=\sum_{I=1}^{n}|a_I+b_I-c_I|\leq\sum_{I=1}^{n}|a_I|+\sum_{I=1}^{n}|b_I|+\sum_{I=1}^{n}|c_I|\leq 4\epsilon+O(\epsilon^2)$, which implies the l_1 norm of the vector is at the order of 4ϵ. □

Remark: If we select state weight $\boldsymbol{\mu}$ such that

$$\frac{\mu_{j_J}}{\mu_J}=\pi^*_{j_J}(1_1),\quad \forall j,J,$$

then term (a) and (b) both equal to zero and the error (5.22) is only from the term (c). This is essentially the aggregation proposed by Simon et. al. in [62], whose error has been established as ϵ^2 in [11].

Our goal is to find a bound for vector $\boldsymbol{\theta}_1(1_L)^T[\mathbf{Q}-\kappa(L)\mathbf{I}_N]$, with $\mathbf{Q}$'s eigenvalue κ. Here we make a technical assumption on the spectrum of $\mathbf{Q}$.

Assumption 3 *Assume the eigenvalue of* $\mathbf{Q}$ *are not all together, specifically, assume there exists a positive number* Δ*, such that* $\max_{1\leq K,K'\leq N}|\kappa(K)-\kappa(K')|\geq\Delta$.

Theorem 7 *For a NCD Markov chain* $\mathbf{P}=\mathbf{P}^*+\epsilon\mathbf{C}$ *as* (2.2)*, let* $\boldsymbol{\pi}(\mathbf{P})$ *be the stationary distribution of* $\mathbf{P}$*. Further assume* $\mathbf{P}^*_I$ *is aperiodic and irreducible for* $I=1,\ldots,N$*, and* $\boldsymbol{\pi}^*_I=[\pi_{1_I},\ldots,\pi_{n_{II}}]$ *is its unique stationary distribution. Let* $\mathbf{Q}$ *be the transition matrix for the aggregated Markov chain resulting from choosing uniform state weights* $\mu_i=\frac{1}{n}$ *for all* i*, and* $\boldsymbol{\pi}(\mathbf{Q})$ *be its stationary distribution of* $\mathbf{Q}$*. Then*

$$\|\boldsymbol{\pi}(\mathbf{Q})-\boldsymbol{\pi}(\mathbf{P})\boldsymbol{\Phi}\|_1\sim O(\epsilon).$$

Proof: Equation (5.22) indicates that each element of vector $\boldsymbol{\theta}_1(1_L)^T[\mathbf{Q} - \lambda(1_L)\mathbf{I}_N]$ is in ϵ. However, from *Lemma 5*, $\boldsymbol{\theta}_1(1_L)^T = \boldsymbol{\pi}^T(1_L)\boldsymbol{\Phi}$ must contain elements exceeding ϵ. Without less of generality, suppose it is the first entry, $\theta_{1_1}(1_L)$. Then to guarantee $\boldsymbol{\theta}_1(1_L)^T[\mathbf{Q} - \lambda(1_L)\mathbf{I}_N]$ be in ϵ element-wise, all elements in the first row of matrix $[\mathbf{Q} - \lambda(1_L)\mathbf{I}_N]$ are in the order of ϵ. Thus we have $\det[\mathbf{Q} - \lambda(1_L)\mathbf{I}_N] \sim \epsilon$, which differs from the characteristic equation of $\mathbf{Q}$ by ϵ. Therefore, the roots of these two equations differ at ϵ, implying that for each L there exists an eigenvalue $\kappa(L')$ of $\mathbf{Q}$ (these eigenvalues are not ordered in a particular way) such that $|\kappa(L') - \lambda(1_L)| \sim \epsilon$ for some index L'. In summary, for each L, there exist some $\kappa(L')$ closed to $\lambda(1_L)$.

Note that $\kappa(L')$ is the $(L')^{\text{th}}$ eigenvalue of $\mathbf{Q}$, let $\boldsymbol{\alpha}(L')^T$ be the corresponding left eigenvector, that is, $\kappa(L')\boldsymbol{\alpha}(L')^T = \boldsymbol{\alpha}(L')^T\mathbf{Q}$. We consider the following vector

$$[\boldsymbol{\theta}_1(1_L)^T - \boldsymbol{\alpha}(L')^T][\mathbf{Q} - \kappa(L')I_N] = \boldsymbol{\theta}_1(1_L)^T[\mathbf{Q} - \lambda(1_L)\mathbf{I}_N] + [\lambda(1_L) - \kappa(L')]\boldsymbol{\theta}_1(1_L)^T\mathbf{I}_N, \quad (5.23)$$

which is of order ϵ, since both terms on the right hand side are in order of ϵ (since $\det[\mathbf{Q}-\lambda(1_L)\mathbf{I}_N] \sim \epsilon$ and $|\kappa(L') - \lambda(1_L)| \sim \epsilon$). On other hand, the K^{th} entry of the vector on the left hand side is given by, $[\boldsymbol{\theta}_1(1_L)^T - \boldsymbol{\alpha}(L')^T]\boldsymbol{\xi}_K$, where $\boldsymbol{\xi}_K$ is the K^{th} column of $[\mathbf{Q} - \kappa(L')\mathbf{I}_N]$. A necessary and sufficient condition to guarantee $[\boldsymbol{\theta}_1(1_L)^T - \boldsymbol{\alpha}(L')^T][\mathbf{Q} - \kappa(L')I_N] \sim O(\epsilon)$ for $L = 1$ is that $|\boldsymbol{\theta}_1(1_L) - \boldsymbol{\alpha}(L')|^T$ being at order of ϵ, element-wise. Let the index $L'' = \arg\max_K \kappa(K)$, i.e., $|\kappa(L'') - \lambda(1_1)| \sim \epsilon$, then $\boldsymbol{\xi}_K$ is beyond ϵ for all $K \neq L''$. Therefore, $[\boldsymbol{\theta}_1(1_1)^T - \boldsymbol{\alpha}(L')^T]_K \sim \epsilon$ for all $K \neq L''$. Moreover, since $\|\boldsymbol{\theta}_1(1_L)\|_1 = \|\boldsymbol{\alpha}(L')\|_1 = 1$, the difference of the L'' entry is also in ϵ. These give the necessity; and sufficiency is straightforward. Rewrite $|\boldsymbol{\theta}_1(1_L) - \boldsymbol{\alpha}(L')| \sim \epsilon$ for $L = 1$, we have $\|\boldsymbol{\pi}(\mathbf{Q}) - \boldsymbol{\pi}(\mathbf{P})\boldsymbol{\Phi}\|_1 \sim O(\epsilon)$ as needed. □

Corollary 2 *If all $\mathbf{P}_I^*$ are doubly stochastic, then aggregation result by taking the uniform state weights attain a ϵ^2 error in the group stationary distribution as follows,*

$$\|\boldsymbol{\pi}(\mathbf{Q}) - \boldsymbol{\pi}(\mathbf{P})\boldsymbol{\Phi}\|_1 \sim O(\epsilon^2).$$

Proof: If $\mathbf{P}_I^*$ is doubly stochastic, then $\frac{1}{n_I}\mathbf{1}^T\mathbf{P}_I^* = \frac{1}{n_I}\mathbf{1}^T$. This implies $\boldsymbol{\pi}_{i_I}^*(1_1) = \frac{1}{n_I}$, for all i, I. Then terms (a) and (b) in (5.22) become zero, the error is given by the term (c), the error in characteristic equation (the same quantity as in *Lemma 7*) becomes $O(\epsilon^2)$. In fact, for doubly stochastic matrix which stationary distribution is also uniform, the case reduces to that of [62]. As in *Theorem 6* an at ϵ^2 error has been proved in [11]. □

Finally we make a remark to close our analysis on Markov chain aggregation. In all analysis in Section 5.2.1 and Section 5.2.2, we assume that the nominal completely decomposable part of the NCD Markov chain, $\mathbf{P}^*$, is in block-diagonal structure. But all results hold when $\mathbf{P} = \bar{\mathbf{P}}^* + \epsilon\mathbf{C}$, in which $\bar{\mathbf{P}}^*$ can be is a permutation of $\mathbf{P}^*$. The reason is that both the state aggregation $\boldsymbol{\Phi}$ and the outgoing vectors of the composite graphs $\mathbf{z}(j)$'s, are fully determined by the pairwise distance of original outgoing vectors. This becomes clear if we recall the clustering formulation (4.9) for a particular *softness* level (set by β), and how the soft association weights $\tilde{\boldsymbol{\Phi}}$ and the representative outgoing vectors $\mathbf{y}$'s evolve during the annealing process.

Proposition 5 *Let the vector distance $d(\cdot,\cdot)$ in (3.8) defined on $\mathbb{R}^n$ satisfy $d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^n d(\mathbf{x}_i,\mathbf{y}_i)$ for all $\mathbf{x},\mathbf{y} \in \mathbb{R}^n$, where $\mathbf{x}_i$ is the projection of vector $\mathbf{x}$ on the i^{th} coordinate, i.e., $\mathbf{x}_i = x_i\mathbf{e}_i$. The result of problem (4.9) obtained by applying the MEP-based algorithm to minimize the soft $\phi-$dissimilarity function is unaffected by state permutations.*

Proof: Note that for any β value during the annealing process, the association weight matrix $\tilde{\boldsymbol{\Phi}}$ is determined only by the pairwise distance between outgoing vectors (5.5). The resulting transition matrix $\tilde{\mathbf{Q}}$ for the aggregated Markov chain (5.3) is determined by $\tilde{\boldsymbol{\Phi}}$ and the chosen state weights $\boldsymbol{\mu}$, which is also unaffected by permutation. Therefore for the proposition to hold true, it is sufficient to show that the state permutation does not affect these pairwise distances.

Let $\phi : \mathcal{N} \to \mathcal{N}$ be an arbitrary partition function between two equally-sized index sets, that is, ϕ defines a reordering of the indices. Then the corresponding aggregation matrix $\boldsymbol{\Phi}$ is a permutation matrix, given by $\boldsymbol{\Phi} = \left[\mathbf{e}_{\phi(1)}, \cdots, \mathbf{e}_{\phi(n)}\right]$. The transition matrix after state reordering is given by

$$\bar{\mathbf{P}} = \boldsymbol{\Phi}^T\mathbf{P}\boldsymbol{\Phi} = \begin{bmatrix} P_{\phi(1)\phi(1)} & \cdots & P_{\phi(1)\phi(n)} \\ \vdots & \ddots & \vdots \\ P_{\phi(n)\phi(1)} & \cdots & P_{\phi(n)\phi(n)} \end{bmatrix}.$$

The state clustering is determined by row comparison of $\mathbf{P}$, in particular, by the pairwise distance $d(\mathbf{p}(i), \mathbf{p}(j))$ for all i and j. If $d(\cdot, \cdot)$ is the l_1 norm, and let $i' = \phi^{-1}(i)$, $j' = \phi^{-1}(j)$, $k' = \phi^{-1}(k)$, then

$$\begin{aligned}\|\mathbf{p}(i) - \mathbf{p}(j)\|_1 &\overset{(a)}{=} \sum_{k=1}^{n} |P_{ik} - P_{jk}| = \sum_{k'=1}^{n} |P_{\phi(i')\phi(k')} - P_{\phi(j')\phi(k')}| \\ &= \sum_{k=1}^{n} |P_{\phi(i')k} - P_{\phi(j')k}| = \|\mathbf{p}(\phi(i)) - \mathbf{p}(\phi(j))\|_1 \\ &= \|\bar{\mathbf{p}}(i) - \bar{\mathbf{p}}(j)\|_1.\end{aligned}$$

This implies that the pairwise distances between states are independent of state reordering, if we select a vector distance function that is additive over all dimensions as assumed (any $p-$norm, $1 \leq p < \infty$ has this property). Moreover, this is also hold by l_∞ since step (a) is true for the vector infinity norm. Hence, all entries of the association weights, $\{\tilde{\Phi}_{ij}\}$ are also independent of reordering, at each value of β. Since all entries $\tilde{\Phi}_{ij}$ of the association weight matrix are continuous functions of the outgoing vectors $\mathbf{p}(i)$, the final clustering assignment, found as $\beta \to \infty$, will also be unaffected by permutations. □

5.3 Simulation

5.3.1 An illustrative example of Markov chain aggregation

We first discuss aggregating a small Markov chain $\mathcal{X}$ with 3 states solely to demonstrate our framework and the annealing process. Let the transition matrix of $\mathcal{X}$ be

$$\mathbf{P} = \begin{bmatrix} 0.97 & 0.01 & 0.02 \\ 0.02 & 0.48 & 0.50 \\ 0.01 & 0.75 & 0.24 \end{bmatrix},$$

and the state weights $\{\mu_i\}$ given by $[0.3471, 0.3883, 0.2646]$, which equals to the limiting distribution of $\mathcal{X}$.

We use the results from Section 5.1 to obtain an aggregated Markov chain $\mathcal{Y}$ with 2 superstates. For small β $(= 0.001)$, the association weights are identical; that is, $p_{j|i} = 0.5$ and therefore $Z_{ij} = \frac{1}{3}$

for $i = 1, 2, 3$ and $j = 1, 2$. The results for β values just beyond the critical $\beta_c = 1.0837$ and a very high value ($\beta = 54.2540$) are shown below:

β	$\mathbf{Z}^T$	$\tilde{\mathbf{\Phi}}$	$\mathbf{Q}$
1.084	$\begin{bmatrix} 0.0243 & 0.9539 \\ 0.5845 & 0.0194 \\ 0.3912 & 0.0267 \end{bmatrix}$	$\begin{bmatrix} 0.0165 & 0.9835 \\ 0.9898 & 0.0102 \\ 0.9928 & 0.0072 \end{bmatrix}$	$\begin{bmatrix} 0.9673 & 0.0327 \\ 0.0614 & 0.9386 \end{bmatrix}$
54.25	$\begin{bmatrix} 0.0159 & 0.9700 \\ 0.5894 & 0.0100 \\ 0.3946 & 0.0200 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 1.0000 \\ 1.0000 & 0.0000 \\ 1.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.9841 & 0.0159 \\ 0.0300 & 0.9700 \end{bmatrix}$

Note that $\tilde{\mathbf{\Phi}}$ in both cases indicates that one superstate is representative of the first state in $\mathcal{X}$ while the other superstate represents the remaining two states; note the large β value gives a hard partition. This example also has been considered in [13]; we obtain the same partition result, and in addition provide the superstate weights reflecting the proportion of states being aggregated.

5.3.2 A NCD Markov chain with index of balance $\eta = 0$

This example is to demonstrate the analysis in Section 5.2, *Lemma 4* when $\eta = 0$. In particular, we show that for a NCD Markov chain $\mathbf{P} = \mathbf{P}^* + \epsilon\mathbf{C}$, satisfying *Assumption 1*, the effectiveness of the MEP-based algorithm in identifying the sub-chains. We design the example according to these two assumptions, that is, the completely decomposable component $\mathbf{P}^*$ of $\mathbf{P}$ consists of repeated rows. However, after adding perturbation, the transition matrix $\mathbf{P}$ is less restrictive. In this example, we select

$$\mathbf{P}^* = \begin{bmatrix} 0.7 & 0.3 & 0 & 0 \\ 0.7 & 0.3 & 0 & 0 \\ 0 & 0 & 0.2 & 0.8 \\ 0 & 0 & 0.2 & 0.8 \end{bmatrix}.$$

The perturbation matrix is generated according to *Assumption 2*. In particular, all entries in the off-diagonal blocks follows an independent, identical distribution (i.i.d): each entry takes the absolute value of a standard normal distribution $\mathcal{N}(0, 1)$ (half-normal distribution). The entries in the diagonal blocks are negative. For each row in a diagonal block, we first generate n_I i.i.d. half-normal random variables, which decide the ratios between them, then we scale them so that

the row sum of $\mathbf{C}$ is zero. For example,

$$\mathbf{C} = \begin{bmatrix} -0.4014 & -0.4325 & 0.7885 & 0.0455 \\ -0.7205 & -0.3341 & 0.5408 & 0.5139 \\ 0.4456 & 0.6599 & -0.2130 & -0.8925 \\ 0.4914 & 0.5054 & -0.5498 & -0.4469 \end{bmatrix}.$$

In this case the maximum degree of coupling (5.6): $\delta = \max_I \left(\sum_{J \neq I} \sum_{j=1}^{n_J} C_{i_I j_J} \right) = 1.1055$. The result of *Theorem 4* indicates $\epsilon < \frac{1}{6\delta} = 0.1508$. We choose $\epsilon = 0.15$, and the resulting $\mathbf{P}$ is given by

$$\mathbf{P} = \begin{bmatrix} 0.6398 & 0.2351 & 0.1183 & 0.0068 \\ 0.5919 & 0.2499 & 0.0811 & 0.0771 \\ 0.0668 & 0.0990 & 0.1681 & 0.6661 \\ 0.0737 & 0.0758 & 0.1175 & 0.7330 \end{bmatrix};$$

note $\mathbf{P}$ does not have a significant "repeated row" structure. For this small example, we can easily compute the stationary distribution of $\mathbf{P}$, and select $\boldsymbol{\mu}$ by the stationary weights. The algorithm successfully aggregates the first two states and the last two states. The resulting $\mathbf{Q}$ matrix for the aggregated chain is given by

$$\mathbf{Q} = \begin{bmatrix} 0.6909 & 0.3091 \\ 0.2718 & 0.7282 \end{bmatrix}.$$

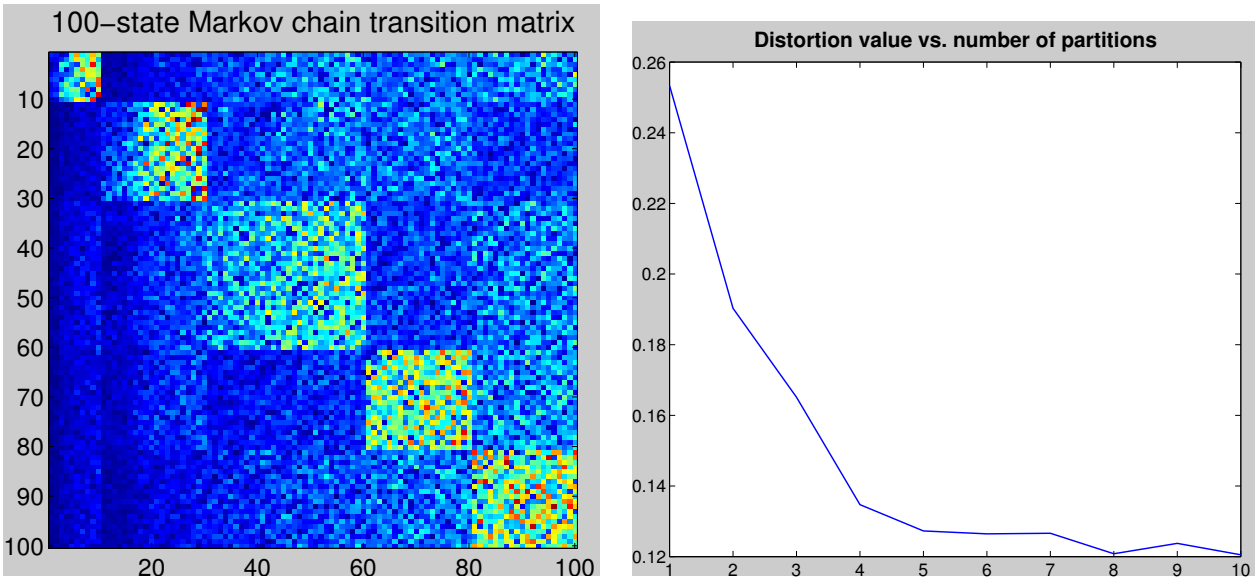
The stationary distributions of a direct aggregation of $\boldsymbol{\pi}(\mathbf{P})$, and $\boldsymbol{\pi}(\mathbf{Q})$ at different stages of the algorithm (indicated by value β) are below. It is evident that the MEP-based clustering algorithm not only identifies correct sub-chains, but also provides increasingly better approximation to the limiting distribution of Markov process $\mathcal{X}$.

Table 5.1: l_1 distance between the stationary distribution of aggregated Markov chain $\mathbf{Q}$ and the aggregation of stationary distribution of $\mathbf{P}$, or $\|\boldsymbol{\pi}(\mathbf{Q}) - \boldsymbol{\pi}(\mathbf{P})\tilde{\boldsymbol{\Phi}}\|_1$, at different stage of annealing.

β	$\boldsymbol{\pi}(\mathbf{P})\tilde{\boldsymbol{\Phi}}$	$\boldsymbol{\pi}(\mathbf{Q})$	l_1 distance
10	[0.5300, 0.4700]	[0.5321, 0.4679]	3×10^{-3}
20	[0.5321, 0.4679]	[0.5319, 0.4681]	2.6981×10^{-4}
50	[0.5321, 0.4679]	[0.5321, 0.4679]	2.1776×10^{-8}

5.3.3 High order Markov chain

We consider a relatively large NCD Markov chain with 100 states whose transition matrix is given in Figure 5.3 (a). There are 5 set of states within which transitions between states are highly possible. Each partition contains $10, 20, 30, 20, 20$ states, respectively, moreover, the states are ordered such that the blocked diagonal structure is instantly observable. Perturbations are added such that constraints (2.3) are satisfied.



(a) (b)

Figure 5.3: (a) - Transition probability matrix $\mathbf{P}$ of the 100-state Markov chain, warm color indicates a larger value, and a colder color indicates a smaller (nonnegative) value. (b) - The aggregation errors (soft dissimilarity $\tilde{\nu}(\mathbf{P}, \mathbf{Q})$) achieved by the lower-order Markov chain $\mathbf{Q}$ resulted from clustering, as function of size of $\mathbf{Q}$.

If we fix the size of $\mathbf{Q}$ to be 5, then regardless of the permutation of the states of $\mathbf{P}$, our row-wise clustering algorithm always identifies the correct sub-chains. The soft aggregation matrix $\tilde{\mathbf{\Phi}} \in [0,1]^{100\times 5}$, the resulting composite transition matrix $\mathbf{Z} \in \mathbb{R}^{5\times 100}$, and the corresponding transition matrix for superstates $\mathbf{Q} \in \mathbb{R}^{5\times 5}$ are given in Figure 5.4 (a) to (c). The $\tilde{\mathbf{\Phi}}$ matrix demonstrates a nearly deterministic partition and the superstate ordering. By adopting the same permutation on the $\mathbf{Q}$ matrix, we obtain a transition matrix corresponding to the 5 superstates in the original order (Figure 5.4 (d)).

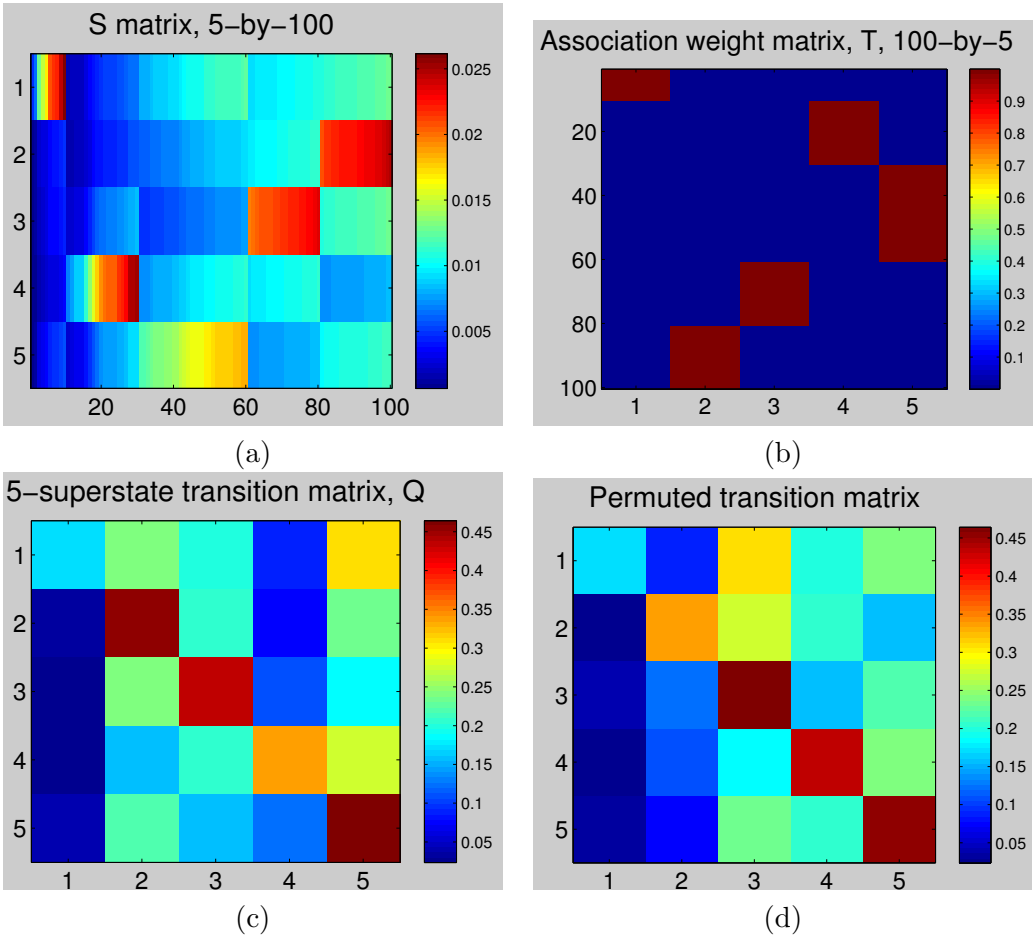


Figure 5.4: The scaled color plots of clustering results for the 100-state Markov chain given by Figure 5.3. Plots (a) and (b) show the composite transition matrix $\mathbf{Z}$ and the soft aggregation matrix $\tilde{\mathbf{\Phi}}$, solved using the relaxed optimization problem (2.18). Plot (c) shows the transition matrix of the aggregated 5-state Markov chain $\mathcal{Y}$, calculated as $\mathbf{Q} = \mathbf{Z}\tilde{\mathbf{\Phi}}$. Note that the states in plot (c) are not ordered according to the usual sub-chain order, specifically, states $\{1, 2, 3, 4, 5\}$ correspond to sub-chains $\{1, 5, 4, 2, 3\}$, which is indicated by the $\tilde{\mathbf{\Phi}}$ matrix. Plot (d) is the transition matrix of the 5-state chain after applying the original sub-chain reordering.

The aggregated stationary distribution of $\mathbf{P}$ is given by

$$\boldsymbol{\pi}\boldsymbol{\Phi} = [0.0365\ 0.1358\ 0.2990\ 0.2468\ 0.2819].$$

It is observed that, the stationary distribution of the aggregated Markov chain $\mathbf{Q}$ meets $\boldsymbol{\pi}\boldsymbol{\Phi}$ with high accuracy. Moreover, this is independent of the initial selection of state weights. Specifically, if we repeat the same procedure with two choices of state weights, (1) $\boldsymbol{\mu}_s := \boldsymbol{\pi}$ being proportional to the stationary distribution, and (2) $\boldsymbol{\mu}_u$ being uniform over all states, the l_1 distances between $\boldsymbol{\pi}(\mathbf{Q})$ and $\boldsymbol{\pi}(\mathbf{P})\boldsymbol{\Phi}$ are plotted in Figure 5.5. As the annealing parameter β increases (that is by pushing the soft aggregation matrix T towards a hard limit), both choices of initial weights $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_u$ achieve excellent accuracy in the limiting distribution (note the scale of the y-axis is 10^{-5}). Considering the ease of implementation using $\boldsymbol{\mu}_u$, this will be an excellent choice for high-order systems.

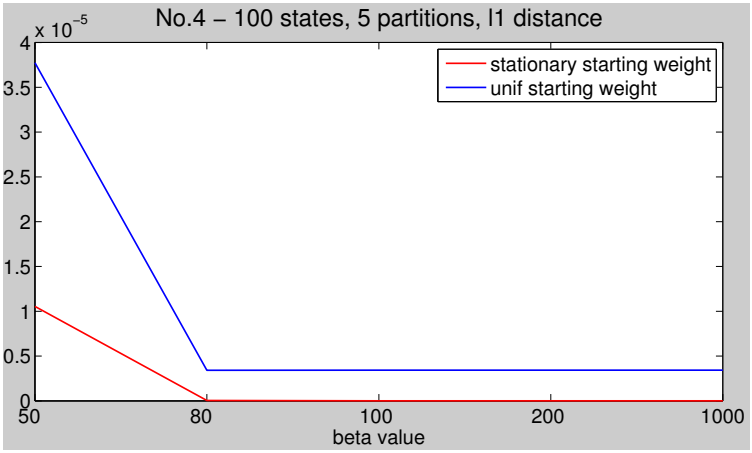


Figure 5.5: The l_1 distances between the limiting distribution of the aggregated chain, and the true partitions, $\|\boldsymbol{\pi}(\mathbf{Q}) - \boldsymbol{\pi}(\mathbf{P})\boldsymbol{\Phi}\|$.

5.3.4 Comparison with Spectral Clustering

In this subsection we provide a set of comparative simulations for our MEP-based clustering and the well-known Normalized Cut algorithm proposed by Shi and Malik [61], which is closely related to random walks. In particular, for NCD Markov chains given by $\mathbf{P} = \mathbf{P}^* + \epsilon\mathbf{C}$, we focus on *subchain identifiability* as the perturbation size ϵ is increased. That is, the effectiveness of the algorithm to

correctly identify the states associated with each noncommunicating subchain of $\mathbf{P}^*$.

The simulations are constructed as follows. For an NCD Markov chain $\mathbf{P} = \mathbf{P}^* + \epsilon\mathbf{C}$, we first generate the completely decomposable component $\mathbf{P}^*$, which is a block diagonal stochastic matrix, and the perturbation matrix $\mathbf{C}$, with the degree of coupling δ being 1 and satisfying *Assumption 1* (negative perturbations in the diagonal blocks). Next we increase the perturbation size ϵ from 0 to some value for which the resulting $\mathbf{P}$ remains non-negative, and observe the partition results from using our MEP-based algorithm and the Normalized Cut algorithm. For the MEP algorithm, we choose uniform state weightings. For the Normalized Cut algorithm, we form fully connected similarity graphs, whose edge weights are specified by the Gaussian similarity function $W_{ij} = Sim(\mathbf{p}(i), \mathbf{p}(j)) := \frac{\exp(\|\mathbf{p}(i)-\mathbf{p}(j)\|_2^2)}{2\times 0.01}$. Then we proceed with the normalized graph Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$, where $\mathbf{D} = \text{diag}\{\sum_{j=1}^{n} W_{ij}\}$ (see [61] and [66] for details).

The identifiability of both algorithms is similar for those cases in which $\mathbf{P}^*$ consists of 2 subchains. In Figure 5.6, we plot the partition accuracies for two sets of small-sized Markov chains (10 states and 100 states), where the $\mathbf{P}^*$ matrices have 2 subchains. The curves show the percentages of correct identification of state partitions in 10 simulation runs, in which 10 different $(\mathbf{P}^*, \mathbf{C})$ pairs are generated. It is seen that as the value of ϵ is increased, the MEP based algorithm tends to retain correct partition results for a slightly wider range of perturbation, which implies robustness of the algorithm.

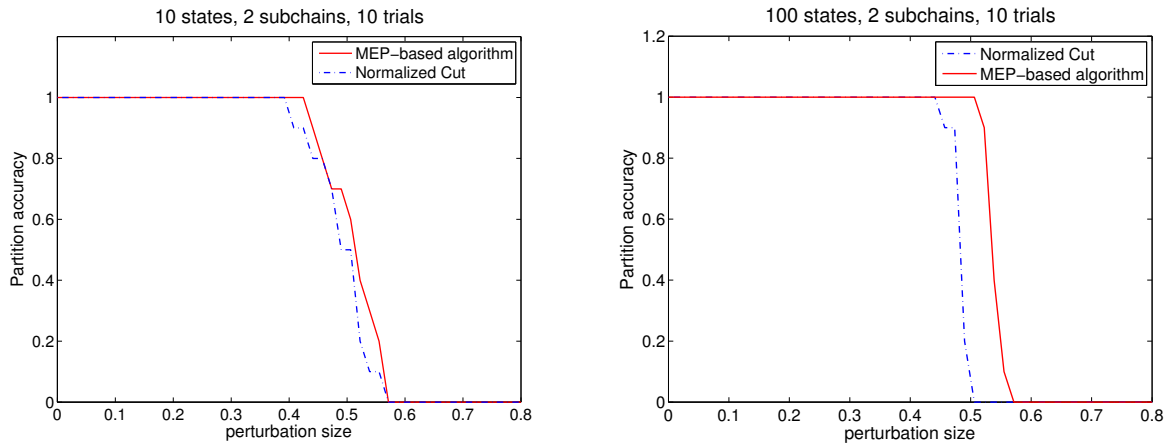


Figure 5.6: Percentage of correct identification of the state partitions in 10 simulation runs using the MEP-based aggregation algorithm and the Normalized Cut algorithm.

For the multiple subchain case, simulations indicate the MEP algorithm recovers the subchains more consistently than the Normalized Cut algorithm; this advantage becomes significant as the Markov chains become larger. Results are provided in Figures 5.7 and 5.8. We also note that both algorithms are affected by the index of unbalance, η (defined in (5.7)), which influences the row separability of the $\mathbf{P}$ matrices. In fact, simulations indicate the identifiability of the MEP algorithm is more robust with increasing η (see Figure 5.9).

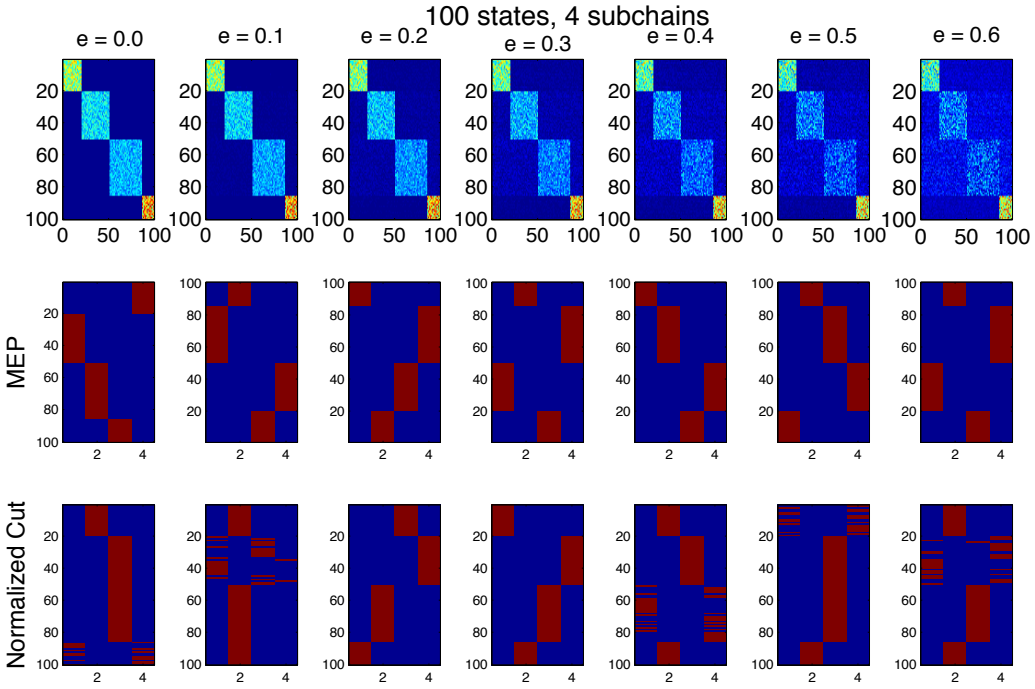


Figure 5.7: State partition results for Markov chains with 100 states and 4 subchains. The first row are the original transition matrices, the second and the third rows are the partition results from using the MEP-based algorithm and the Normalized Cut algorithm, respectively. In each column, the perturbation value ϵ increases from left to right for each row. In these cases, the index of unbalance of $\mathbf{P}^*$ is given by $\eta = 0.3235$.

A summary of results are provided in Table 5.2. We show the numbers of successfully identifying all subchains in 10 trials, where we randomly generate the $\mathbf{P}^*$ and $\mathbf{C}$ matrices. We test for perturbation sizes ϵ in the range of $[0, 0.5]$, and dimensions from 10-state to 500-state. Throughout all cases the η values are controlled so that they are all in the $[0.25, 0.4]$ range. We can see that both algorithms achieve high accuracy for bi-partitions, and the Normalized Cut appears to be

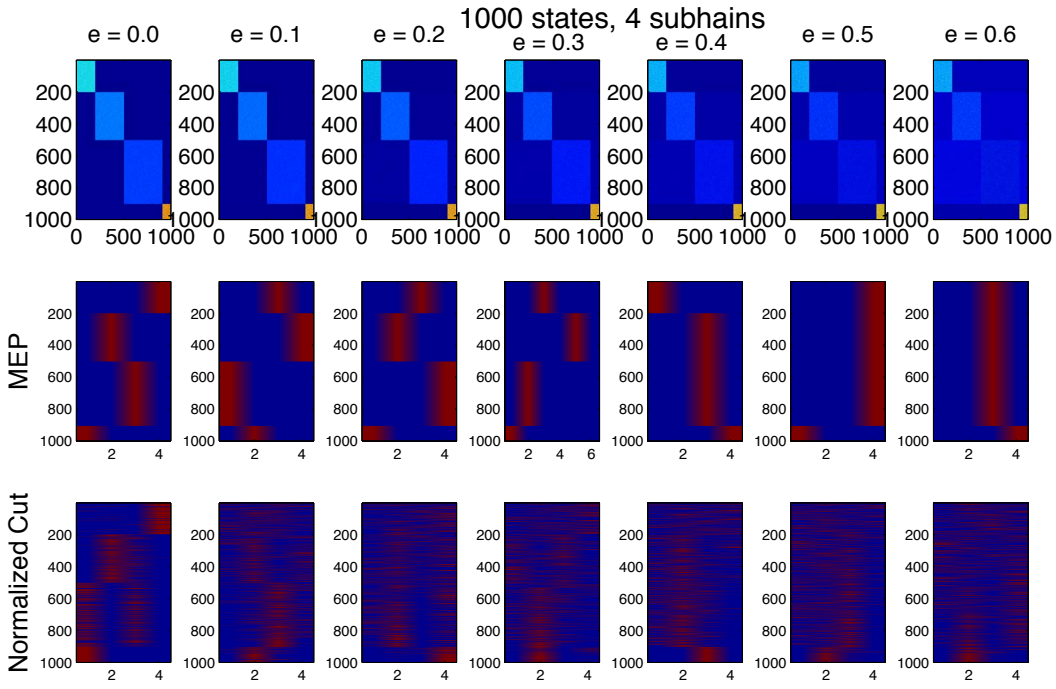


Figure 5.8: State partition results for Markov chains with 1000 states and 4 subchains. In these cases, the index of unbalance of $\mathbf{P}^*$ is given by $\eta = 0.2798$.

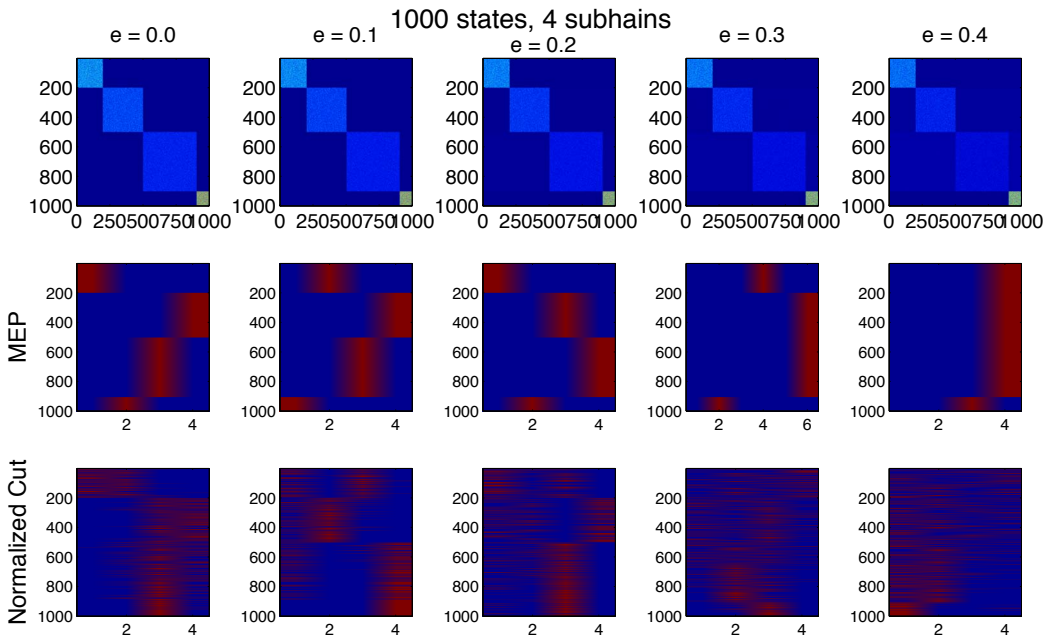


Figure 5.9: State partition results for Markov chains with 100 states and 4 subchains, the index of unbalance of $\mathbf{P}^*$ is given by $\eta = 0.6252$.

slightly better for the $n = 500$ case. But when the number of subchains is larger than 2, the MEP algorithm outperforms the Normalized Cut algorithm, in particular, the latter fails to identify in all examples when $n = 500$, and the results are similar to those shown in the last rows of Figures 5.8 and 5.9.

Summary of This Chapter

Our main achievements in this chapter are summarized as follows.

1. Adaptation of the resource allocation framework for aggregating Markov chains.

2. Development of the state aggregation results (*Theorem 2*) and phase transition conditions (*Theorem 3*) for Markov chains when the state comparison is defined by the KL divergence.

3. Characterization of a sufficient condition for the NCD Markov chains under which our MEP-based aggregation algorithm is guaranteed to identify the decomposable sub-chains (*Theorem 4*).

Table 5.2: Number of correct partitions over 10 instances with randomly generated $\mathbf{P}^*$ and $\mathbf{C}$. The number of states is n, the number of subchains is m, the perturbation size is ϵ.

$m = 2$: MEP	$\epsilon = 0$	0.1	0.2	0.3	0.4	0.5
$n = 10$	10	10	10	10	10	8
$n = 100$	10	10	10	10	10	10
$n = 500$	10	10	10	6	6	4
$m = 2$: Normalized Cut						
$n = 10$	10	10	10	10	9	7
$n = 100$	10	10	10	10	10	9
$n = 500$	10	10	10	10	10	9
$m = 3$: MEP	$\epsilon = 0$	0.1	0.2	0.3	0.4	0.5
$n = 10$	10	10	10	10	10	10
$n = 100$	10	10	10	10	10	10
$n = 500$	10	10	10	10	8	0
$m = 3$: Normalized Cut						
$n = 10$	6	7	8	6	8	2
$n = 100$	8	7	7	8	7	5
$n = 500$	0	0	0	0	0	0
$m = 5$: MEP	$\epsilon = 0$	0.1	0.2	0.3	0.4	0.5
$n = 10$	10	10	10	10	10	10
$n = 100$	10	10	10	10	10	10
$n = 500$	10	10	10	6	5	0
$m = 5$: Normalized Cut						
$n = 10$	5	2	4	3	6	3
$n = 100$	2	2	1	2	4	4
$n = 500$	0	0	0	0	0	0

4. Discussion of the steady state dynamics of the Markov chain aggregation results, and development of an error bound between the stationary distribution of the aggregated chain and the aggregated stationary distribution of the original chain (*Theorem 7*).

Chapter 6

Applications: Aggregation of Parameterized Models

Studies in various disciplines that include biology, economics, social sciences, computer science, and seismology, require studying stochastic, dynamic networks of interacting processes. Parametric generative models based on historical observations are typically used to represent natural or engineered processes to make policies or operational decisions (e.g., in networked queueing systems and high-frequency finance [4, 18]), to avoid hazards (e.g., in earthquake prediction and wildfire response [49, 69]) and to gain a stronger understanding of underlying physical mechanisms (e.g., in neuroscience studies [53]). These models are characterized by parameters that describe the behavior of each process as well as parameters that describe the interdependencies between the processes, and hence naturally induce a graph representation. For instance, we can model each stochastic process as a node, and the mutual interactions (inhibition or excitation) as edges between a pair of nodes.

In many cases, the system contains numerous interacting processes, and directional relations are necessary to describe the system behavior; these usually result in a large directed graph model. Again as before, it is useful to have aggregated representations that capture the dominant or ensemble interrelations in systems.

In this chapter, we aim to derive methods for the aggregation (via clustering) of large networks of parametric generative models. This is an adaptation of the MEP-based aggregation framework introduced in Chapter 2 and developed in Chapter 4. The main focus here is to investigate the functional aggregation of a class of parametrized generative models. More specifically, we aggregate similar *random processes* and obtain a set of *representative processes* in an ensemble, where a precise notion of similarity is developed. To implement this aggregation process, we extend the maximum entropy principle based methods developed for graph models to discover the underlying structure of functional units in networked random processes. As described in Chapter 2, the functional

aggregation objective can be formulated as a combinatorial optimization problem which aims to minimize the dissimilarity between the original and the aggregated networked processes. After aggregation, processes with similar functions are identified as belonging to a single functional unit. Thus the goal of studying functional interactions among K random processes is recast as a study of the relations of K' functional units, where $K' < K$, therefore providing a visualization of a coarser representation of the ensemble.

6.1 Problem Setup

We consider K random processes that have been recorded simultaneously. Let the i^{th} random process be $\mathcal{X}_i = (X_i(1), \ldots, X_i(M), \ldots)$ and $x_i(\tau)$ be the realization of X_i at the τth time unit, $1 \leq i \leq K, \tau \geq 1$. We denote the collection of K random processes as $\underline{\mathbf{X}} = \{\mathcal{X}_1, \ldots, \mathcal{X}_K\}$, with realization $\underline{x}$. The joint distribution of $\underline{\mathbf{X}}$ is given by

$$\mathbb{P}_{\underline{\mathbf{X}}}(\underline{x}) = \prod_{i=1}^{K} \prod_{\tau \geq 1} \mathbb{P}_{X_i(\tau)|\underline{\mathbf{X}}(1:\tau-1)}(x_i(\tau)|\underline{x}(1:\tau-1)), \tag{6.1}$$

where $\underline{\mathbf{X}}(1:M) \stackrel{\triangle}{=} \{X_{i',\tau'} : 1 \leq i' \leq K,\ 1 \leq \tau' \leq M\}$ denotes all the processes up to the M^{th} time unit. We define the *history* as $\mathcal{H}_M \stackrel{\triangle}{=} \underline{\mathbf{X}}(1:M-1)$. Thus the joint distribution (6.1) can be rewritten as

$$\mathbb{P}_{\underline{\mathbf{X}}}(\underline{x}) = \prod_{i=1}^{K} \prod_{\tau \geq 1} \mathbb{P}_{X_i(\tau)|\mathcal{H}_\tau}(x_i(\tau)|\mathcal{H}_\tau). \tag{6.2}$$

In this chapter, we are interested in parametric statistical models for random processes, whose representations at time t depend solely on a *finite* history $\mathcal{H}_\tau$ ($\tau < t$), as shown in the following two examples.

Example [Gauss-Markov processes] A network of Gauss-Markov processes with K individual processes and M-step history dependence are modeled as

$$X_i(t) = \gamma_{i,0,0} + \sum_{j=1}^{K} \sum_{\tau=1}^{M} \gamma_{i,j,\tau} X_j(t-\tau) + W_i(t), \quad 1 \leq i \leq K$$

where $W_i(t)$ are i.i.d. and Gaussian ($\mathcal{N}(0, \epsilon_i)$). For each process $\mathbf{X}_i$, the model parameters $\{\gamma_{i,\cdot,\cdot}\} \in \mathbb{R}^{K\times M}$ indicate how the future of $\mathbf{X}_i$ depends on the past K processes through M previous time units.

Example [Renewal process models for earthquakes] Let us first recall the definition of a Poisson process: A *Poisson process* $N(t)$ of rate $\lambda \geq 0$ is a *counting process* satisfying the following conditions:

1. For any interval $(t, t+\Delta t](t > 0), \Delta N_{(t,t+\Delta t]} = N(t+\Delta t) - N(t)$ has a Poisson distribution $Poi(\lambda \Delta t)$;

2. (Independence) For any non-overlapping intervals $(t, t+\Delta t]$ and $(s, s+\Delta s], t \neq s$, the counts $\Delta N_{(t,t+\Delta t]}$ and $\Delta N_{(s,s+\Delta s]}$ are independent.

A general renewal process extends the Poisson process by allowing the rate to depend on time and history. Suppose there are K renewal processes in an ensemble and $\mathcal{H}(t)$ contains the information of all processes up to time $t-1$, let the rate function be $\lambda(t|\mathcal{H}(t))$.

It is well known that a renewal process is completely characterized by $\lambda(t|\mathcal{H}(t))$. Though there are some non-parametric methods to estimate these rate functions [27], parametric models are more popular and can provide tools to analyze the behaviors and interactions of the individual processes.

Earthquakes over time and geological locations provide an important example that can be modeled as a renewal process [28, 48], in which the aftershocks of a major earthquake exceeding certain magnitude are modeled as events. The conditional rate of this process at the i^{th} location, $\lambda_i(t|\mathcal{H}_t)$, depends on the background seismicity rate at that location g_{i0}, and the aftershock counts in other locations $\Delta N_j, j = 1, 2, \cdots, K$ up to time $t-1$. One of the most common and basic parametrizations is called the *epidemic-type* model first introduced in [28]:

$$\lambda_i(t) = g_{i0} + \sum_{j=1}^{K} \int_0^t g_{ij}(t-\tau) dN_j(\tau), \quad 1 \leq i \leq K \tag{6.3}$$

in which $g_{ij}(t)$ is a function of the aftershock magnitude observed at location j and time t. These models show the dependence between different processes, and enable comparison between processes in terms of their responses under the same inputs. That is, suppose that aftershock rates in two locations are modeled by (6.3); the prediction results would be indistinguishable if the two models

have the same parameters, regardless of their geological locations. Therefore, we can cluster the processes by aggregating them according to their responses under the same inputs, providing models of dominant earthquake propagation trends.

6.2 Aggregation of Parameterized Models

As is introduced in Section 6.1, our goal of identifying the functional interactions among processes can be interpreted from a clustering perspective: partition all processes in a system of interest into cells by aggregating *similar* processes, then determine the interactions among these cells.

We first set up a mapping from networked random processes described by generative models to weighted directed graphs, and convert the problem into clustering the nodes in the graph. Then we define a function that evaluates the "dissimilarity" of the input-output behaviors of two generative models in terms of their parameters. Finally, we adapt the graph aggregation framework proposed in Chapter 4 and obtain an aggregation of the corresponding parametric random processes.

6.2.1 Graph representation

Generative parametric models are determined by fitting observation data. For example, in earthquake studies where the conditional rate functions $\lambda_i(t|\mathcal{H}(t))$ are parametrized by (6.3). Random processes at multiple locations can easily be cast in a graph structure, whose i^{th} node represents the random process in the i^{th} location, and the edge weight on directed edge (i, j) reflects the influence of the process from the j^{th} location to the i^{th} location. Note that the edge weights may vary with time. Therefore, aggregating similar random processes can be converted to aggregating similar nodes in the graph, where the aggregation varies with time.

For example, suppose there are K conditional rate models (6.3) for earthquakes in K different locations. Let $\boldsymbol{\gamma}_i(t) \triangleq [g_{i0}, g_{i1}(t), \cdots, g_{iK}(t)]$ be the model parameters for the i^{th} process, in which $g_{im}(t) = \int_0^t g_{im}(t-\tau)dN_m(\tau)$ is the instantaneous influence from the aftershock process in the m^{th} location. Then in the directed graph describing the relations of these K processes, $g_{im}(t)$ represents the (dynamic) weight on the edge (m, i) at time t. Therefore, the parametric vector $\boldsymbol{\gamma}_i$ completely characterizes the statistical model of i^{th} process.

6.2.2 Similarity function

In order to use the graph aggregation framework proposed in Chapter 4, we need to define the the notion of similarity, or equivalently of *dissimilarity* between two nodes. The setting considered in Chapter 3 refers to static graphs, with constant edge weights. Here, in general, the edge weights can be in vector form (e.g., when the history information is discretized) and varying with time.

We say that two processes are *similar* if given the same inputs, they produce similar outputs. We choose the dissimilarity function d_{ij} between two processes such that the following conditions are satisfied.

1. d_{ij} is a function of $\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_j$, such that d_{ij} is increasing as $\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j\|$ increases, and $d_{ij} = 0$ only if $\boldsymbol{\gamma}_i = \boldsymbol{\gamma}_j$, here $\|\cdot\|$ stands for some norm compatible with the model parameters.

2. When considering the i^{th} and j^{th} processes, influences between these two processes are treated separately from influences from an outside process k ($k \neq i, j$). We denote these two types of dissimilarities by $d_{ij}(pair)$ and $d_{ij}(out)$. For example, in Figure 6.1 (without loss of generality, assume that all edges have the same weight), the influences between node i and node j are the same (symmetric) in cases (b) and (c), but are asymmetric in case (a), that is $d_{ij}(pair)(b) = d_{ij}(pair)(c) = 0 < d_{ij}(pair)(a)$. On the other hand, nods i and node j in cases (a) and (b) are equally influenced by outside nodes, that is $d_{ij}(out)(a) = d_{ij}(out)(b) = 0$. So only in case (b), we have $d_{ij}(out)(b) + d_{ij}(pair)(b) = 0$.

3. Geometric location (and/or other useful intrinsic features) of a process is considered. Let $d_{ij}(loc)$ be the locational distance between the i^{th} and j^{th} processes. d_{ij} can depend on $d_{ij}(loc)$ for geographic-dependent models such as earthquakes.

Therefore we can choose a dissimilarity function with the following structure

$$d_{ij} = \alpha_1 d_{ij}(out) + \alpha_2 d_{ij}(pair) + \alpha_3 d_{ij}(loc), \tag{6.4}$$

in which the vector $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \alpha_3] \geq \mathbf{0}$ weights the three terms, and the specific forms of $d_{ij}(out)$ and $d_{ij}(pair)$ depend on the structure of the edge weights.

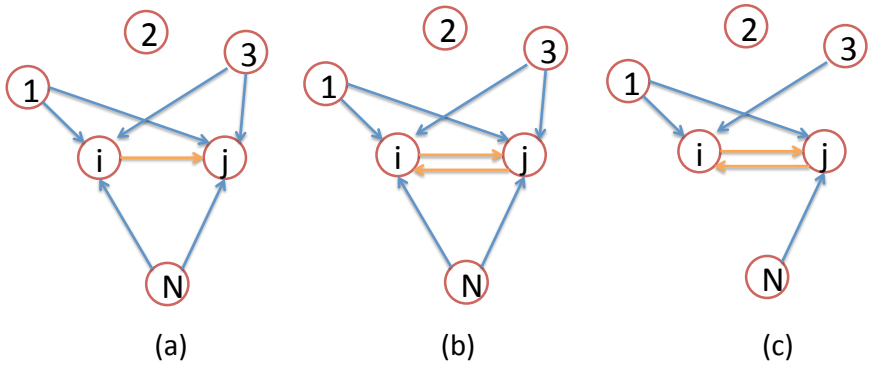


Figure 6.1: Illustration of the selection of distance function. Assume all edge weights are the same. The i^{th} and j^{th} nodes in case (a) and case (b) have the same outside influences, but $d_{ij}(b) < d_{ij}(a)$ because $d_{ij}(pair)(a) > d_{ij}(pair)(b)$. In cases (b) and (c), i and j have the same pairwise influence between each other, but the outside influences are different, so $d_{ij}(b) < d_{ij}(c)$.

6.2.3 Aggregation of the parametrized models

We have modeled a set of parametrized stochastic processes as a graph. After defining a dissimilarity function between individual processes as in (6.4), the aggregation of similar parameterized models reduces to the aggregation of the corresponding nodes in a graph. To proceed, we will use the MEP-based aggregation framework we proposed for general weighted directed graphs in Chapter 4.

The objective of this clustering problem is as follows: Given K parametric random process models for conditional rates $\lambda_1, \lambda_2, \cdots, \lambda_K$, with model parameters $\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_K$, determine K' clusters of ensemble representative random processes, whose conditional rates $\hat{\lambda}_1, \hat{\lambda}_2, \cdots, \hat{\lambda}_{K'}$ with model parameters $\hat{\boldsymbol{\gamma}}_{\mathbf{1}}, \cdots, \hat{\boldsymbol{\gamma}}_{\mathbf{K'}}$, such that the $\hat{\lambda}_i$ with $\hat{\boldsymbol{\gamma}}_i$ represents the parametrized model for the *process corresponding to the i^{th} cluster.*

In terms of the aggregation of large graphs, this can be stated as: Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ with K nodes, edges $\mathcal{E}$ and edge weights $\mathbf{W}$ being specified by $W_{ij} = \boldsymbol{\gamma}_{i,j} = \{\gamma_{i,j,t} | t > 0\}$ for continuous models, (or $W_{ij} = \boldsymbol{\gamma}_{i,j} = [\gamma_{i,j,1}, \cdots, \gamma_{i,j,\tau}, \cdots]$, for discretized models). Find an aggregated graph $\hat{\mathcal{G}}(\hat{\mathcal{V}}, \hat{\mathcal{E}}, \hat{\mathbf{W}})$ with K' *supernodes*, such that the dissimilarity between the two graphs is minimized. Therefore, we are in the same setting as the general graph clustering/aggregation problems, thought the concepts of nodes, edges and edge weights are different. Nevertheless, the aggregation methodology still applies here. So we modify the two step clustering approach for

graphs with constant scalar edge weights proposed in Chapter 4 to accommodate vector valued and possibly non-constant edge weights:

1. (Node aggregation) For an aggregated graph with a desired size, say K' nodes, search over the set of composite graphs $\boldsymbol{\mathcal{C}}$ and find a $\mathcal{C}^{\phi}(\mathbf{Z})^*$ that achieves the minimal dissimilarity between $\mathcal{G}$ and the set $\tilde{\mathbf{G}}$. In other words, this step calculates an intermediate parametric model for each aggregated process, as a function of the original processes.

   For example, in the earthquake example (6.3), node aggregation results in the conditional rate functions, given by

$$\tilde{\lambda}_i(t) = \tilde{g}_{i0} + \sum_{j=1}^{K} \int_0^t \tilde{g}_{ij}(t-\tau) dN_j(\tau),\ 1 \le i \le K'.$$

2. (Edge aggregation) Aggregate the edges of $\mathcal{C}^{\phi}(\mathbf{Z})^*$ to obtain an aggregated graph $\mathcal{G}^*$. This aggregated graph corresponds to the parametrized models of representative processes in terms of the history of the supernodes activities.

   In the earthquake example, these are the conditional rate models among different cluster centers:

$$\hat{\lambda}_i(t) = \hat{g}_{i0} + \sum_{j=1}^{K'} \int_0^t \hat{g}_{ij}(t-\tau) d\hat{N}_j(\tau),\ \ 1 \le i \le K',$$

   where $d\hat{N}_j(\tau) \stackrel{\triangle}{=} \sum_{j'} d\hat{N}_{j'}(\tau)$, for all processes j' associated with cluster j.

The optimization problem associated with the first step is given by minimizing the following *distortion* function:

$$\min_{\mathcal{G} \in \mathcal{C}^{\phi}(\mathbf{Z})} \rho(\mathcal{G}, \mathcal{C}) \stackrel{\triangle}{=} \sum_{i=1}^{K} \mu_i \min_{1 \le j \le K'} d_{ij}, \tag{6.5}$$

in which d_{ij} is defined in (6.4), and μ_i is simply $\frac{1}{K}$ since we treat the models equally. In general, μ_i can be any (non-negative) normalized weighting parameters for each node. For example, when the model parameters are estimated with different errors, then μ_i can be chosen as $\exp(\sigma_i)/[\sum_{k=1}^{K} \exp \sigma_k]$, where σ_i denotes the error standard deviation in estimating λ_i.

The formulation (6.5) still defines a combinatorial optimization problem by our selection of dissimilarity function between nodes. For an efficient determination of a good approximation solution, we again apply the MEP-based algorithm. Specifically, let $p_{j|i}$ be the association weights between the j^{th} supernode and the i^{th} node, we have the following soft dissimilarity function:

$$\rho'(\mathcal{G}, \tilde{\mathcal{G}}) \stackrel{\triangle}{=} \sum_{i=1}^{K} \mu_i \sum_{j=1}^{K'} p_{j|i} d_{ij}. \tag{6.6}$$

An entropy term evaluates the uncertainty of the soft associations, given by

$$H(\tilde{\mathcal{C}}|\mathcal{G}) = -\sum_{i=1}^{K} \mu_i \sum_{j=1}^{K'} p_{j|i} \log p_{j|i}. \tag{6.7}$$

And during the annealing process, we minimize the following Lagrangian using an increasing value of annealing parameter β:

$$F(\tilde{\mathcal{G}}, \mathcal{C}) = \rho'(\mathcal{G}, \tilde{\mathcal{G}}) - \frac{1}{\beta} H(\tilde{\mathcal{G}}|\mathcal{G}). \tag{6.8}$$

As was derived earlier for the scalar case, solving the optimization problem (6.8) at a fixed β value results in the following optimal association weights and edge weights:

$$p_{j|i} = \frac{\exp\{-\beta d_{ij}\}}{\sum_{h=1}^{K'} \exp\{-\beta d_{ih}\}} \tag{6.9}$$

$$\tilde{\boldsymbol{\gamma}}_j = \sum_{i=1}^{K} q_{i|j} \boldsymbol{\gamma}_i, \tag{6.10}$$

in which $q_{i|j}$ is the posterior distribution of (6.9). Using these results, the second edge aggregation step is computed as

$$\hat{\boldsymbol{\gamma}}_{il} = \sum_{i=1}^{K} p_{l|j} \tilde{\boldsymbol{\gamma}}_{ij}. \tag{6.11}$$

In the iterative computational algorithm, we increase the annealing parameter β from a very small value to a very large value. Thus we begin the process with a convex Lagrangian (6.8) (which has a unique global optimizer) far from the original objective (6.4), and gradually move to a less

convex, but closer approximation of (6.4). As noted in [55,59,60,72], in this process, the number of supernodes automatically increases from one to many, with the goal of tracking the local minima of (6.4).

6.3 Example and Simulation

To illustrate our aggregation framework, we consider epidemic-type earthquake models (6.3), for the mutually excitatory effects of aftershock activities in different locations. The model depends on the history of the aftershocks, represented by a counting process for $0 < \tau \leq t$. The goal of our study is to obtain coarser models that describe the propagation of aftershocks across spatial locations and over time using a clustering approach. The ensemble influence across wider geological regions can be used for future earthquake modeling and forecasting.

To avoid extensive (unnecessary) computational expense, we first discretize the time horizon into intervals of length Δt, such that the probability of having more than one aftershock within a time interval is $o(\Delta t)$. Moreover, we assume the parametrized model has *finite* history dependence. Therefore, the epidemic models in (6.3) for K interactive processes are approximated by summation over finite past information, that is,

$$\lambda_i(N\Delta t) \approx \omega_{i,0} + \sum_{j=1}^{K}\sum_{s=1}^{M} \omega_{i,j,s} dN_j((K-s)\Delta t), \quad 1 \leq i \leq K, \tag{6.12}$$

where $dN_j(s) = N_j((s+1)\Delta t) - N_j(s\Delta t) \in \{0,1\}$, and the history dependence is assumed to be M time steps. For each process, $1 \leq i \leq K$, the aftershock rate is fully captured by coefficients $\boldsymbol{\gamma}_i = (\omega_{i,0}, \cdots, \omega_{i,j,s}, \cdots)$ for $1 \leq j \leq K, 1 \leq s \leq M$, in which $\omega_{i,0}$ represents the base activity level at location i, and $\omega_{i,j,s}$ indicates the mutually excited level.

In the experiment, we first select 25 geographical locations in a domain (see the red circles in Figure 6.2) and construct epidemic models for each location: we assume a maximum history dependence of 8 steps, and randomly select the model coefficients $\omega_{i,0}$ for $1 \leq i \leq 25$. We then begin with an arbitrarily chosen initial condition $\underline{N}(0) \in \{0,1\}^{25}$, and use the prespecified models to compute the aftershock rates λ_i, and the counts $\Delta N_i(k\Delta t)$ for all i and $k > 0$. Repeating this process, we obtain a realization of counting processes (aftershock counts) for each location. These

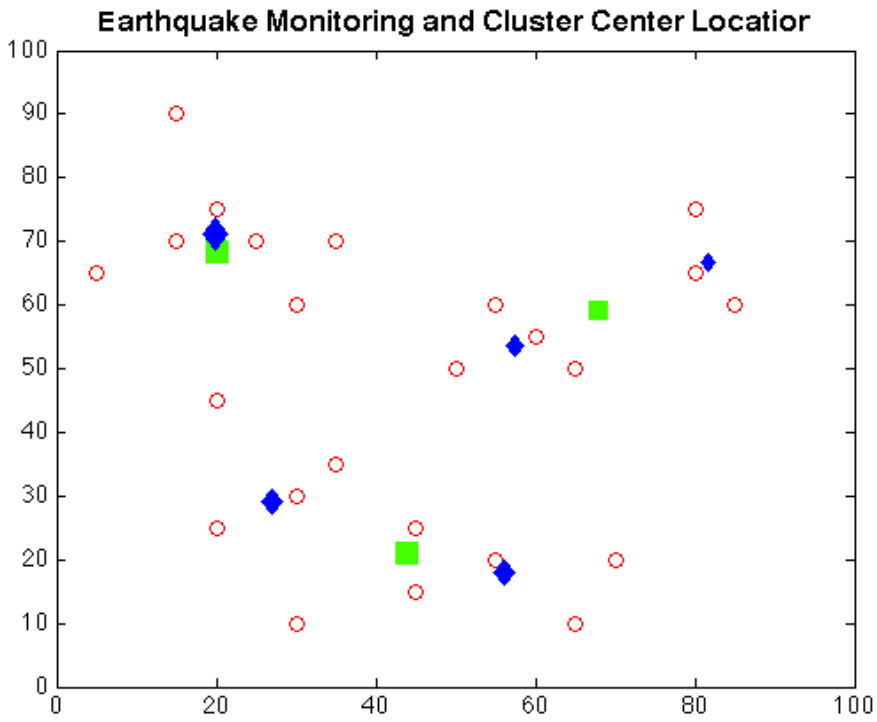


Figure 6.2: The 25 locations where the parametric models are built in a domain (red circles); the equivalent representative locations of the 3-cluster aggregation (green squares) and 5-cluster aggregation (blue diamonds). In latter two cases, the area of each symbol is proportional to the number of processes represented by that cluster.

processes are run for 2000 time steps, and the last 500 steps are used for parametric model fitting. Models in the form of (6.12) with $M = 5$ are selected using the Akaike information criterion (AIC), and the coefficients are shown in Figure 6.3. Note that $\boldsymbol{\gamma}_{i0}$ for each process contains $1+5\times 25 = 126$ parameters.

The clustering algorithm is applied to the resulting generative models, or simply the model coefficients $\boldsymbol{\gamma}$. We define the dissimilarity function between the i^{th} and j^{th} processes by (6.4), with

$$d_{ij}(out) \stackrel{\triangle}{=} \|{\boldsymbol{\gamma}_i}^{[i,j]} - {\boldsymbol{\gamma}_j}^{[i,j]}\|_2^2 \text{ and } \; d_{ij}(pair) \stackrel{\triangle}{=} \|{\boldsymbol{\gamma}_i}^{[j]} - {\boldsymbol{\gamma}_j}^{[i]}\|_2^2, \tag{6.13}$$

in which ${\boldsymbol{\gamma}_i}^{[i,j]} \in \mathbb{R}^{1+M(K-2)}$ is obtained from vector $\boldsymbol{\gamma}_i$ by removing $\gamma_{i,j,s}, s = 1, \cdots, M$, and ${\boldsymbol{\gamma}_i}^{[j]} = \{\gamma_{i,j,s}\}(s = 1, \cdots, M) \in \mathbb{R}^M$ for $K = 25$ and $M = 5$. Thus, $d_{ij}(out)$ accounts for all *outside* model coefficients in process i and process j and $d_{ij}(pair)$ penalizes the asymmetry in the influences between process i and process j. This dissimilarity function is used as the basis of the

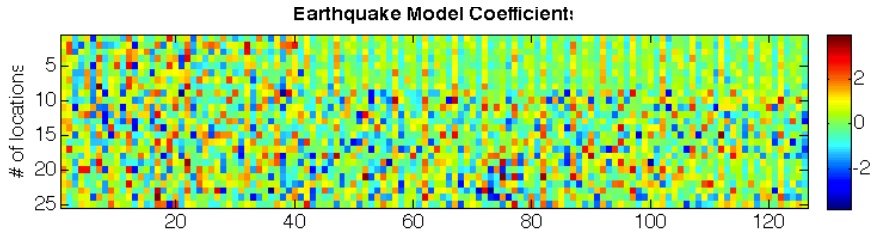


Figure 6.3: The model parameters of the earthquake model for the aftershock activities at the 25 locations. A warmer color indicates a high value and a colder color indicates a low value.

cluster analysis.

We define a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ for these models of aftershock processes where the node v_i represents the process at location i, and the edge weight $W_{ij} \triangleq \{\gamma_{i,j,s}\}(s = 1, \cdots, M) \in \mathbb{R}^M$. As discussed in the general framework in Section 4.1.2, we first aggregate the models by the dissimilarity function defined in (6.4) and (6.13). In this example, we design the dissimilarity function to be more sensitive with the connectivity based distance, while less sensitive to locational information, by choosing $\alpha = [1, 1, 0.5]$. After the node aggregation step, we obtain a set of intermediate epidemic models for cluster representatives (in terms of the original process models), given by

$$\tilde{\lambda}_i(N\Delta t) = \tilde{\omega}_{i,0} + \sum_{j=1}^{K}\sum_{s=1}^{M} \tilde{\gamma}_{i,j,s} dN_i((K-s)\Delta t)), \quad i = 1, \cdots, K'..$$

The second edge aggregation step yields the parametric models that characterize the interactions among different cluster representatives, that is

$$\hat{\lambda}_i(N\Delta t) = \hat{\omega}_{i,0} + \sum_{j=1}^{K'}\sum_{s=1}^{M} \hat{\gamma}_{i,j,s} d\hat{N}_i(s\Delta t)), \ j = 1, \cdots, K'.$$

where $\hat{N}_i(s\Delta t) = \sum_{h \in C_i} R_h(s\Delta t)$ is the total number aftershocks in the locations associated with the i^{th} cluster.

The proper number of supernodes is unknown at the outset and determined by the nature of the problem. In this simulation, we let the desired number of supernodes increase from 1 to 10 and plot the curve of distortion (Figure 6.4). The distortion improves rapidly when the number of

desired clusters surpasses 3 and 5. Figure 6.5 shows the partition results of the 25 processes into 3 and 5 clusters. For example, when we separate the processes into 3 cells of similar processes, each cell contains 8, 10 and 7 locations respectively, which is the correct partition with our design. Figure 6.6 and 6.7 show the coefficients of the resulting epidemic models with 3 and 5 clusters.

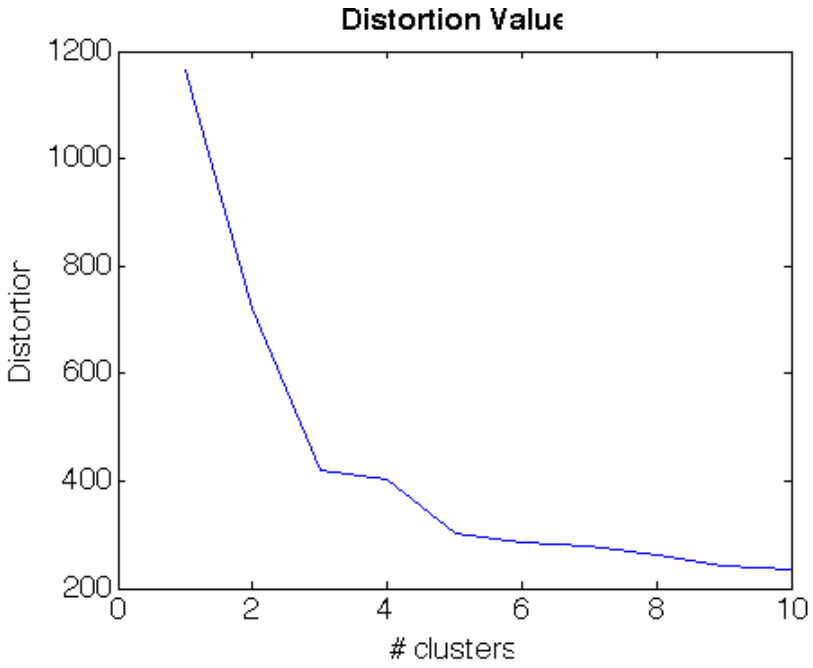


Figure 6.4: The distortion curve vs. the number of clusters.

6.4 Discussion

One of the most important contributions of clustering analysis on parametrized stochastic process models is to improve scalability in studying the functional connections among different processes, such as the causality analysis. For example in neuroscience research, the neuron spiking activities are typically modeled as one-dimensional point processes with generalized linear models [16]. In order to investigate the information propagation among neurons in the brain, various type of causality tests are developed for each pair of neurons. In particular, when checking the Granger causality (one of the most widely studied definitions of causality) [26] from the j^{th} spiking process to the i^{th} spiking process, generalized linear models of process i are estimated without and with the information of process j. The extent of causal relationship is determining from the improvement of

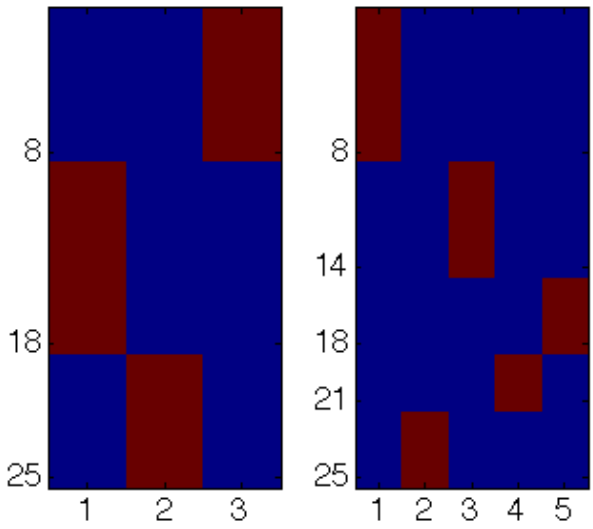


Figure 6.5: The partition results of the 25 process when the objective model sizes are 3 and 5. The vertical axis marked with the cut-off number of processes in each cluster.

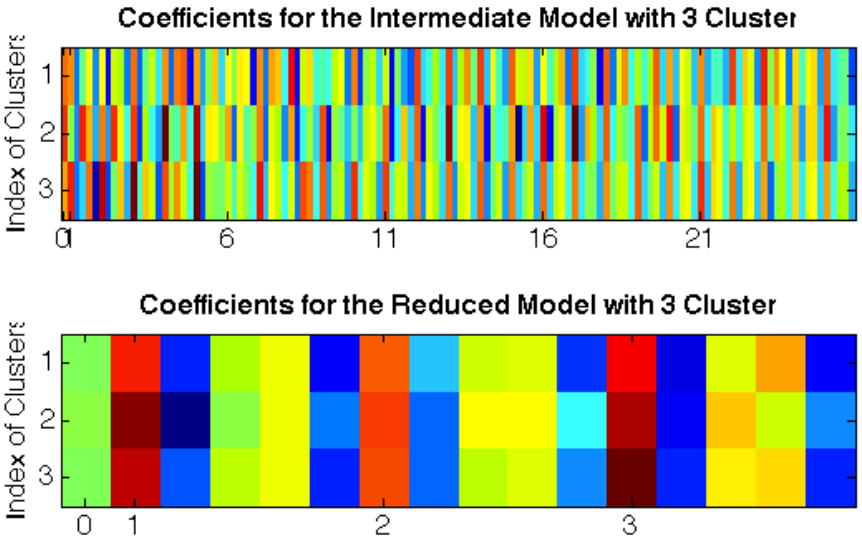


Figure 6.6: The 3-cluster aggregation results: the model coefficients of the intermediate models (upper) and reduced models (bottom).

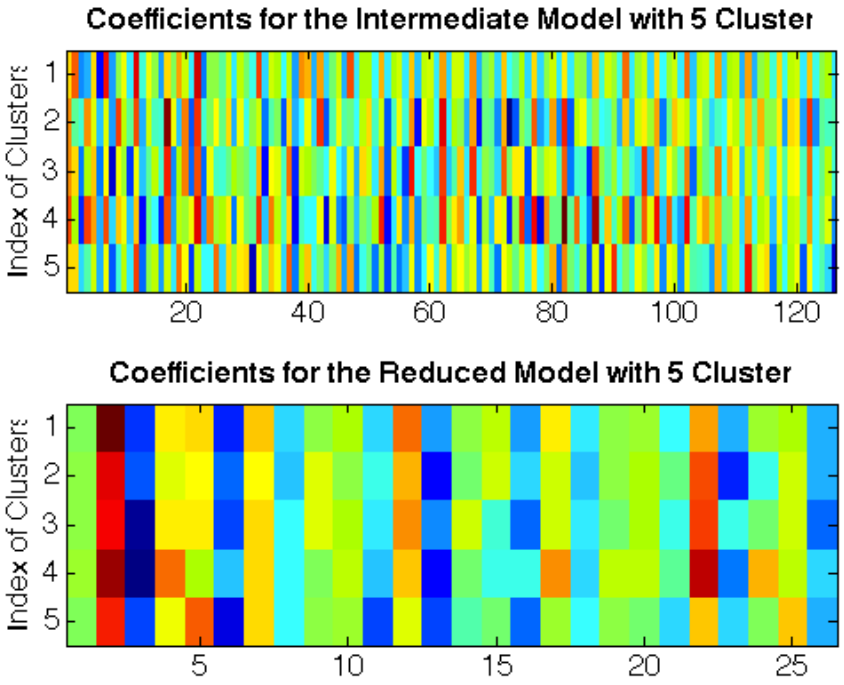


Figure 6.7: The 5-cluster aggregation results: the model coefficients of the intermediate models (upper) and reduced models (bottom).

the model prediction capacity achieved by including the information of process j [7,37,50]. This is a computationally expensive step, and the resulting causal relations are sometimes more detailed than needed. Suppose, for example, there are K point processes under consideration, then the number of parameters we need to estimate is proportional to K^2 and the number of causality tests required is K^2. If after aggregation, the resulting network consists of $K'(< K)$ processes of cluster representatives, then implementing K'^2 causality analyses on a network of representatives, and estimating the model parameters will provide a significant improvement in computational cost.

Summary of This Chapter

Our main achievements in this chapter are summarized as follows.

1. Recasting of systems with interconnected stochastic processes as graphs, whose edge weight matrices are defined using the parameters of the parametric models for every process; proposition of a node dissimilarity function that accounts for node properties and edge connections.

2. Application of the graph aggregation framework to identify ensemble connectivities in the system through aggregation (Section 6.2.3).

3. Example study on earthquake activities described by parametric models at different locations (Section 6.3).

Part II

Dynamic Coverage Problems

Chapter 7

Dynamic Clustering and Coverage Control

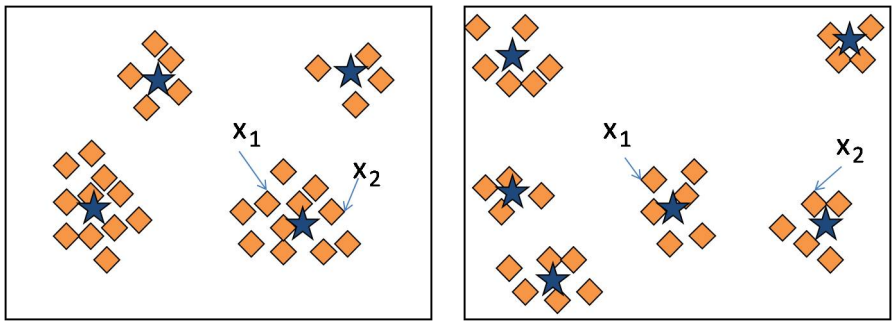
In Part I, we address two graph aggregation problems - the aggregation of graphs (with the Markov chain aggregation as a special example), and networks of interconnected stochastic processes described by parametric models. The main difference between these problems are essentially in terms of the choice of the distance functions that characterize the dissimilarity between two nodes. Note that the solution approach in both cases is comprised of transforming these problems as resource allocation (or clustering of nodes) problems and then applying the clustering framework described in Section 2.2. In essence these problems reduce to clustering of objects characterized by static vectors (the connectivity and the parameter vectors).

In this chapter, we develop a method for clustering of *dynamic* objects, that is where the vectors characterizing the objects vary with time. Specifically, we consider the problem of clustering a set of *moving* objects (nodes), each node has non-fixed locational coordinates over time. For ease of illustration, we consider a geographical sensor coverage problem, where the goal is to detect and track a group of moving nodes (or *nodes*) in an area with a small number of autonomous vehicles (or *resources*) equipped with sensors. The locations and dynamics of the vehicles must be determined, such that the sensors continuously provide adequate *coverage* of the moving nodes at all times. That is the sensors should track clusters of the moving nodes when these nodes may change cluster associations, and the clusters themselves can split and rejoin over time. (See Figure 7.1.)

7.1 Problem Formulation

7.1.1 Problem Setting

Let $\Omega \subset \mathbb{R}^2$ be a compact domain of interest; we note that the results directly extend to more general domains and higher dimensions. Suppose there are N mobile nodes denoted by s and M



(a) (b)

Figure 7.1: Clustering moving nodes. (a) and (b) denote two snapshots of an area with dynamic nodes (squares) and resources (stars). nodes x_1 and x_2 reside in the same cluster at the time instance shown in (a). A split occurs and causes them to reside in different clusters at the time instance shown in (b).

resources denoted by r ($M \ll N$), and the time horizon is $[0, +\infty)$. For arbitrary $t \geq 0$, let $z_i(t)$ and $y_j(t) \in \mathbb{R}^2$ denote the locations of the i^{th} node and the j^{th} resource, respectively, then their derivatives $\dot{z}_i(t)$ and $\dot{y}_j(t)$ represent their instantaneous velocities. Let the nodes move under a prescribed continuously differentiable acceleration field $\gamma(z_1, \ldots, z_N, \dot{z}_1, \ldots, \dot{z}_N) = [\gamma_1^T, \gamma_2^T, \ldots, \gamma_N^T]^T \in \mathbb{R}^{2N}$, with $\gamma_i \in \mathbb{R}^2$ representing the acceleration of the i^{th} node. Note that we have dropped the dependence on "(t)" for notational convenience unless required for clarification. Similarly, the resources move under a controlled acceleration field $u(t) = [u_1^T, u_2^T, \ldots, u_M^T]^T \in \mathbb{R}^{2M}$, which is to be designed. Thus, we can represent the combined dynamics of all nodes and resources by the state space equations:

$$\left.\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \gamma(x_1(t), x_2(t)) \\ \dot{x}_3(t) &= x_4(t) \\ \dot{x}_4(t) &= u(x_1(t), x_2(t), x_3(t), x_4(t)) \end{aligned}\right\} \Rightarrow \dot{\xi} = f(\xi, \gamma, u), \tag{7.1}$$

where $x_1(t) = [z_1^T(t),\ z_2^T(t), \ldots, z_N^T(t)]^T \in \mathbb{R}^{2N}$, $x_3(t) = [y_1^T(t),\ y_2^T(t), \ldots, y_M^T(t)]^T \in \mathbb{R}^{2M}$, and the compact system equation is defined by state variable $\xi \triangleq [x_1^T, x_2^T, x_3^T, x_4^T]^T$ and $f(\xi, \gamma, u) = [x_2^T, \gamma^T, x_4^T, u^T]^T$. We also denote $x_{k_i}(t) = [x_{k_{2i-1}}(t), x_{k_{2i}}(t)]^T \in \mathbb{R}^2$ for $k = 1, \ldots, 4$.

7.1.2 Coverage measure

We adopt the concept of *distortion* for the general clustering problem overviewed in Section 2.2 to measure coverage at a particular time instant (the lower the distortion, the better the coverage), and adapt this to a dynamic setting. For any time t, the distortion resulting from using a set $\{r_j\}$ to cover a set $\{s_i\}$ has the general form given in (2.6), that is the average distance of a node to its nearest resource, given by

$$D(s(t), r(t)) = \sum_{i=1}^{N} \mu_i \min_{1 \le j \le M} d(s_i(t), r_j(t)) \tag{7.2}$$

where $s_i(t) \stackrel{\triangle}{=} [z_i^T(t), \dot{z}_i^T(t)]^T = [x_{1_i}^T(t), x_{2_i}^T(t)]^T \in \mathbb{R}^4$, $r_j(t) \stackrel{\triangle}{=} [y_j^T(t), \dot{y}_j^T(t)]^T = [x_{3_j}^T(t), x_{4_j}^T(t)]^T \in \mathbb{R}^4$, $d(s_i(t), r_j(t)) \in \mathbb{R}_+$ denotes the distance between the i^{th} node s_i and j^{th} resource r_j, and μ_i is a given positive constant (without loss of generality, we assume $\sum_i \mu_i = 1$) that denotes relative weight of the ith node. The distortion is defined for every time instance t due to the implicitly assumed dynamics. For static coverage problems, $d(s_i, r_j)$ is typically chosen to be the squared Euclidean distance $\|x_{1_i} - x_{3_j}\|^2$ to reflect the geometrical distance. The resource locations, y_j's (or x_3), that minimize (7.2) are at the "centroid" of the clusters of nodes z_i's (or x_1) [55, 56, 60].

In a dynamic setting, we desire the resources to dynamically cover the clusters, that is, each resource r_j should not only reach the instantaneous centroid of the cluster it represents at a particular time t, but should also consider the heading (velocity) of the cluster. For instance, a resource that is at the position centroid of a cluster but has a different velocity (magnitude or direction) than the average velocity of nodes in the cluster cannot be thought of as *covering* (or *tracking*) the cluster. Accordingly, we define the node-resource distance by

$$d(s_i, r_j) = \|x_{1_i} - x_{3_j}\|^2 + \theta \|x_{2_i} - x_{4_j}\|^2, \tag{7.3}$$

where the weighting constant $\theta \ge 0$ characterizes the emphasis on the locational clustering relative to the velocity-based clustering. With this distance function, covering means that for every resource j, $r_j \stackrel{\triangle}{=} [x_{3_j}^T,\ x_{4_j}^T]^T$ is close to the corresponding *cluster centroid position* and *cluster average velocity* (defined and quantified in Section 7.2). Therefore, to maximize the coverage, we need to minimize

the distortion $D(s(t), r(t))$ in (7.2) for all time t.

As in the static case in Section 2.2, we modify the instantaneous distortion D to be a soft distortion function by associating every node i to every resource j through the association weight $p_{j|i}$ [55]) by

$$\tilde{D}(s, r) = \sum_{i=1}^{N} \mu_i \sum_{j=1}^{M} p_{j|i} d(s_i, r_j). \tag{7.4}$$

We use the MEP as in Chapter 2 and [55,60] to choose the association weights $p_{j|i}$ by minimizing $\tilde{D}$ with entropy constraint $H(r|s) = -\sum_{i=1}^{N} \mu_i \sum_{j=1}^{M} p_{j|i} \log(p_{j|i}) = H_0$. Then apply the DA algorithm to control the trade-off between decreasing the local influence in the hard clustering formulation and the deviation of $\tilde{D}$ from the original distortion D in (7.2). The process of the DA algorithm is driven by an annealing parameter β that geometrically increases from a very small value (≈ 0) to a sufficiently large value. It has been shown that the optimal association weights at any annealing stage β obtained by minimizing the free energy $F = \tilde{D} - \frac{1}{\beta} H$ with respect to $\{p_{j|i}\}$ has the form of the Gibbs distribution as in (2.12),

$$p_{j|i} = \frac{\exp\{-\beta d(s_i, r_j)\}}{\sum_{k=1}^{M} \exp\{-\beta d(s_i, r_k)\}}.$$

with $d(s_i, r_j)$ being defined as in (7.3). And the free energy F simplifies as

$$F(r) = -\frac{1}{\beta} \sum_{i=1}^{N} \mu_i \log \sum_{k=1}^{M} \exp\{-\beta d(s_i, r_k)\}. \tag{7.5}$$

Therefore, we have transformed the problem of minimizing instantaneous distortion $D(s(t), r(t))$ as in (7.2) to an optimal assignment of resource locations and velocities that minimizes the free energy (7.5) where d is as specified in (7.3).

7.1.3 Iterative solution based on MEP

The incorporation of the annealing process in the DA algorithm is along the lines of the graph clustering framework as presented in Chapter 4. We specify this process for locational clustering here.

Due to the one-to-one correspondence between the value of parameter β and the level of randomness H_0 we grant to all the associations (see Section 2.2 for details). As in the graph aggregation, we apply the DA algorithm such that the free energy minimization problem is iteratively solved at a succession of values $\beta = \beta_k$, where β_k increases with k. When $\beta \to 0$, this corresponds to entropy maximization, (a convex problem), and for $\beta \to \infty$, the association weights (7.5) become nearly binary and the free energy $F \approx \tilde{D} \approx D$ in (7.2). Therefore the algorithm is insensitive to the initial resource allocations and eventually achieves an allocation that closely approximates a solution to (7.2).

(1) First order condition

The first order necessary condition of (7.5) (i.e., $\nabla_{(x_{3_j}, x_{4_j})} F = 0$) yields the stationary point

$$x^*_{3_j} = \sum_{i=1}^{N} q_{i|j} x_{1_i}; \quad x^*_{4_j} = \sum_{i=1}^{N} q_{i|j} x_{2_i}, \quad j = 1, \cdots, N \tag{7.6}$$

$$\text{where } \; q_{i|j} = [p_{j|i}\mu_i]/\omega_j \; \text{ and } \; \omega_j = \sum_{i=1}^{N} p_{j|i}\mu_i$$

represent the posterior and total weight of a resource respectively. These equations imply that x^*_3 and x^*_4 are the *weighted centroids* of all nodes, where the weights are specified by $\{q_{i|j}\}$. When the positions and velocities of all resources satisfy the centroid equations (7.6), we say the resources attain the *cluster centroids* and denote these states as x^c_3 and x^c_4. That is, $x^c_{3_j}$ and $x^c_{4_j}$ satisfy the following implicit equations:

$$x^c_{3_j} = \sum_{i=1}^{N} q^*_{i|j} x_{1_i}, \quad x^c_{4_j} = \sum_{i=1}^{N} q^*_{i|j} x_{2_i}, \tag{7.7}$$

$$\text{and } p^c_{j|i} = \frac{\exp\left\{-\beta d(s_i, r^c_j)\right\}}{\sum_k \exp\left\{-\beta d(s_i, r^c_k)\right\}}, \quad \forall j.$$

(2) Phase transitions

The first order solution (7.7) is a local minimum of the free energy only if $F(r^*)$ is locally convex at $r = r^* = [x^*_3; x^*_4]$, which can be determined based on the sign of the Hessian $\nabla^2_r F$.

When the distance is given by (7.3), the $\nabla^2_r F$ is positive definite except at a discrete set of

values β_c, referred to as the *critical values*. The explicit expression of critical values are given by

$$\beta_c^{-1} = 2\lambda_{max}(C_{s|r_j}), \quad \text{for some } 1 \leq j \leq M,$$

where $\lambda_{max}(\cdot)$ represents the largest eigenvalue, and $C_{s|r_j} \triangleq \sum_{i=1}^{N} q_{i|j}(s_i - r_j)(s_i - r_j)^T)$. When β surpasses β_c, a local minimum (which consists of a set of r_j^*) of the cost function F becomes unstable and *bifurcation* of the cost surface occurs along some eigen-vector direction of the Hessian. Therefore, the number of *distinct* resource locations when $\beta > \beta_c$ is always greater than the number when $\beta < \beta_c$, and this is called the *phase transition*. Details can be found in [55].

(2) Insensitivity to annealing parameter β

We also show that, when the value of β is far from β_c, the solution (7.6) is insensitive to the changes in the values of β (see the *sensitivity-to-temperature property* derived in [60] for quantitative results).

Therefore in implementation, between two consecutive values of β_c, the number and assignment of the resources are unchanged as the β value is increased. In other words, if in the dynamic case the state of some resource (location or velocity) is not at the centroid, that is does not satisfy (7.6), that is due to the node dynamics rather than the change of β value. We exploit this insensitivity and choose not to change β except near the critical conditions, thus greatly reducing the number of potential computations. Since tracking cluster centers is necessary for the occurrence of critical conditions, β remains constant while the resources are far from the cluster centers. Once the cluster centers are reached, we increase β to effect the critical condition.

Here we assume that the time required to implement temperature changes in the algorithm is negligible compared to the time constant of the node dynamics. When β surpasses the β_c level, the resource(s) *split(s)*, that is the number of distinct resource locations increases and therefore more resources are required to track the minima of F. It should be noted that the algorithm induces *splitting* only when identification of *finer* clusters is sought. If finer clusters are not sought, β values need not be changed thus avoiding adding new resources (See [60] for details).

7.2 Control Design for the DME Framework

As proposed in the DME framework in [60] and [72], we increase β only after the resources adequately track the cluster centers and finer clustering is sought. Therefore, we focus on designing control laws that drive the resources to pursue the cluster centers. In [60], it has been assumed that all nodes are moving under a velocity field; and a form of *static* feedback control law that governs the resource velocities has been presented to provide progressively better resource coverage. This is sufficient to drive all resources towards the cluster centers when the node dynamics are defined by velocities. However, when we consider complicated and more realistic acceleration (or force) based dynamics (7.1), and seek a control law to guarantee asymptotic tracking of the cluster centroids, dynamic control design becomes insufficient. For this case we prove necessity of *dynamic* feedback control laws. We make the following assumptions regarding the node dynamics and cluster mass:

Assumption 4 (Smoothness Assumption) *The node dynamics $\gamma(\xi)$ in* (7.1) *are continuously differentiable.*

Assumption 5 (Positive Mass Assumption) *All clusters have* positive mass*, that is, for all j, the weight of resource r_j is bounded away from zero, or $p(r_j) \geq \varepsilon > 0$ for a constant ε.*

The tracking objective can be interpreted as $x_3 \to x_3^c$ and $x_4 \to x_4^c$ as $t \to \infty$, with x_3^c and x_4^c being the instantaneous cluster position and velocity centers, as defined in (7.7). In other words, for a fixed value of β, we want to minimize the free energy (7.5) (thus minimize the soft distortion (7.4)) at all times. To achieve the goal of asymptotic tracking, or all asymptotically approaching the corresponding cluster centroids, we choose the control input u (actuation force applied to each resource) in (7.1) that renders the free energy decreasing with time, that is $\dot{F} = \frac{d}{dt}F(t) \leq 0$, whenever all cluster centers are not attained ($\left[x_3;\, x_{4_j}\right] \neq [x_3^c;\, x_4^c]$). Once all cluster centers are reached, our aim then is to split the resources (by taking advantage of the phase transition property) if coverage with finer resolution is sought. The derivative of the free energy function (7.5) is given

by:

$$\dot{F} = \left(\frac{\partial F}{\partial \xi}\right)^T \dot{\xi} \;=\; 2\xi^T \gamma f(\xi, \gamma, u),$$

$$\text{where } \gamma \;=\; \begin{bmatrix} \mathbf{P}_{1\otimes} & & -\mathbf{P}_{12\otimes} & \\ & \theta\mathbf{P}_{1\otimes} & & -\theta\mathbf{P}_{12\otimes} \\ -\mathbf{P}_{12\otimes}^T & & \mathbf{P}_{2\otimes} & \\ & -\theta\mathbf{P}_{12\otimes}^T & & \theta\mathbf{P}_{2\otimes} \end{bmatrix}, \tag{7.8}$$

with $\mathbf{P}_1 \triangleq \text{diag}(\mu_1, \cdots, \mu_N) \in \mathbb{R}^{N\times N}$, $\mathbf{P}_2 \triangleq \text{diag}(\omega_1, \cdots, \omega_M) \in \mathbb{R}^{M\times M}$ and $[\mathbf{P}_{12}]_{i,j} = \mu_i p_{j|i} = \omega_j q_{i|j}$ denoting the joint association weights. We further define matrices $\mathbf{Q}_1, \mathbf{Q}_2 \in \mathbb{R}^{N\times M}$, where $[\mathbf{Q}_1]_{i,j} = p_{i|j}$ and $[\mathbf{Q}_2]_{i,j} = q_{i|j}$. These matrices satisfy $\mathbf{P}_{12} = \mathbf{P}_1\mathbf{Q}_1 = \mathbf{Q}_2\mathbf{P}_2$. As is standard, the notation $\otimes$ denotes the Kronecker product, and $\mathbf{P}_{k\otimes} \triangleq \mathbf{I}_2 \otimes \mathbf{P}_k$ for $k \in \{1, 2, 12\}$.

7.2.1 Necessity of dynamic dontrol

We first show that *static* feedback control laws are *not* sufficient to achieve the tracking objective in an acceleration-driven setting. So constructing control inputs similar to those in [60] does not work in this setting, which can be made evident as follows. We rewrite the derivative $\dot{F}$ to be affine in control u as

$$\dot{F}(\xi) = \underbrace{2x_1^T\mathbf{P}_{1\otimes}(x_2 - \mathbf{Q}_{1\otimes}x_4) - 2x_3^T P_{2\otimes}(\mathbf{Q}_{2\otimes}^T x_2 - x_4) + 2\theta(x_2 - \mathbf{Q}_1 x_4)^T\mathbf{P}_1\gamma}_{a(\xi)}$$

$$+ \underbrace{2\theta(x_4 - \mathbf{Q}_{2\otimes}^T x_2)^T P_{2\otimes}}_{b(\xi)} u.$$

Since $\|b(\xi)\| = 2\theta\|\mathbf{P}_2(x_4 - \mathbf{Q}_{2\otimes}^T x_2)\| \geq 2\theta\varepsilon\|x_4 - \mathbf{Q}_{2\otimes}^T x_2\|$ from *Assumption 5*, the control u affects $\dot{F}(\xi)$ only when the velocity centroid has not been reached (i.e., when $b(\xi) \neq 0$ or $x_4 \neq x_4^c$). That is, the control u becomes ineffective when all resource velocities attain the cluster centroid velocities, regardless of their positions. Therefore the static control laws are insufficient for tracking cluster centers with these dynamics, for example, for initial conditions that satisfy $x_4 = x_4^c$, but $x_3 \neq x_3^c$.

7.2.2 Construction of a dynamic control law

We propose a dynamic feedback control law in the form,

$$\dot{x}_5(t) = v(t), \quad \text{and} \quad u(t) = x_5(t), \tag{7.9}$$

where we have introduced a new state variable x_5 and v is the new design parameter. With this extension, the augmented system equation becomes $\dot{\xi}_{cl}(t) = g(\xi_{cl}, \gamma, v)$, where $\xi_{cl} \triangleq [x_1^T\ x_2^T\ x_3^T\ x_4^T\ x_5^T]^T$ denotes the augmented closed-loop states, and $g = [x_2^T\ \gamma^T\ x_4^T\ x_5^T\ v^T]^T$. To complete our control design, we rewrite $\dot{x}_3 = \zeta + (x_4 - \zeta)$ and $\dot{x}_4 = \eta + (x_5 - \eta)$, where ζ and η are independently designed to control the states x_3 and x_4. We then design v in (7.9) to drive $(x_4 - \zeta)$ and $(x_5 - \eta)$ to 0. We use the following *augmented energy function* as the control Lyapunov function

$$V(\xi_{cl}) = F(\xi) + W(x_5) + \frac{1}{\beta}\log M, \tag{7.10}$$

in which F is the free energy given in (7.5) and $W(x_5) \triangleq \frac{1}{2}x_5^T x_5$. We seek v that makes $\dot{V} \leq 0$ for all $[x_3; x_4] \neq [x_3^c; x_4^c]$ and use the LaSalle theorem to prove the asymptotic tracking.

Lemma 8 *The augmented energy function $V(\xi_{cl}) \geq 0$ for all ξ_{cl}.*

The nonnegativity of V is easy to verify since

$$V(\xi_{cl}) \geq F(\xi) + \frac{1}{\beta}\log M = \frac{1}{\beta}\sum_i \mu_i \log\left\{\frac{M}{\sum_{j=1}^M \exp(-\beta d(s_i, r_j))}\right\} \geq 0.$$

The equalities hold only when $x_5 = 5$ and $s_i = r_j$ for all $i = 1, \ldots, N$ and $j = 1, \ldots, M$, that is when all nodes and resources are coincident.

The time derivative $\dot{V}$ is affine in v

$$\dot{V}(\xi_{cl}) = \nabla_{\xi_{cl}} V(\xi_{cl})^T \dot{\xi}_{cl} = \underbrace{a_1(\xi) + b_1(\xi)^T \begin{bmatrix} \zeta(\xi) \\ \eta(\xi) \end{bmatrix}}_{\dot{V}_1(\xi)} + \underbrace{a_2(\xi_{cl}) + x_5^T v(\xi_{cl})}_{\dot{V}_2(\xi_{cl})}, \tag{7.11}$$

where

$$a_1(\xi) \stackrel{\triangle}{=} 2 \begin{bmatrix} x_1 - \mathbf{Q}_1 x_3 \\ \theta(x_2 - \mathbf{Q}_1 x_4) \end{bmatrix}^T \mathbf{P}_1 \begin{bmatrix} x_2 \\ \gamma \end{bmatrix}, \tag{7.12}$$

$$b_1(\xi) \stackrel{\triangle}{=} 2\mathbf{P}_2 \begin{bmatrix} x_3 - \mathbf{Q}_2^T x_1 \\ \theta(x_4 - \mathbf{Q}_2^T x_2) \end{bmatrix} \text{ and } a_2(\xi_{cl}) \stackrel{\triangle}{=} b_1(\xi)^T \begin{bmatrix} x_4 - \zeta \\ x_5 - \eta \end{bmatrix}.$$

We exploit the affine structure in (7.11) to design ζ, η, and v that guarantee $\dot{V} \leq 0$ whenever the resources are not at the cluster centers (i.e., $b_1(\xi) \neq 0$) and thus achieve asymptotic tracking. One such design can be constructed using the formulation proposed by Sontag in [63], giving us

$$\begin{bmatrix} \zeta \\ \eta \end{bmatrix} = \begin{cases} -\left[k_1 + \frac{a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2}}{b_1^T b_1}\right] b_1 & \text{if } b_1 \neq 0 \\ 0 & \text{if } b_1 = 0 \end{cases}, \tag{7.13}$$

$$\text{and } \; v = \begin{cases} -\left[k_2 + \frac{a_2 + \sqrt{a_2^2 + (x_5^T x_5)^2}}{x_5^T x_5}\right] x_5 & \text{if } x_5 \neq 0 \\ 0 & \text{if } x_5 = 0, \end{cases} \tag{7.14}$$

where k_1 and k_2 are arbitrary positive constants. We then show that by this constructive design, the asymptotic tracking goal (or $[x_3; x_4] \to [x_3^c; x_4^c]$ as $t \to \infty$) is accomplished .

Lemma 9 *Let $\phi : \mathbb{R} \to \mathbb{R}$ be a nonnegative function of bounded variation on $[0, \infty)$. Suppose that $\lim_{t\to\infty} \int_{r=0}^{t} \phi(r)dr$ exists and is finite, then $\lim_{t\to\infty} \phi(t) = 0$.*

Proof: Suppose $\lim_{t\to\infty} \phi(t) \neq 0$, then there exists a $\delta > 0$ and a sequence $\{t_n\}$ with $t_n - t_{n-1} > 1$, such that $\phi(t_n) > \delta$ for all n. Moreover, since $s(t) \stackrel{\triangle}{=} \int_{r=0}^{t} \phi(r)dr$ is increasing, and $s(t) \leq \bar{s} \stackrel{\triangle}{=} \lim_{t\to\infty} s(t) < \infty$ as assumed, there exists a subsequence $\{t_{n_k}\} \subset \{t_n\}$, such that $|s(t_{n_k}) - \bar{s}| < \frac{1}{2}\delta, \forall k$. Let $\check{\phi}(n_k) \stackrel{\triangle}{=} \min_{t_{n_k} \leq r \leq t_{n_{k+1}}} \phi(r)$, we have $\check{\phi}(n_k) < \check{\phi}(n_k)(t_{n_{k+1}} - t_{n_k}) \leq \int_{r=t_{n_k}}^{t_{n_{k+1}}} \phi(r)dr \leq |s(t_{n_k}) - \bar{s}| < \frac{1}{2}\delta$. Then $\frac{1}{2}\delta \leq \sup_{t_{n_k} \leq r_1, r_2 \leq t_{n_{k+1}}} |\phi(r_1) - \phi(r_2)|$. And the total variation of ϕ on interval $[0, t_N] \subset [0, \infty)$ is $T_0^{t_N} = \sup \sum_{i=1}^{N'} |\phi(r_i) - \phi(r_{i-1})| \geq \sum_{k=1}^{N} \phi(n_k) - \check{\phi}(r_k) \geq \frac{1}{2}\delta N$, in which the supremum is taken over all subdivisions $0 < r_1 < \cdots < r_{N'} = t_N$ with $N' \in \mathbb{N}$. This contradicts the bounded variation assumption. □

The following *Theorem 8* establishes that the control law u based on v in (7.9) asymptotically tracks cluster centroid locations and velocities.

Theorem 8 (Asymptotic tracking of clusters) For a system with node dynamics given by (7.1), under the *Assumptions 4* and *5*, the control law $\dot{u} = v(\zeta, \eta, x_5)$ with ζ, η and v given by (7.13) achieves asymptotic tracking. That is, $x_3 \to x_3^c$ and $x_4 \to x_4^c$ as $t \to \infty$ for all $j = 1, 2, \cdots, M$.

Proof: Note that $W(\xi_{cl}(t)) = \frac{1}{2} x_5(t)^T x_5(t) \geq 0$ for all ξ_{cl}, and $W(\xi_{cl}(t)) = 0$ only when $x_5(t) = 0$. Also $\dot{W}(\xi_{cl}(t)) = x_5(\xi_{cl}(t))^T v(\xi_{cl}(t)) \leq 0$ for all $t \geq 0$ since

$$\dot{W}(\xi_{cl}(t)) = \begin{cases} -k_2 x_5^T x_5 - \sqrt{a_2^2 + (x_5^T x_5)^2} - a_2 < 0 & \text{if } x_5(t) \neq 0 \\ 0 & \text{if } x_5(t) = 0 \end{cases}, \tag{7.15}$$

where a_2 is given in (7.12). As a consequence, the real-valued function $W(\xi_{cl}(t))$ converges to a finite value W_∞ as $t \to \infty$. Further, the smoothness of γ ensures that $\dot{W}$ is of bounded variation, and since $\int_{t_0}^{\infty} -\dot{W}(\xi_{cl}(t))dt = W(\xi_{cl}(t_0)) - W_\infty < \infty$, we have $\dot{W}_\infty \triangleq \lim_{t\to\infty} \dot{W}(\xi_{cl}(t)) = 0$ from *Lemma 9*. Therefore, from (7.15), $\lim_{t\to\infty} x_5(t) = 0$. Note that this result holds for any choice of positive constant k_1 in (7.13).

Now $x_5(t) \to 0$ can be studied under the following two cases:

Case 1: $x_5(t^*) = 0$ at some finite time t^*,

Case 2: $x_5(t) \neq 0$ at any finite time t.

We show that in both cases, as $t \to \infty$, $b_1(\xi(t)) \to 0$, or equivalently $x_3 \to x_3^c$ and $x_4 \to x_4^c$:

Case 1: $x_5(t^*) = 0$ for $t^* < \infty$.

$x_5(t^*) = 0$ implies $v(t^*) = 0$ by (7.14), $W(\xi_{cl}(t^*)) = 0$, and $\dot{W}(\xi_{cl}(t^*)) = 0$ by (7.15). Then we conclude that $x_5(t) \equiv 0$ for all $t > t^*$, since otherwise, if $x_5(t') \neq 0$ for $t' > t^*$, $W(\xi_{cl}(t')) = \frac{1}{2}\|x_5(t')\|^2 > 0$, which yields $\dot{W} > 0$ for some $t^* < t < t'$ and contradicts $\dot{W} \leq 0$ for all t (see (7.15)). Therefore, for all $t > t^*$, $\dot{x}_4 = x_5 \equiv 0$, implying $x_4 \equiv c_4$ for some constant vector c_4.

We then claim that there exists a time $T > t^*$, such that $a_2(\xi_{cl}(t)) = 0$ for all $t > T$. Since for all $t > t^*$, $x_5(t) = 0$ and $x_4(t) = c_4$, we can make $a_2(\xi_{cl}(t)) = a_2|_{x_5=0, x_4=c_4} = b_1^T c_4 + k_1 b_1^T b_1 + a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2} \geq 0$ by choosing a sufficiently large constant k_1. If this claim is false, we can find a sequence of time instances $\{t_k\} > t^*$ with $t_k \to \infty$ as $k \to \infty$ such that $a_2(\xi_{cl}(t_k)) > 0$, since a_2

is a continuous and bounded function of t. However, in this case, using (7.15) we can show that $\dot{W}(\xi_{cl}(t_k))|_{x_5=0} = -|a_2(t_k)| - a_2(t_k) = -2a_2(t_k) < 0$, which contradicts $\dot{W}(\xi_{cl}(t)) = 0$ for all $t > t^*$. Therefore $a_2(\xi_{cl}(t)) = 0$ for all $t > t^*$. Note that since this equality is achieved only when $b_1(t) = 0$, we conclude that $b_1(t) = 0$ for all $t > t^*$.

Case 2: $x_5 \to 0$ as $t \to \infty$ with $x_5(t) \neq 0$ for all $t < \infty$.

If $b_1(\hat{t}) = 0$ for some finite $\hat{t}$, then the cluster centroid is attained at the time instance $\hat{t}$, and no control authority is needed to *improve* tracking.

If $b_1(t) \neq 0$ for all finite t, that is, not all resources simultaneously track the cluster centers in finite time, then

$$\dot{V}(\xi_{cl}) = -k_1 b_1^T b_1 - \sqrt{a_1^2 + (b_1^T b_1)^2} - k_2 x_5^T x_5 - \sqrt{a_2^2 + (x_5^T x_5)^2} < 0, \forall t < \infty,$$

thus $\dot{V}(\xi_{cl}(t)) \to 0$ as $t \to \infty$ (from *Lemma 9*). Since $k_1 > 0$, $0 \leq k_1 b_1^T b_1 \leq |\dot{V}(\xi_{cl})|$, we then have $\lim_{t\to\infty} b_1(t) = 0$. □

In the next theorem we show the control law (7.13) is non-conservative, that is, if there exists a Lipschitz control that makes $\dot{V} \leq 0$, then our design achieves the same.

Theorem 9 (Boundedness of control) If there exists Lipschitz functions $\hat{v}, \hat{\zeta}$ and $\hat{\eta}$ that asymptotically track cluster centers, that is, $\dot{V}_1(\xi) = a_1(\xi) + b_1(\xi)\left[\hat{\zeta}^T\ \hat{\eta}^T\right]^T \leq 0$ and $\dot{V}_2(\xi_{cl}) = a_2(\xi_{cl}) + x_5^T \hat{v} \leq 0$ (see (7.11) for the definition of V_1 and V_2) whenever $\xi_{cl} \neq \xi_e \triangleq \left[x_1^T\ x_2^T\ x_3^{cT}\ x_4^{cT} 0\right]^T$, then ζ, η and v given in (7.13) and (7.14) also track the cluster centers and are Lipschitz.

Proof: Since $\hat{\zeta}, \hat{\eta}$ and $\hat{v}$ are Lipschitz at ξ_e by assumption, there exists an $r_1 > 0$ and $\delta > 0$ such that $\|[\hat{\zeta}^T\ \hat{\eta}^T\ \hat{v}^T]^T\| \leq r_1\|\xi_{cl} - \xi_e\|$ for all $\xi \in \Omega_\delta \triangleq \{\xi_{cl} : \|\xi_{cl} - \xi_e\| < \delta\}$, thus $\max\{\|(\hat{\zeta}^T\ \hat{\eta}^T)^T\|, \|\hat{v}\|\} \leq r_1\|\xi_{cl} - \xi_e\|$ in Ω_δ.

Case 1: $a_1(\xi) > 0$ and $b_1 \neq 0$. Since $\dot{V}_1(\hat{\zeta}, \hat{\eta}) \leq 0$ by assumption, then $a_1(\xi) < -b_1\left[\hat{\zeta}^T\ \hat{\eta}^T\right]^T \leq r_1\|b_1\|\|\xi_{cl} - \xi_e\|$ in Ω_δ (7.11). This implies $\eta \triangleq \frac{a_1}{\|b_1\|^2} \leq \frac{r_1\|\xi_{cl}-\xi_e\|}{\|b_1\|}$. So we have $0 < \frac{a_1+\sqrt{a_1^2+(b_1^T b_1)^2}}{b_1^T b_1} + k_1 \leq k_1 + 1 + 2\eta \leq 1 + k_1 + 2\frac{r_1\|\xi_{cl}-\xi_e\|}{\|b_1\|}$. For ζ and η selected by (7.13) when $b_1 \neq 0$, we have $\|\left[\zeta^T\ \eta^T\right]\| \leq \|k_1 + \frac{a_1+\sqrt{a_1^2+\|b_1\|^4}}{\|b_1\|^2}\| \cdot \|b_1\| \leq (k_1+1)\|b_1\| + 2r_1\|\xi_{cl} - \xi_e\| \leq [2r_1 + (k_1+1)r_2]\,\|\xi_{cl} - \xi_e\|$, where in the last inequality, we use the fact $\|b_1\| \leq \|\mathbf{P}_2\| \cdot \left\| \begin{bmatrix} x_3 - \mathbf{Q}_2^T x_1 \\ \theta(x_4 - \mathbf{Q}_2^T x_2) \end{bmatrix} \right\| \leq r_2\|\xi_{cl} - \xi_e\|$, for

some $r_2 > 0$. Let $r_3 \stackrel{\triangle}{=} [2r_1 + (k_1 + 1)r_2]$.

Case 2: $a_1(\xi) \leq 0$ and $b_1 \neq 0$, we have $0 \leq a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2} \leq a_1 + |a_1| + \|b_1\|^2 = \|b_1\|^2$, and $\| \left[\hat{\zeta}^T \ \hat{\eta}^T\right] \| \leq (1 + k_1)\|b_1\| \leq (1 + k_1)r_2\|\xi_{cl} - \xi_e\|$. Let $r_4 \stackrel{\triangle}{=} (1 + k_1)r_2$.

In both cases, ζ, η as defined in (7.13) are Lipschitz with constant $R \stackrel{\triangle}{=} \max\{r_3, r_4\}$ in Ω_δ. v is similarly shown to be Lipschitz. Since v is Lipschitz in Ω_δ and bounded outside Ω_δ from (7.13), the boundedness of control follows. □

Note that a_1, a_2, b_1, x_3^c and x_4^c in *Theorem 9* are defined in (7.12) and (7.7). The proof of *Theorem 9* is analogous to the proof of Proposition 3.43 in [58], in which we exploit the algebraic structure of the control defined by (7.13) and (7.14). Once the tracking cluster objective is achieved, we exploit the phase-transition property to effect resource splitting (desired) to achieve better coverage. If no higher resolution coverage is sought, resources continue to track the cluster centroids.

7.3 Simulation

We consider a scenario with 64 mobile nodes, each of which has the same weight, and which comprise four natural clusters (Figure 7.2 and Figure 7.3). We choose node dynamics such that all nodes have zero initial velocities, and nodes within one cluster have similar accelerations so that the clusters are maintained. The initial acceleration of each cluster is indicated by an arrow in plot (a); the individual node accelerations are generated by adding small random perturbations on those cluster accelerations. Figure 7.2 (a) - (h) record the featured time instances, in which plots (a) - (d) demonstrate that our DME algorithm seeks better coverage through resource splits until the coverage requirement is met. Note that a single resource location gradually splits into two, three, and then finally becomes four resource locations (that is all 4 clusters are successfully identified). Upon splitting, the instantaneous cluster centroid positions and velocities are assigned to the corresponding new resources. The subsequent deviations of the resources from the cluster centers due to initial mismatch of their accelerations from that of respective clusters, are corrected by using the tracking control in (15) and (16) (plots (d) - (g)).

Figure 7.3 (c) and (d) compare the computational effort for tracking of cluster centers of our DME algorithm with a frame-by-frame (FBF) method, where MEP-based static clustering is used in a predetermined sequence of sampled time instants. In this simulation, we applied the FBF

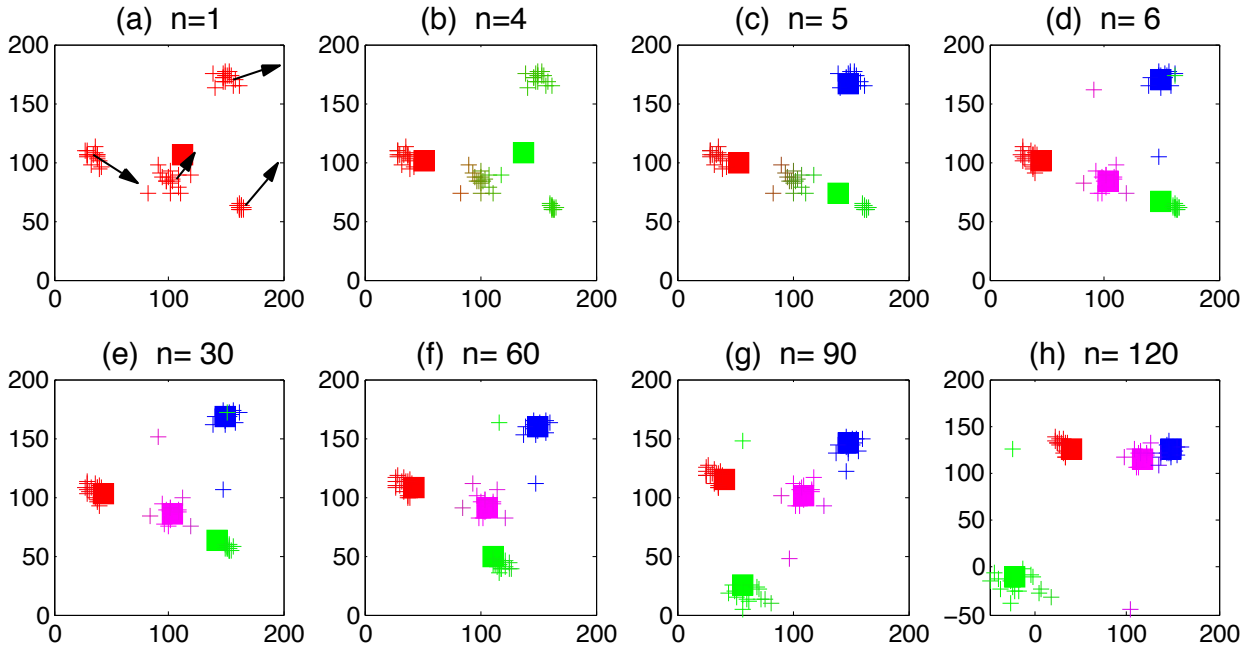


Figure 7.2: Plots (a) - (h) show snapshots of a domain with mobile nodes (plus symbols) and resources (square symbols). The nominal initial accelerations of natural clusters are given by $\gamma_{n1}(0) = [0.05; 0.12]$, $\gamma_{n2}(0) = [-0.02; 0.09]$, $\gamma_{n3}(0) = [-0.01; -0.17]$ and $\gamma_{n4}(0) = [-0.61; -0.29]$ respectively, whose directions and magnitudes are indicated by arrows and their lengths in plot (a). The accelerations of individual nodes are from normal distributions around those nominal values. At $n = 1$, the algorithm is initiated at a high temperature value ($T = 5000$) and places all resources at the velocity and position centroid of all nodes ($x_3(1) = x_3^c(1)$ and $x_4(1) = x_4^c(1)$) since all association weights $p_{j|i}$ are nearly uniform. Hence the centroid condition is automatically satisfied. To improve coverage, more distinct resource locations are added through phase transitions, i.e., by increasing the β value, until the resolution requirement is reached (plots (b) - (d)). The color of a node is determined by the average rgb value of all resources, weighted by the association weights. The algorithm progressively computes ζ, μ and v to adjust resource dynamics, and drives resources towards cluster centers (plots (e) - (h)), until all cluster centers are identified and tracked (plots (g), after 90 time steps). The temperature value remains the same during this period.

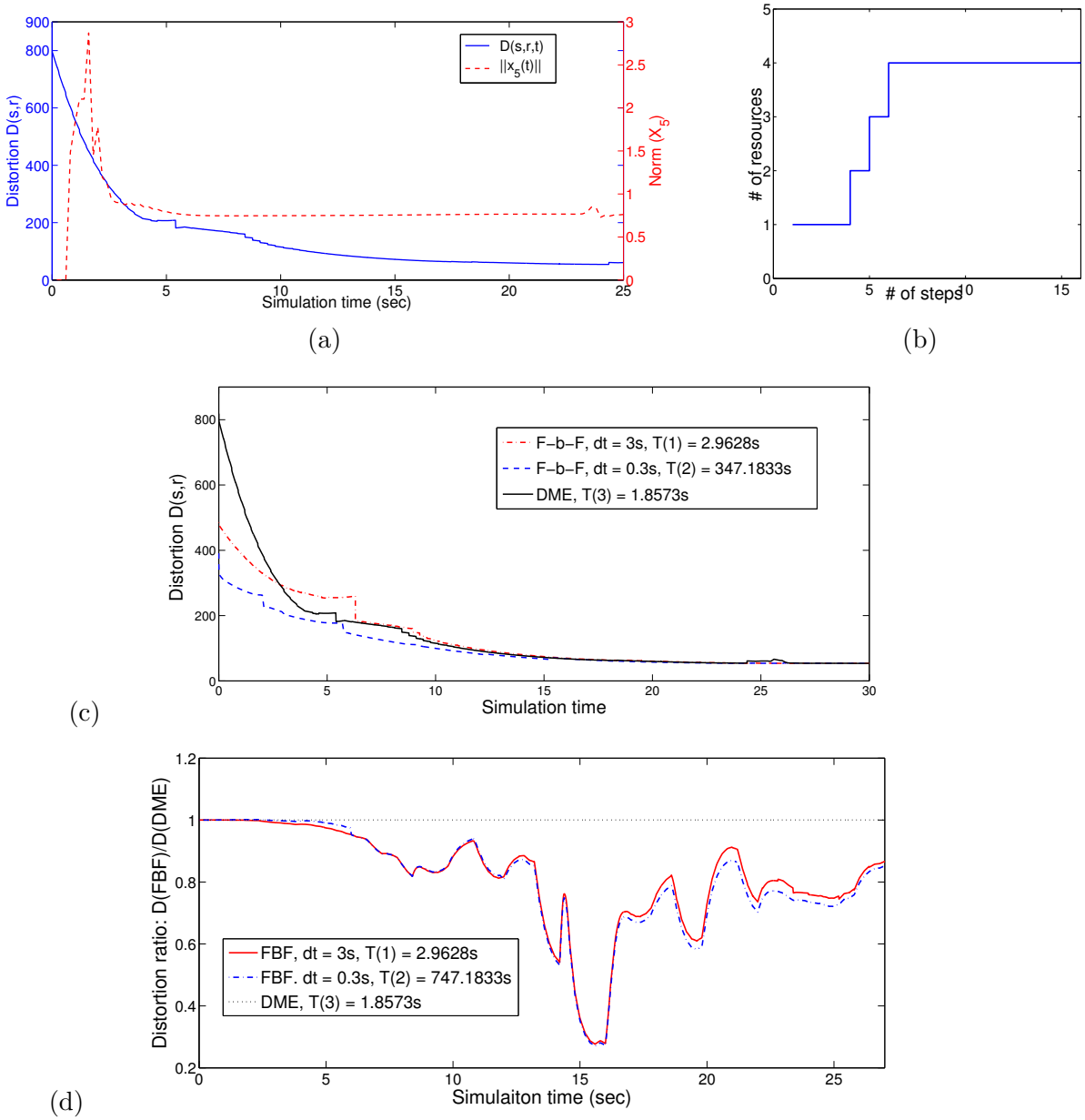


Figure 7.3: Plot (a) depicts the distortion D as time progresses, and it also provides the value changes of the control energy $\|x_5\|$, and its fluctuation indicates the effort used to correct resource dynamics for tracking. Initially, single-cluster placement achieves perfect tracking so no acceleration adjustment is needed. After several resource splits, the resources chase the cluster centers by modifying their accelerations (the peak of $\|x_5\|$). When all cluster centers are attained, the resource accelerations converge to the centroid accelerations, and $\|x_5\|$ tends to a steady-state value. Plot (b) shows the number of distinct resource locations. The comparison between the proposed DME algorithm and the FBF approaches is shown in plot (c) and (d), during which process we assume no occurrence of resource splittings. The distortion D achieved by the DME algorithm is comparable with the FBF method using small time steps, while the computational time of DME is much smaller (1.8573 seconds versus 347.1833 seconds for the latter).

method two times - the first with fewer time instants (10) and the second with many time instants (100), using static clustering in both cases. The DME algorithm, when compared to the FBF approach with 100 time instants, gives comparable distortion (within 88.79%) after a transient time of 25 sec (see Figure 7.3 (a) and (d)) while taking only 5.35% of the computation time. The DME algorithm takes 62.69% computation time when compared to the FBF with fewer time instants (see caption for details).

Note that the comparison of the resulting distortion values in the above simulations is conservative since we have assumed that the results of static clustering at each time sample of the FBF method is applied *instantaneously*. If we account for the drift of the nodes during the computation of the static DA algorithm, the resulting distortions from the FBF method will be more likely to be larger.

7.4 Discussion

7.4.1 Flexibility in implementation

The DME framework presented herein enables cluster splits for better coverage as discussed in Section 7.2. However, it does not explicitly monitor cluster mergers. An easy way to accommodate cluster mergers is to track the pairwise distance for all pairs of resources $d(r_i, r_j)$, and combine two resources if $d(r_i, r_j) < \sigma$ for some threshold σ. Alternatively, we can add a cost term to the distortion function that penalizes nearby resources. An example of one such term is $\sum_{i,j} -\frac{1}{\beta} \log d(r_i, r_j)$ under which two nearby resources will induce a large cost that will increase as the annealing process continues (β increases), making splitting less likely. Inclusion of cluster mergers in our framework is part of our ongoing work.

The DME framework provides a fundamental algorithmic structure for dynamic coverage problems, where the notion of coverage is not limited to geographical objective functions and can easily be extended to include domain specific objectives for other problems. For example, in web-based software engineering, the coverage can be defined in terms of the *quality of service* function [25]; in network routing problems, where decisions on both the locations of communication centers and the routes are required, both distortion and communication costs can be accounted for in the distance

function [34].

7.4.2 Computational complexity

The proposed method adopts the MEP to solve the clustering objective whenever β changes. Therefore the association weights given by (7.5) need to be calculated for all node-resource pairs, which can be computationally expensive. To reduce computational effort, we can approximate the Gibbs distributions by lower-complexity distributions [12]. Since the association between pairs of distant resources and nodes becomes very small as β increases, we can also gradually eliminate the influence of remote nodes and perform progressively distributed computations, leading to a more tractable algorithm. Further approaches to improving scalability of the algorithm are under investigation, and some preliminary results are given in [59].

Summary of This Chapter

Our main achievements in this chapter are summarized as follows.

1. Extension of the DME framework that integrates control design with clustering analysis by considering acceleration-driven dynamics.

2. Proof of the necessity of a dynamic control law in fulfillment of tracking in the presence of acceleration fields.

3. Development of a nonlinear dynamic feedback control law for the resources, under which asymptotic tracking of cluster centers is achieved under mild conditions (*Theorem 8*)

4. Proof of the non-conservativity of our construction method in finding a Lipschitz continuous control that achieves asymptotic tracking (*Theorem 9*).

Chapter 8

Conclusion

We conclude this thesis with a brief summary of our work, and note a few related problems for future research.

8.1 Summary

This work has focused on aggregation of large interconnected systems into subsystems using clustering and aggregation methods, and the development of various implementation algorithms for problems in different settings. The systems under study can be described by directed weighted graphs, in which subsystems are modeled as nodes, and the interrelations between subsystems are modeled as edges. From the clustering perspective, we have formulated the goal of determining a smaller representative graph as a resource allocation problem, where supernodes (nodes of the smaller graph) are viewed as resources and determined accordingly to represent a set of similar original nodes. The edges between supernodes are determined by minimizing the representation error.

The aggregation performance, or the representativeness in using a smaller graph to approximate a large one, has been quantified by a dissimilarity measure introduced in this work. Moreover, the definition of this measure naturally merges the problem of evaluating the representativeness of a particular small graph, and solving a small graph that yields a better approximation. Even with a simplified problem formulation, the problem under study is still inherently combinatorial, and obviously suffers from the computational complexity. We have developed a two state aggregation framework which adopts the MEP-based algorithm in the implementation. Equipped with adequate flexibility, the framework can be adapted to aggregate 1) general directed weighted graphs, 2) large Markov chains, and 3) systems with numerous stochastic processes, and we have presented

the treatment for each scenario. For the Markov chain aggregation case, we have presented a sufficient identifiability condition for nearly completely decomposable chains corresponding to nearly reducible systems, which has directly linked back to the celebrated perturbation theory.

The clustering or resource allocation framework, when applied to tracking and coverage of mobile objects in a low-dimensional space, can be integrated with control design and provide significantly better solutions than the frame-by-frame clustering. In this case, our emphasis has been extended from one snapshot of the dynamic system that needs to be covered by a small number of resources, to a set of consecutive snapshots taken as the evolvement of an acceleration-driven dynamic system. We have posed the objective of sequential resource placement as determining an acceleration field for the resources that always drive the resources towards the actual cluster centers. The design of resource dynamics heavily relies on Lyapunov stability theory, and a Lipschitz continuous and bounded control input has been presented.

8.2 Future Research

Some related future studies include the follow topics.

1. In this thesis, we have noted that one limitation of using the DA algorithm as a central component in solving the optimal aggregation is that centralized calculations are required. Nevertheless, by exploiting an inherent natural localization as the DA algorithm, these calculations can be made increasingly distributed. An initial attempt of a guided decentralized implementation has been discussed in Section 4.1.3.

2. Our work has applied the DA algorithm, which was first proposed in 1992, a complete analysis of its computational complexity has yet to be completed.

3. The focus of this work has been on a clustering and aggregation approach to determine a low-order approximation for large graphs and/or networked systems. Alternative methods are also being actively studied. A rigorous comparison of different methods should be completed.

4. We have discussed a preliminary attempt to use this method to extract the ensemble interactions among a large number of stochastic processes; namely, we have worked on earthquake

models and synthesis data to demonstrate the effectiveness of this direction. Further analytical conclusions could be drawn with additional application domains.

Appendix A

Background for Control Systems

We briefly present some definitions and results from system analysis and control theory to be used in Part II, most of them can be found in [36].

We consider the coverage problems in $\mathbb{R}^n$; for the ease of notation, and the consistency with most standard control system literature, we overwrite some notations in this part. We use regular face small letters (e.g., x) and capital letters (e.g., A) for vectors and matrices respectively.

A.1 Lyapunov stability theory

Consider a nonlinear time-invariant system $\dot{x} = f(x)$, where $f : \mathbb{R}^n \to \mathbb{R}^n$, a point $x_e \in \mathbb{R}^n$ is said to be an *equilibrium point* of the system if $f(x_e) = 0$. The equilibrium point x_e is

- *stable*, if for each $\epsilon > 0$, there exists $\delta = \delta(\epsilon) > 0$, such that for every trajectory $x(t)$ with $\|x(0) - x_e\| < \delta$, $\|x(t) - x_e\| \leq \epsilon$ for all $t \geq 0$.
- *asymptotically stable (AS)*, if it is stable and the δ can be chosen such that $\|x(0) - x_e\| < \delta \Rightarrow x(t) \to x_e$ as $t \to \infty$.
- *globally asymptotically stable (GAS)*, if the $\delta = \infty$, or equivalently, for every trajectory $x(t)$, we have $x(t) \to x_e$ as $t \to \infty$.

The Lyapunov theory provides a way to conclude system stability without solving the trajectories of a system.

A function $V : \mathbb{R}^n \to \mathbb{R}$ is *positive definite* if

$$V(0) = 0 \text{ and } V(x) > 0 \text{ for all } x \neq 0.$$

If it satisfies the weaker condition $V(x) \geq 0$ for all $x \neq 0$, then it is said to be *positive semidefinite*.

Lyapunov Stability Theorem

Let $x_e = 0$ be an equilibrium point for system $\dot{x} = f(x)$ and $\mathcal{D} \subset \mathbb{R}^n$ be a domain containing x_e. Suppose there is a continuously differentiable function $V : \mathcal{D} \to \mathbb{R}$ such that

(i) V is positive definite,

(ii) $\dot{V}(x) < 0$ for all $x \neq 0$, and $\dot{V}(0) = 0$,

(iii) $V(x) \to \infty$ as $\|x\| \to \infty$,

then the system is GAS, or every trajectory converges to zero as $t \to \infty$.

These Lyaponov theorems provide sufficient conditions for system stability. Sometimes, only weak version of condition (i) or (ii) is satisfied (say, holds with semidefinite), then the following LaSalle's theorem provides a test of stability under extended conditions.

LaSalle's Invariant Theorem

A set $\mathcal{M} \subset \mathbb{R}^n$ is said to be an *positive invariant set* with respect to system $\dot{x} = f(x)$ if every trajectory starting at a point in $\mathcal{M}$ will stay in M for the entire time in the future, that is, $x(0) \in \mathcal{M} \Rightarrow x(t) \in \mathcal{M}, \forall t \geq 0$.

Let $\Omega \subset \mathcal{D} \subset \mathbb{R}^n$ be a compact positively invariant set with respect to the system $\dot{x} = f(x)$. Let $V : \mathcal{D} \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leq 0$ in Ω. Let $\mathcal{V} \subset \Omega$ be the set of all points in Ω where $\dot{V}(x) = 0$. Let $\mathcal{M} \subset \mathcal{V}$ be the largest invariant set in $\mathcal{V}$. Then every trajectory $x(t)$ with $x(0) \in \Omega$ approaches $\mathcal{M}$ as $t \to \infty$, that is $\lim_{t\to\infty} \text{dist}(x(t), \mathcal{M}) = \lim_{t\to\infty} \inf_{z\in\mathcal{M}} \|x(t) - z\| = 0$.

The LaSalle's theorem extends the Lyapunov's theorem by allowing the system to have an *equilibrium set*, rather than a single equilibrium point; also, the function $V(x)$ does not have to be positive definite (and $\dot{V}(x)$ does not have to be negative definite).

A.2 Control system stabilization

We state the Artstein's theorem [3] for systems that are linear in control, which guarantees the existence of a feedback stabilizing control law; then provide an explicit stabilizing control presented in [63].

Sontag's Constructive Proof of Artstein's Theorem

Let a control system have the form

$$\dot{x}(t) = f(x) + g(x)u, \tag{A.1}$$

in which $f : \mathbb{R}^n \to \mathbb{R}^n$ is continuous and $g : \mathbb{R}^n \to \mathcal{U} \subset \mathbb{R}^m$ defines linear functionals from $\mathcal{U}$ to $\mathbb{R}^n$. There exists a closed loop control $u(x) : \mathbb{R}^n \to \mathcal{U}$, continuous (except possibly at $x = 0$), and makes system (A.1) asymptotically stable if and only if a neighborhood $\mathcal{W}$ of 0 exists and a continuously differentiable control Lyapunov function $V : \mathcal{W} \to \mathbb{R}$ satisfies (i) positive definite and (ii) $\dot{V}(x) < 0$ for at least one $u \in \mathcal{U}$ for all $x \neq 0$.

One such control u that guarantees asymptotical stability can be constructed as follows. Write the time derivative of V as, $\dot{V}(x) = \nabla_x V(x)^T [f(x) + g(x)u]$ as

$$\dot{V}(x) = \underbrace{\nabla_x V(x)^T f(x)}_{a(x)} + \underbrace{\nabla_x V(x)^T g(x)}_{b^T(x)} u. \tag{A.2}$$

Then a feedback control that stabilizes the system is given by

$$u = -\left[\frac{a(x) + \sqrt{a^2(x) + (b^T(x)b(x))^2}}{b^T(x)b(x)}\right] b(x), \tag{A.3}$$

which is easy to verify. For more details see [63].

Appendix B

Mathematical Results

We present and prove some mathematical results that are used in this thesis.

Lemma 10 *Let $\boldsymbol{\mu}$ and $\mathbf{x}$ be probability distribution over a finite space of dimension n, and $\boldsymbol{\mu}$ is given. The largest l_1 distance between $\mathbf{x}$ and $\boldsymbol{\mu}$ is given by $2(1-\min_i \mu_i)$. In particular, when $\boldsymbol{\mu}$ is the uniform distribution, $\max\|\mathbf{x}-\boldsymbol{\mu}\|_1 = 2(1-\frac{1}{n})$.*

Proof: Without loss of generality, assume $\mathbf{x}, \boldsymbol{\mu} \in \mathbb{R}^n_+$. Their maximum l_1 distance is the solution to the following optimization problem.

$$\max \sum_{i=1}^{n} |x_i - \mu_i| \quad \text{s.t.} \sum_{i=1}^{n} x_1 = 1, \;\; x_i \geq 0, \forall i.$$

Since the objective function and the feasible set are both convex, the maximum is attained at the boundary of the feasible set, that is, either at one vertex (unique maximum) or the entire edge (with equal function value). By symmetry, it is sufficient to consider one edge connecting two vertices $\mathbf{x}_1$ and $\mathbf{x}_2$ with coordinates $\mathbf{e}_1$ and $\mathbf{e}_1$.

The function values at these vertices are $f(\mathbf{x}_1) = 2(1-\mu_1)$ and $f(\mathbf{x}_2) = 2(1-\mu_2)$. Any point on the edge can be written as $\mathbf{x}_t = t\mathbf{x}_1 + (1-t)\mathbf{x}_2$ for $0 \leq t \leq 1$, and $f(\mathbf{x}_t) = \|[t, 1-t, 0, \ldots, 0] - \boldsymbol{\mu}\|_1 = |t-\mu_1| + |1-\mu_2-t| + 1 - \mu_1 - \mu_2$. It is easy to get

$$\max_{0\leq t\leq 1} f(\mathbf{x}_t) \begin{cases} 2(1-\mu_2) & 0 \leq t \leq \mu_1 \\ 2[1-(\mu_1+\mu_2)] & \mu_1 \leq t \leq 1-\mu_2 \\ 2(t-\mu_1) & 1-\mu_2 \leq t \leq 1 \end{cases}$$

Take all t into account, we conclude that $\max f(\mathbf{x}) = 2(1-\min_{1\leq i\leq n}\mu_i)$. In particular, when $\boldsymbol{\mu}$ is the uniform distribution $\mu_i = \frac{1}{n}$ for all i, the claim holds too. □

References

[1] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

[2] Albert Ando and Franklin M Fisher. Near-decomposability, partition and aggregation, and the relevance of stability discussions. *International Economic Review*, 4(1):53–67, 1963.

[3] Zvi Artstein. Stabilization with relaxed controls. *Nonlinear Analysis: Theory, Methods and Applications*, 7(11):1163 – 1173, 1983.

[4] S. Asmussen. *Applied Probability and Queues*. Applications of mathematics. Springer, 2003.

[5] C.L. Beck, S. Lall, T. Liang, and M. West. Model reduction, optimal prediction, and the Mori-Zwanzig representation of Markov chains. In *Proc. of the 48th IEEE Conf. on Decision and Control*, pages 3282 –3287, 2009.

[6] Vincent D Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM review*, 46(4):647–666, 2004.

[7] E.N. Brown, R.E. Kass, and P.P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7(5):456–461, 2004.

[8] Gary Chartrand, Grzegorz Kubicki, and Michelle Schultz. Graph similarity and distance in graphs. *aequationes mathematicae*, 55:129–145, 1998.

[9] Derek G Corneil and Calvin C Gotlieb. An efficient algorithm for graph isomorphism. *Journal of the ACM*, 17(1):51–64, 1970.

[10] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Trans. on Robotics and Automation*, 20(2):243–255, 2004.

[11] Pierre-Jacques Courtois. Error analysis in nearly-completely decomposable stochastic systems. *Econometrica: Journal of the Econometric Society*, pages 691–709, 1975.

[12] K. Demirciler and A. Ortega. Reduced-complexity deterministic annealing for vector quantizer design. *EURASIP Journal on Advances Signal Processing*, 2005:1807–1820, 2005.

[13] Kun Deng, P.G. Mehta, and S.P. Meyn. Optimal Kullback-Leibler aggregation via spectral theory of Markov chains. *IEEE Trans. on Automatic Control*, 56(12):2793–2808, Dec. 2011.

[14] I.S. Dhillon, Yuqiang Guan, and B. Kulis. Weighted graph cuts without eigenvectors: a multilevel approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, Nov. 2007.

[15] D.V. Dimarogonas, E. Frazzoli, and K.H. Johansson. Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, 57(5):1291–1297, 2012.

[16] Annette J. Dobson. *An Introduction to Generalized Linear Models*. London: CRC Press, 2002.

[17] Z. Drezner. *Facility Location: A Survey of Applications and Methods*. Springer Verlag, New York, 1995.

[18] Robert F. Engle and Asger Lunde. Trades and quotes: A bivariate point process. *Journal of Financial Econometrics*, 1(2):159–188, 2003.

[19] B.S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. A Hodder Arnold Publication. Wiley, 2001.

[20] M.-L. Fernández and G. Valiente. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters*, 22(6):753–758, 2001.

[21] Robert J Fletcher, Miguel A Acevedo, Brian E Reichert, Kyle E Pias, and Wiley M Kitchens. Social network models predict movement and connectivity in ecological landscapes. *Proceedings of the National Academy of Sciences*, 108(48):19282–19287, 2011.

[22] E. Frazzoli and F. Bullo. Decentralized algorithms for vehicle routing in a stochastic time-varying environment. In *Proc. of the 43rd IEEE Conf. on Decision and Control*, volume 4, pages 3357 – 3363, Dec. 2004.

[23] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:721–741, Nov. 1984.

[24] A. Gersho and R. Gray. *Vector Quantization and Signal Compression*. Kluwer, Boston, Massachusetts, 1st edition, 1991.

[25] Hamoun Ghanbari, Cornel Barna, Marin Litoiu, Murray Woodside, Tao Zheng, Johnny Wong, and Gabriel Iszlai. Tracking adaptive performance models using dynamic clustering of user classes. In *Proc. of the 2nd Joint WOSP/SIPEW International Conf. on Performance Engineering*, ICPE '11, pages 179–188, New York, NY, USA, 2011. ACM.

[26] CWJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[27] Peter Guttorp and Mary Lou Thompson. Nonparametric estimation of intensities for sampled counting processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52(1):h157–173, 1990.

[28] A. G. Hawkes and L. Adamopoulos. Cluster models for earthquakes - regional comparisons. *Bull. Int. Statist. Inst.*, 45:454 – 461, 1973.

[29] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[30] I. I. Hussein and D. M. Stipanovic. Effective coverage control for mobile sensor networks with guaranteed collision avoidance. *IEEE Trans. on Control Systems Technology*, 15(4):642–657, 2007.

[31] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[32] E. T. Jaynes. *Probability Theory - The Logic of Science*. Cambridge University Press, 2003.

[33] Edwin T Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.

[34] Nachiket V. Kale and Srinivasa M. Salapaka. Maximum entropy principle-based algorithm for simultaneous resource location and multihop routing in multiagent networks. *IEEE Trans. on Mobile Computing*, 11(4):591–602, Apr. 2012.

[35] Yakov Keselman, Ali Shokoufandeh, M Fatih Demirci, and Sven Dickinson. Many-to-many graph matching via metric embedding. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–850. IEEE, 2003.

[36] H.K. Khalil. *Nonlinear Systems*. Prentice Hall PTR, 2002.

[37] Sanggyun Kim, David Putrino, Soumya Ghosh, and Emery N. Brown. A Granger causality measure for point process models of ensemble neural spiking activity. *PLoS Comput Biol*, 7(3):e1001110, Mar 2011.

[38] C. Kittel and H. Kroemer. *Thermal Physics*. W. H. Freeman, 1980.

[39] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[40] G. Kotsalis, A. Megretski, and M.A. Dahleh. Balanced truncation for a class of stochastic jump linear systems and model reduction for hidden Markov models. *IEEE Trans. on Automatic Control*, 53(11):2543 –2557, Dec. 2008.

[41] A. Kwok and S. Martinez. A distributed deterministic annealing algorithm for limited-range sensor coverage. *IEEE Trans. on Control Systems Technology*, 19(4):792–804, Jul. 2011.

[42] Chung Laung Liu. *Introduction to combinatorial mathematics*. Computer science series. McGraw-Hill, 1968.

[43] S.P. Lloyd. Least squares quantization in PCM. *IEEE Trans. on Information Theory*, 28(2):129–137, 1982.

[44] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is NP-hard. In *Proc. of the 3rd International Workshop on Algorithms and Computation*, pages 274–285, Berlin, Heidelberg, 2009. Springer-Verlag.

[45] Magnus Egerstedt Mehran Mesbahi. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[46] Marina Meilă. Comparing clusteringsan information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[47] Bruno T. Messmer and Horst Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(5):493–504, 1998.

[48] Yosihiko Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83(401):pp. 9–27, 1988.

[49] Yosihiko Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50:379–402, 1998.

[50] Murat Okatan, Matthew A. Wilson, and Emery N. Brown. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural Comput.*, 17:1927–1961, Sep. 2005.

[51] R. Phillips and P. Kokotovic. A singular perturbation approach to modeling and control of Markov chains. *IEEE Trans. on Automatic Control*, 26(5):1087–1094, Oct. 1981.

[52] C. Quinn, T. P. Coleman, N. Kiyavash, and N. Hatsopoulos. Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *J.Computational Neuroscience*, Jan 2011.

[53] C.J. Quinn, N. Kiyavash, and T.P. Coleman. Equivalence between minimal generative model graphs and directed information graphs. In *IEEE International Symposium on Information Theory Proceedings (ISIT),*, 2011.

[54] Z. Rached, F. Alajaji, and L.L. Campbell. The Kullback-Leibler divergence rate between Markov sources. *IEEE Trans. on Information Theory*, 50(5):917–921, May 2004.

[55] K. Rose. Deterministic annealing for clustering, compression, classification, regression and related optimization problems. *Proc. of the IEEE*, 86(11):2210–2239, Nov. 1998.

[56] Srinivasa M. Salapaka, A. Khalak, and M.A. Dahleh. Constraints on locational optimization problems. In *Proc. of the 42nd IEEE Conf. on Decision and Control*, volume 2, pages 1741 – 1746, Dec. 2003.

[57] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[58] Rodolphe Sepulchre, Mrdjan Jankovic, and Petar Kokotovic. *Constructive Nonlinear Control*. Springer-Verlag, 1997.

[59] P. Sharma, Srinivasa M. Salapaka, and Carolyn L. Beck. A scalable approach to combinatorial library design for drug discovery. *Journal of Chemical Information and Modeling*, 48(1):27–41, 2008.

[60] P. Sharma, Srinivasa M. Salapaka, and Carolyn L. Beck. Entropy-based framework for dynamic coverage and clustering problems. *IEEE Trans. on Automatic Control*, 57(1):135–150, Jan. 2012.

[61] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug. 2000.

[62] Herbert A Simon and Albert Ando. Aggregation of variables in dynamic systems. *Econometrica: journal of the Econometric Society*, pages 111–138, 1961.

[63] E. D. Sontag. A 'universal' construction of Artstein's theorem on nonlinear stabilization. *System and Control Letters*, 13(2):117–123, 1989.

[64] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[65] M. Vidyasagar. A metric between probability distributions on finite sets of different cardinalities and applications to order reduction. *IEEE Trans. on Automatic Control*, 57(10):2464–2477, Oct. 2012.

[66] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[67] Xiang Wang and Ian Davidson. Flexible constrained spectral clustering. In *Proc. of the 16th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, KDD '10, pages 563–572, New York, NY, USA, 2010. ACM.

[68] J.H. Wilkinson. *The Algebraic Eigenvalue Problem*. Monographs on numerical analysis. Clarendon Press, 1988.

[69] Haiyong Xu and Frederic P. Schoenberg. Point process modeling of wildfire hazard in los angeles county, california. *Annals of Applied Statistics*, 5(2A):684–704, 2011.

[70] Rui Xu, Donald Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

[71] Yunwen Xu, Srinivasa M Salapaka, and Carolyn L Beck. On reduction of graphs and Markov chain models. In *50th IEEE Conference on Decision and Control and European Control Conference, 2011*, pages 2317–2322. IEEE, 2011.

[72] Yunwen Xu, Srinivasa M Salapaka, and Carolyn L Beck. Aggregation of Graph Models and Markov Chains by Deterministic Annealing. *IEEE Transactions on Automatic Control*, PP(99):1–1, 2014.

[73] Luh Yen, Francois Fouss, Christine Decaestecker, Pascal Francq, and Marco Saerens. Graph nodes clustering with the sigmoid commute-time kernel: A comparative study. *Data Knowledge Engineering*, 68(3):338–361, Mar. 2009.

[74] Zhenjie Zhang, Yin Yang, A.K.H. Tung, and D. Papadias. Continuous k-means monitoring over moving objects. *IEEE Transactions on Knowledge and Data Engineering*, 20(9):1205 –1216, Sept 2008.