ABSTRACT

WESSELL, CHARLES DAVID. Stochastic Data Clustering. (Under the direction of Carl D. Meyer.)

Data clustering, the search for hidden structure in data sets, is a field with many different methodologies, all of which work well in some situations and poorly in others. Because of this, there is growing interest in finding a consensus clustering solution that combines the results from a large number of clusterings of a particular data set. These large number of solutions can be stored in a square matrix that is often nearly uncoupled, and through clever use of theory regarding dynamical systems first published in 1961 by Herbert Simon and Albert Ando, a clustering method can be developed.

This thesis will explain the rationale behind this new clustering method and then make sure it has a solid mathematical foundation. One of the key steps in this new method is converting a nearly uncoupled matrix to doubly stochastic form. Among the contributions of this thesis is a measure of near uncoupledness that can be applied to matrices both before and after that conversion and rigorous proofs that the conversion to doubly stochastic form does not destroy the symmetry, irreducibility, or near uncoupledness of the original matrix.

Additionally, the connection between the second eigenvalue of an irreducible, symmetric, doubly stochastic matrix and the nearly uncoupled structure of that matrix will be rigorously proven, with the result being that examination of the second eigenvalue will play an essential role in the new clustering algorithm.

Actual clustering results will be presented to show that the intuitive notions and mathematical theory that constructed this method do indeed produce high quality clustering results.

Stochastic Data Clustering

by
Charles David Wessell

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Mathematics

Raleigh, North Carolina

2011

APPROVED BY:

David A. Dickey	Ilse C. F. Ipsen
Ernest L. Stitzinger	Carl D. Meyer Chair of Advisory Committee

DEDICATION

To Kim and Rachel.

BIOGRAPHY

Charles David Wessell was born in Glen Cove, New York on August 27, 1962. He attended elementary school in Cairo and Greenfield, New York before moving to North Carolina in 1974 and graduating from Hillsborough's Orange High School in 1980.

After beginning his college education at the Georgia Institute of Technology, he transferred to North Carolina State University and received his bachelor's degree in Math Education in 1985, followed by a master's degree in Applied Math in 1989. In 1988, Chuck was captain of North Carolina State's national champion College Bowl team. In recognition of this, he was awarded the Order of the Long Leaf Pine, North Carolina's highest civilian honor.

In the labyrinth of the work world Chuck has been a math tutor, singing telegram delivery man, IBM software support person, flower delivery man, high school teacher, bookseller, shopkeeper, and community college math instructor.

Chuck's athletic feats include hitting a half-court shot at halftime of a college basketball game, hiking all 2,175 miles of the Appalachian Trail over many trips between 1997 and 2007, completing 10 marathons, and run/eat/running five Krispy Kreme Challenges.

Chuck is married to partial differential equations researcher Kimberly Renee Spayd. Their daughter, Rachel, likes to play in sand, has very strong arms, and can say the word *dissertation*.

ACKNOWLEDGEMENTS

Thank you to the teachers whose influence I still feel today, particularly Mrs. Bartman, Mrs. Cooke, Mrs. Severson, Mrs. Richon, Mrs. Eidener, Dr. Penick, and Dr. Barreau. Thank you to Lee Ann Spahr, the best boss I have had, who understood why I needed to go back to school. Thanks to Tim Chartier for the advice, assistance, and stories. Thanks to all the math department staff for their help over the past four years, especially Denise Seabrooks, the de facto director of the mathematics graduate program.

Rachel, Kim, and I have been tremendously fortunate to have Kim's mother Shirley Spayd care for Rachel on school days. Rachel has grown to be an active, inquisitive, happy, determined girl and Shirley is a huge part of the reason why.

The first person in the math department I spoke to when I decided to return to school was my master's advisor Ernie Stitzinger, who was very encouraging then and very helpful since. Ilse Ipsen has made this thesis much better with her probing questions, her eye for detail, and my fear that she will find some huge error. Many thanks to David Dickey for finding room in his busy schedule to serve on my committee. He found several minor errors and made a suggestion for future research that is already proving fruitful.

Carl Meyer has made me a better mathematician, speaker, writer, and thinker. He is a tremendous source of matrix theory knowledge, not only of the definitions and theorems, but also of the people who created them. He has given advice and guidance, but has always encouraged me to explore, to make mistakes, and to learn. It is a mark of my growth as a mathematician that over the past two-and-a-half years I have gone from understanding ten percent of what he says to ninety percent. Becky Meyer has become a friend and source of chocolate. She is certainly the great woman behind the man.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

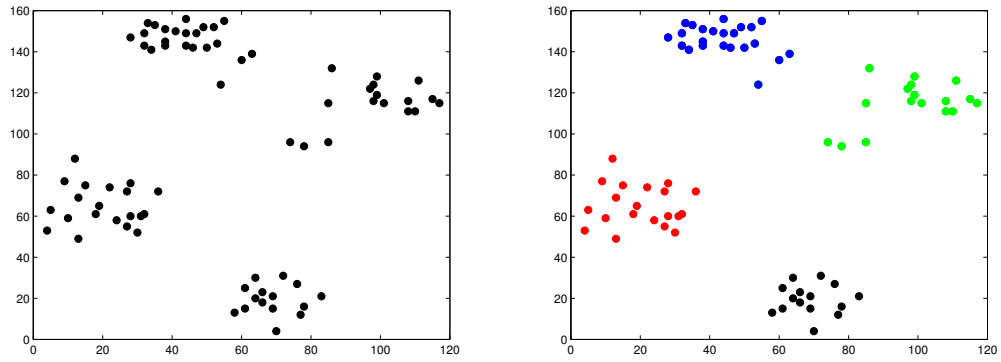
CHAPTER 1

Overview

1.1 Cluster Analysis

Cluster analysis, or data clustering, is the search for hidden structure in data sets. Such a search is not difficult if the data set contains a small number of elements each of which is of low dimension. For example, consider the graph in Figure 1.1. These 75 two-dimensional points are collectively known as the Ruspini data set and are often used to illustrate data clustering techniques [69]. Most people have no difficulty grouping the points into four clusters.

The search for patterns becomes much more difficult if there are hundreds, thousands, or even millions of data points. It is also more difficult if the dimensionality of the data is greater than three, since now the human eye cannot be used to cluster. It then becomes necessary to create algorithms to examine and cluster the data. In cluster analysis of



(a) The Ruspini data set

(b) A natural clustering of this data set.

Figure 1.1: The Ruspini data set can be easily separated into four clusters by the human eye.

real-world data there is rarely going to be a "right" answer. Instead the goal of those who create and use clustering algorithms is to gain some new insight into a data set. A marketing executive who discovers some well-hidden similarity between the online shopping habits of young, urban professionals and small-town housewives may hold the key to increasing his company's profits. A national security employee who finds some obscure connection between the vocabulary used during the phone calls of a domestic organized crime group and the emails amongst members of a European terrorist cell may thwart a potential disaster, while a campaign manager with knowledge of the clustering of precincts in a candidate's district may hold the key to winning an election. In each of these cases having just some knowledge of a data set's structure may be enough to make a difference.

The first use of the term cluster analysis is in a 1939 monograph by the psychologist Robert Tryon [86]. It was not until 1954 that the term appeared in the title of a referreed journal article, this time used by an anthropologist [15]. Over the last 70 years many

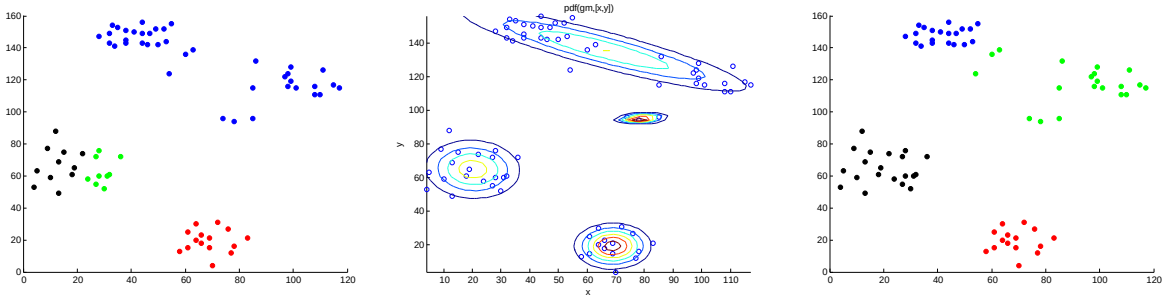
algorithms have been developed to cluster data sets and during that time a number of books (for example [3, 37, 79, 43, 22, 77]) and academic review articles (including [21, 44, 45, 7, 42]) have been written surveying the field. A 2007 monograph on clustering methods lists 58 different algorithms in its appendix [29].

Having access to a large number of data clustering algorithms is not necessarily a good thing for a researcher since it may be unclear which algorithm will work best with any particular data set. The fact that many algorithms provide one or more input parameters that can be set to a variety of values only adds to the confusion. And finally, there are clustering algorithms such as k-means, nonnegative matrix factorization (NMF), and mixture models that use random initializations, which can lead to different final results even on relatively simple data sets.[1] Figure 1.2 shows three less-than-spectacular clusterings of the Ruspini data set.

Knowing there are many clustering algorithms to choose from and that at least some of them do not give consistent results, may lead one to wonder if there is a single method better than all the rest. In the preface to his *Introduction to Clustering Large and High-Dimensional Data*, Jacob Kogan comments on this possibility by adopting the following "theorem" from control theorist George Leitmann (the emphasis on the words "best" and "superior" is Leitmann's):

> **Theorem.** *There does not exist a* best *method, that is, one which is* superior *to all other methods, for solving all problems in a given class of problems. [50, 57]*
>
> *Proof.* By contradiction. □

[1] NMF and k-means will be used almost exclusively in this thesis. Two good references by the developers of NMF are [54] which features applications, and [55] which focuses on implementation and convergence proofs. Since k-means is so popular, there are many elementary explanations of it in print and on the web. For a more mathematically rigorous explanation, try Chapters 2 and 4 of [50]. A treatment that also considers the programming aspect of k-means can be found in Chapter 9 of [29].



(a) k-means (b) Mixture of Gaussians (c) Complete link HC

Figure 1.2: Examples of bad clustering for the Ruspini data set. In graph (1.2a) k-means splits one of the natural groups while clustering two separate groups together. In graph (1.2b) a model looking for multiple Gaussian distributions forms one compact cluster at the expense of creating another elliptical one with high eccentricity, while choosing the complete link parameter for MATLAB's hierarchical clustering command leads to the misclustering in graph (1.2c).

So rather than continue on some quixotic quest for the perfect clustering method, it may be more productive to see if these varied solutions can be combined in some way to arrive at a single, robust clustering of the original data set. This idea is the topic of the next section.

1.2 Consensus Clustering

The term for combining multiple clustering results into a single clustering will be called *consensus clustering* throughout this thesis, but that phrase is not the only one used to describe such an endeavor. Since a collection of clustering methods can be referred to as an ensemble, this process is sometimes called *ensemble clustering* [38] while other authors use the term *cluster aggregation* [30]. The rest of this section will introduce the vocabulary and notation associated with consensus clustering along with an overview of the approaches others have brought to this problem.

The starting point for any clustering method is an m-dimensional data set of n elements. The data set can thus be stored as an $m \times n$ matrix $\mathbf{A}$ where each column represents an element of the data set and each row contains the value of a particular attribute for each of the elements. If the assignment of clusters from a single run of a clustering algorithm is denoted by $\mathcal{C}_i$, then the input to any consensus method is $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$.

One approach for solving this problem is attempting to find a clustering $\mathcal{C}^*$ that is as close as possible to all the $\mathcal{C}_i$'s. This is an optimization problem known as *median partition*, and is known to be NP-complete. A number of heuristics for the median partition problem exist. Discussion of these heuristics with comparisons and results on real world data sets can be found in [24, 25, 31].

Other researchers have brought statistical techniques to bear on this problem, using bootstrapping or other more general resampling techniques to cluster subsets of the original data set, and then examining the results using some measure of consistency to settle on the final clustering [27, 65].

Additional approaches include a consensus framework built on a mixture of Gaussians model [34] and using algorithms originally intended for rank aggregation problems [2].

Other approaches to this problem begin by storing the information from each $\mathcal{C}_i$ in an $n \times n$ adjacency matrix $\mathbf{A}^{(i)}$ whose elements are defined by

$$a_{jk}^{(i)} = \begin{cases} 1 & : \text{when } \mathcal{C}_i \text{ places elements } j \text{ and } k \text{ in the same cluster} \\ 0 & : \text{when } \mathcal{C}_i \text{ does not place elements } j \text{ and } k \text{ in the same cluster.} \end{cases}$$

Definition 1.1. *If* $\{\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}, \ldots, \mathbf{A}^{(r)}\}$ *is a collection of adjacency matrices created from clusterings of the same data set, then the sum of these matrices*

$$\mathbf{S} = \mathbf{A}^{(1)} + \mathbf{A}^{(2)} + \mathbf{A}^{(3)} + \cdots + \mathbf{A}^{(r)} \tag{1.1}$$

is called the consensus similarity matrix *or the* consensus matrix. *Throughout this thesis the symbol* **S** *will be reserved for this type of matrix.*

It should be noted, that some cluster researchers prefer to define **S** as the sum of the adjacency matrices multiplied by $\frac{1}{r}$, so that the resulting matrix entry s_{ij} equals the fraction of the time elements i and j cluster together and $s_{ij} \in [0, 1]$. In this thesis we will always use the Definition 1.1.

Once the consensus similarity matrix is created, one can then cluster the original data by clustering the columns of **S** using a method of the researcher's choosing. This method has been shown to create meaningful clusters using a variety of methods both to create the original clusterings and to cluster the columns of the consensus similarity matrix [26, 67], though typically elements of **S** below a certain threshold are replaced by zero.

The collection of these r adjacency matrices can be used to define a hypergraph which can then be partitioned (i.e. clustered) using known hypergraph partitioning algorithms [83, 82] (see Figure 1.4).

A new methodology developed to cluster different conformations of a single drug molecule comes the closest to the approach developed in this thesis. For this application, a Markov chain transition matrix can be created where the ij-th entry gives the probability the molecule changes from conformation i to conformation j. The goal is to then find sets of conformations such that if the molecule is currently in a particular set, it will remain in that set for a relatively long time. Approaches to this clustering problem have included examination of the first few eigenvectors of the transition matrix ([18] and then improved in [19]), clustering the data based on the second singular vector [28, 85], and spectral analysis of a family of Hermitian matrices that is a function of the transition matrix [41].

$$\mathcal{C}_1 = \mathcal{C}_4 = \mathcal{C}_5 = \{D, U\}; \{E, N, W\}$$

$$\mathbf{A}_1 = \mathbf{A}_4 = \mathbf{A}_5 = \begin{array}{c} \\ \mathrm{D} \\ \mathrm{E} \\ \mathrm{N} \\ \mathrm{U} \\ \mathrm{W} \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{D} & \mathrm{E} & \mathrm{N} & \mathrm{U} & \mathrm{W} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

(a) The clusterings $\mathcal{C}_1$, $\mathcal{C}_4$, and $\mathcal{C}_5$ are identical. A given clustering occurring multiple times is common in consensus clustering. For this example, this adjacency matrix will be used three times in constructing the consensus matrix.

$$\mathcal{C}_2 = \{D, W\}; \{E, N, U\}$$

$$\mathbf{A}_2 = \begin{array}{c} \\ \mathrm{D} \\ \mathrm{E} \\ \mathrm{N} \\ \mathrm{U} \\ \mathrm{W} \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{D} & \mathrm{E} & \mathrm{N} & \mathrm{U} & \mathrm{W} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

(b) The adjacency matrix for $\mathcal{C}_2$

$$\mathcal{C}_3 = \{D, U, W\}; \{E, N\}$$

$$\mathbf{A}_3 = \begin{array}{c} \\ \mathrm{D} \\ \mathrm{E} \\ \mathrm{N} \\ \mathrm{U} \\ \mathrm{W} \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{D} & \mathrm{E} & \mathrm{N} & \mathrm{U} & \mathrm{W} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

(c) The adjacency matrix for $\mathcal{C}_3$

$$\mathbf{S} = \begin{array}{c} \\ \mathrm{D} \\ \mathrm{E} \\ \mathrm{N} \\ \mathrm{U} \\ \mathrm{W} \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{D} & \mathrm{E} & \mathrm{N} & \mathrm{U} & \mathrm{W} \end{array} \\ \begin{pmatrix} 5 & 0 & 0 & 4 & 2 \\ 0 & 5 & 5 & 1 & 3 \\ 0 & 5 & 5 & 1 & 3 \\ 4 & 1 & 1 & 5 & 1 \\ 2 & 3 & 3 & 1 & 5 \end{pmatrix} \end{array}$$

(d) $\mathbf{S} = \mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3 + \mathbf{A}_4 + \mathbf{A}_5$

Figure 1.3: In this small example, the set $\{D, E, N, U, W\}$ has been clustered five times. The first, fourth, and fifth clusterings are identical. The consensus matrix $\mathbf{S}$ is the sum of all five adjacency matrices.

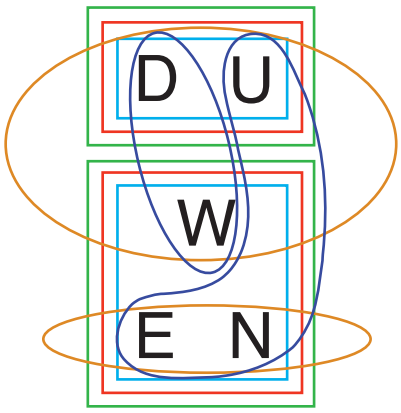


Figure 1.4: A hypergraph of the set and clusterings from Figure 1.3. In a hypergraph, an edge surrounds all the vertices it connects, so in a clustering context elements in the same cluster would be enclosed by the same edge. In our example the identical clusterings of $\mathcal{C}_1$, $\mathcal{C}_4$, and $\mathcal{C}_5$ are represented by the blue, red and green rectangles, respectively. The purple ellipse and boomerang shape represent $\mathcal{C}_2$, and the two orange ellipses represent $\mathcal{C}_4$. To cluster using this approach, one wishes to cut the smallest number of edges that will leave k distinct groups. For example, if both orange and both purple edges are cut, the $k = 2$ clustering $\{D, U\}, \{E, N, W\}$ remains. Those four cuts are the smallest number that can be made to arrive at two clusters.

1.3 A New Approach

If the rows and columns of $\mathbf{S}$ are interchanged in such a way that elements in the same cluster are adjacent to one another, the structure of $\mathbf{S}$ is often that of a nearly block diagonal matrix. In the next chapter we will develop the ideas behind Simon-Ando theory and see that the theory is designed to take advantage of this structure and hence is a candidate to be applied to the consensus clustering problem. During that development it will become apparent that the clustering algorithm will be greatly simplified if $\mathbf{S}$ can be converted to doubly stochastic form, so in Chapter 3 we will review the algorithm we will use to convert the similarity matrix to doubly stochastic form and show that such a conversion does not destroy characteristics like symmetry or near uncoupledness. Chapter 4 will address additional theoretical concerns before Chapter 5 formally defines

the stochastic clustering algorithm and demonstrates it with a small example. We examine some algorithm implementation issues in Chapter 6, present some results in Chapter 7, and share some final thoughts in Chapter 8.

1.4 Notation

Throughout this thesis, bold-face capital letters denote matrices. The symbol m_{ij} denotes the value at row i, column j of $\mathbf{M}$. If $\mathbf{M}$ has a block structure, then $\mathbf{M}_{ij}$ represents the sub-matrix located in the ith row and jth column of blocks. Bold-face small letters denote column vectors. Below is a list of certain letters reserved for a specific role.

- A general data set with n elements each described by m numerical attributes stored in matrix form: $\mathbf{A}_{m\times n}$
- Measures of similarity between the n elements of a data set will be stored in an $n \times n$ symmetric matrix $\mathbf{S}$, called the consensus matrix (see Definition 1.1).
- $\mathbf{P}$ is a doubly stochastic matrix.
- $\mathbf{D}$ denotes a square, diagonal matrix. When multiple diagonal matrices are under consideration subscripts will be used.
- The column vector of all zeros except for a one in position i: $\mathbf{e}_i$
- The column vector of all ones: $\mathbf{e}$
- The t^{th} iterate of the power method $\mathbf{x}_t^T = \mathbf{x}_{t-1}^T\mathbf{P} : \mathbf{x}_t^T$
- $\lambda_i(\mathbf{P})$ denotes the i^{th} largest eigenvalue of $\mathbf{P}$. This notation will only be used when all of the eigenvalues of $\mathbf{P}$ are real and thus can be ordered.

CHAPTER 2

(Reverse) Simon-Ando Theory

The data clustering method introduced in this thesis is based on the variable aggregation work of the Nobel prize winning economist and twentieth century polymath Herbert Simon and his collaborator Albert Ando [74]. In Section 2.1 we will introduce the assumptions and conclusions of Simon-Ando theory. To keep the exposition as uncluttered as possible this section will refer to the data clustering application only when absolutely necessary. In Section 2.4, we will return to the data clustering problem and show how a reinterpretation of this theory suggests a clustering algorithm.

2.1 Simon-Ando theory, Part One

Simon-Ando theory was originally designed to help understand the short and long-term behavior of a large economy that could be divided into two or more almost independent

economies. One example of such a system would be a collection of robust national economies that have little interaction with each other. Another example would be a set of industrial sectors where trade is prevalent between companies in the same sector but not across sectors. Figure 2.1 illustrates a simple Simon-Ando system and shows how to construct a matrix that represents such a system.

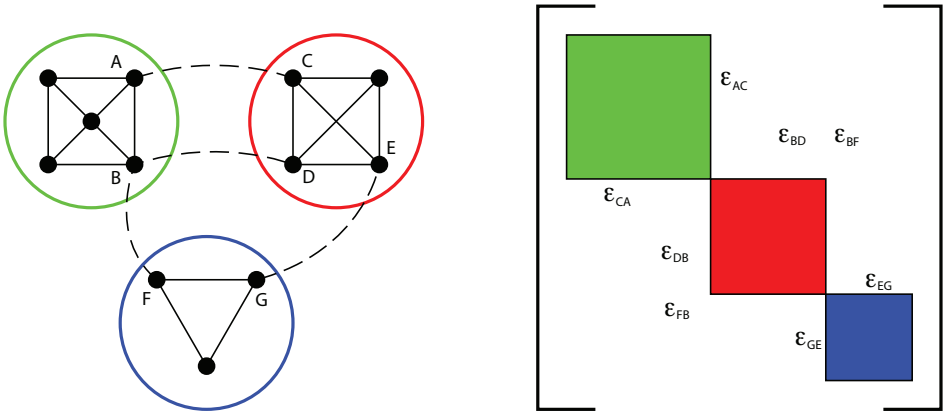


Figure 2.1: A simple Simon-Ando system in pictorial and matrix form. The circles on the left could represent three countries with strong internal trade (solid lines), but little international trade (dashed lines). Each matrix row and its corresponding column represent a particular industry. Thus the entry at row i, column j of the matrix represents some flow of goods or capital from company i to company j. For a Simon-Ando system, the matrix will have relatively large values in the diagonal blocks and relatively small ones elsewhere.

Such a closed economic system, without any outside influences, is known to eventually reach a state of equilibrium, that is, after some initial fluctuations, the flow of goods and capital between any two industries will remain more or less constant. Rather than waiting for this economic equilibrium to occur, Simon and Ando tried to predict the long-term equilibrium by making only short-term observations. They proved that what happens in the short run completely determines the long-term equilibrium.

Over the years scholars in a variety of disciplines have realized the usefulness of a framework that represents a number of tightly-knit groups that have some loose association with each other, and Simon-Ando theory has been applied in areas as diverse as ecology [58], computer queueing systems [16], brain organization [80], and urban design [70]. Simon himself went on to apply the theory to the evolution of multicellular organisms [73].

Definition 2.1. *An $n \times n$ real-valued matrix $\mathbf{P}$ is* uncoupled *if there exists a permutation matrix $\mathbf{Q}$ such that*

$$\mathbf{QPQ}^T = \begin{bmatrix} \mathbf{P}^\star_{11} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^\star_{22} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{P}^\star_{kk} \end{bmatrix},$$

where each $\mathbf{P}^\star_{ii}$ is square.

A matrix with the structure of the one in Figure 2.1 is not uncoupled since there are non-zero values in the off-diagonal blocks. But these values are small relative to the magnitude of the diagonal blocks, so we will use the term *nearly uncoupled* to describe such a matrix and beg the reader's indulgence on the imprecision of such a term until we develop a measure of "near uncoupledness" in Section 3.2.3.

If the consensus similarity matrix $\mathbf{S}$ defined in section 1.2 is nearly uncoupled, this thesis aims to show how Simon-Ando theory can be used to cluster the data it describes. Matrix $\mathbf{S}$ also has other properties not required by Simon-Ando but which will be useful as we develop this clustering method.

First, since $\mathbf{S}$ was constructed to show how often element i clustered with element j it is necessarily true that $s_{ij} = s_{ji}$, that is $\mathbf{S}$ is symmetric. The fact that $\mathbf{S}$ is both nearly

uncoupled and symmetric allows us to make another statement about its structure.

Definition 2.2. [1] *An $n \times n$ real-valued matrix $\mathbf{S}$ is* reducible *if there exists a permutation matrix $\mathbf{Q}$ such that*

$$\mathbf{QSQ}^T = \begin{bmatrix} \mathbf{X} & \mathbf{Z} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix},$$

where both $\mathbf{X}$ and $\mathbf{Y}$ are square.

Definition 2.3. *A matrix $\mathbf{S}$ is* irreducible *if it is not reducible.*

Theorem 2.4. *If $\mathbf{P}$ is a symmetric matrix, then $\mathbf{P}$ is irreducible if and only if $\mathbf{P}$ is not uncoupled.*

Proof. ($\Rightarrow$) By contradiction. Assume $\mathbf{P}$ is irreducible and uncoupled. Since $\mathbf{P}$ is uncoupled there exists a permutation matrix $\mathbf{Q}$ such that

$$\mathbf{QPQ}^T = \begin{bmatrix} \mathbf{P}_{11}^\star & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{22}^\star & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{P}_{kk}^\star \end{bmatrix} = \begin{bmatrix} \mathbf{X} & \mathbf{Z} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix},$$

where $\mathbf{X} = \mathbf{P}_{11}^\star$ and

$$\mathbf{Y} = \begin{bmatrix} \mathbf{P}_{22}^\star & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{P}_{kk}^\star \end{bmatrix}.$$

But $\mathbf{Q}$ also demonstrates that $\mathbf{P}$ is reducible, which is a contradiction and thus $\mathbf{P}$ is not uncoupled.

[1] From here through page 14, definitions are given for some common terms in the study of nonnegative matrices and of Markov chains that may be unknown to the general reader. If more background is needed, consult any good treatment of nonnegative matrices (for example, Chapter 8 in either [39] or [60]).

($\Leftarrow$) By contradiction. Assume $\mathbf{P}$ is not uncoupled and reducible. Since $\mathbf{P}$ is reducible there exists a permutation matrix $\mathbf{Q}$ such that

$$\mathbf{QPQ}^T = \begin{bmatrix} \mathbf{X} & \mathbf{Z} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix},$$

where $\mathbf{X}$ and $\mathbf{Y}$ are both square. Since $\mathbf{P}$ is symmetric, this implies

$$\mathbf{QPQ}^T = \begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix},$$

which means $\mathbf{P}$ is uncoupled - a contradiction. So, $\mathbf{P}$ is irreducible. □

Definition 2.5. *An $n \times n$ real-valued matrix $\mathbf{P}$ whose entries all lie in the interval $[0, 1]$ is* row stochastic *if*

$$\sum_{j=1}^{n} p_{ij} = 1 \quad \text{for each } \; i = 1, 2, \ldots, n.$$

and column stochastic *if*

$$\sum_{i=1}^{n} p_{ij} = 1 \quad \text{for each } \; j = 1, 2, \ldots, n.$$

Definition 2.6. *A matrix $\mathbf{P}$ is* doubly stochastic *if it is both row and column stochastic.*

Definition 2.7. *The row vector $\boldsymbol{\pi}^T$ is a* stationary distribution vector *of the stochastic matrix $\mathbf{P}$ if it satisfies the equations*

$$\begin{aligned} \boldsymbol{\pi}^T &= \boldsymbol{\pi}^T\mathbf{P}, \\ \boldsymbol{\pi}^T &\geq \mathbf{0} \qquad \textit{i.e. each element of } \boldsymbol{\pi}^T \textit{ is nonnegative, and} \\ \boldsymbol{\pi}^T\mathbf{e} &= 1. \end{aligned}$$

For reasons that will soon become apparent, our clustering method will also require that we convert the matrix $\mathbf{S}$ into doubly stochastic form, creating a new matrix we will call $\mathbf{P}$. We will address how this is done in Chapter 3 along with proving that such a conversion preserves irreducibility, symmetry, and near uncoupledness. The preservation of these properties is important since our new clustering algorithm uses the contents of the stationary distribution vector of $\mathbf{P}$ to identify clusters. Irreducibility guarantees the existence and uniqueness of the stationary distribution vector ([72], p. 119), while double stochasticity guarantees its form.

2.2 Stochastic complementation

We will now take a short detour to define the *stochastic complement* of a diagonal block $\mathbf{P}_{ii}$ and prove some results concerning stochastic complements that will aid in the explanation of Simon-Ando theory and in the development of our clustering algorithm.

Definition 2.8. *(Meyer [59]) If* $\mathbf{P}$ *is an irreducible, stochastic matrix with the structure*

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1k} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \mathbf{P}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{k1} & \mathbf{P}_{k2} & \dots & \mathbf{P}_{kk} \end{bmatrix},$$

then each diagonal block $\mathbf{P}_{ii}$ *has a stochastic complement in* $\mathbf{P}$ *defined by*

$$\mathbf{C}_{ii} = \mathbf{P}_{ii} + \mathbf{P}_{i\star} \left(\mathbf{I} - \mathbf{P}_i\right)^{-1} \mathbf{P}_{\star i}, \tag{2.1}$$

where $\mathbf{P}_i$ *is the matrix obtained by deleting the* i*th row and* i*th column of blocks from* $\mathbf{P}$*,*

$\mathbf{P}_{i\star}$ *is the ith row of blocks of* $\mathbf{P}$ *with* $\mathbf{P}_{ii}$ *removed, and* $\mathbf{P}_{\star i}$ *is the ith column of blocks of* $\mathbf{P}$ *with* $\mathbf{P}_{ii}$ *removed.* (See Figure 2.2 for an example using this definition.)

We are assured that $(\mathbf{I}-\mathbf{P}_i)^{-1}$ in (2.1) exists since every principal submatrix of $\mathbf{I}-\mathbf{P}$ of order $n-1$ or smaller is a nonsingular M-matrix. Furthermore, each of the entries of $(\mathbf{I}-\mathbf{P}_i)^{-1}$ is nonnegative [8].

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} & \mathbf{P}_{14} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} & \mathbf{P}_{24} \\ \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} & \mathbf{P}_{34} \\ \mathbf{P}_{41} & \mathbf{P}_{42} & \mathbf{P}_{43} & \mathbf{P}_{44} \end{bmatrix}$$

$$\mathbf{C}_{22} = \mathbf{P}_{22} + \mathbf{P}_{2\star}\left(\mathbf{I}-\mathbf{P}_2\right)^{-1}\mathbf{P}_{\star 2}$$

$$\mathbf{C}_{22} = \mathbf{P}_{22} + \begin{bmatrix} \mathbf{P}_{21} & \mathbf{P}_{23} & \mathbf{P}_{24} \end{bmatrix} \begin{bmatrix} \mathbf{I}-\mathbf{P}_{11} & -\mathbf{P}_{13} & -\mathbf{P}_{14} \\ -\mathbf{P}_{31} & \mathbf{I}-\mathbf{P}_{33} & -\mathbf{P}_{34} \\ -\mathbf{P}_{41} & -\mathbf{P}_{43} & \mathbf{I}-\mathbf{P}_{44} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{P}_{12} \\ \mathbf{P}_{32} \\ \mathbf{P}_{42} \end{bmatrix}$$

Figure 2.2: The equation for the stochastic complement $\mathbf{C}_{22}$ when $\mathbf{P}$ is a matrix with four square diagonal blocks.

The stochastic complements of $\mathbf{P}$ share some of the same characteristics as pointed out in this theorem.

Theorem 2.9. *(Meyer [59]) If* $\mathbf{P}$ *is an irreducible, stochastic matrix then each stochastic complement* $\mathbf{C}_{ii}$ *is irreducible and stochastic.*

We need to extend this result to show that if $\mathbf{P}$ is irreducible and doubly stochastic, then so is each $\mathbf{C}_{ii}$. We will prove this after first establishing a lemma that provides a tool that helps with the theorem's proof.

Lemma 2.10. *(Meyer [59]) Let* $\mathbf{P}$ *be the* $n \times n$ *irreducible, doubly stochastic, nearly uncoupled matrix*

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1k} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \mathbf{P}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{k1} & \mathbf{P}_{k2} & \dots & \mathbf{P}_{kk} \end{bmatrix},$$

where each diagonal block $\mathbf{P}_{ii}$ *is square and of size* $n_i \times n_i$*. Let* $\mathbf{Q}$ *be the permutation matrix associated with an interchange of the first and* i*th block rows and let* $\tilde{\mathbf{P}}$ *be defined as*

$$\tilde{\mathbf{P}} = \mathbf{Q}\mathbf{P}\mathbf{Q}^T.$$

If $\tilde{\mathbf{P}}$ *is partitioned into a* 2×2 *block matrix*

$$\tilde{\mathbf{P}} = \begin{bmatrix} \tilde{\mathbf{P}}_{11} & \tilde{\mathbf{P}}_{12} \\ \tilde{\mathbf{P}}_{21} & \tilde{\mathbf{P}}_{22} \end{bmatrix} \quad \textit{where} \quad \tilde{\mathbf{P}}_{11} = \mathbf{P}_{ii}, \tag{2.2}$$

then the stochastic complement of $\mathbf{P}_{ii}$ *in* $\mathbf{P}$ *is*

$$\mathbf{C}_{ii} = \tilde{\mathbf{C}}_{11} = \tilde{\mathbf{P}}_{11} + \tilde{\mathbf{P}}_{12}\left(\mathbf{I} - \tilde{\mathbf{P}}_{22}\right)^{-1}\tilde{\mathbf{P}}_{21}. \tag{2.3}$$

Proof. First let us demonstrate how to construct $\mathbf{Q}$. Let a be the number of the first row (column) of $\mathbf{P}_{ii}$, and b the number of the last row (column) of $\mathbf{P}_{ii}$. These two quantities can be defined as

$$\begin{aligned} a &= n_1 + n_2 + \cdots + n_{i-1} + 1, \quad \text{and} \\ b &= n_1 + n_2 + \cdots + n_i. \end{aligned}$$

$\mathbf{Q}$ can then be constructed to be the permutation matrix corresponding to the permutation

$$\begin{array}{ll} (a, a+1, \ldots, b, 1, \ldots, a-1) & \text{if } b = n, \text{ or} \\ (a, a+1, \ldots, b, 1, \ldots, a-1, b+1, \ldots, n) & \text{otherwise.} \end{array}$$

It now needs to be shown that the expressions for $\mathbf{C}_{ii}$ in (2.1) and (2.3) are equivalent, that is

$$\mathbf{P}_{ii} + \mathbf{P}_{i\star} \left(\mathbf{I} - \mathbf{P}_i\right)^{-1} \mathbf{P}_{\star i} = \tilde{\mathbf{P}}_{11} + \tilde{\mathbf{P}}_{12} \left(\mathbf{I} - \tilde{\mathbf{P}}_{22}\right)^{-1} \tilde{\mathbf{P}}_{21}.$$

Since $\tilde{\mathbf{P}}_{11} = \mathbf{P}_{ii}$ this simplifies to showing

$$\mathbf{P}_{i\star} \left(\mathbf{I} - \mathbf{P}_i\right)^{-1} \mathbf{P}_{\star i} = \tilde{\mathbf{P}}_{12} \left(\mathbf{I} - \tilde{\mathbf{P}}_{22}\right)^{-1} \tilde{\mathbf{P}}_{21}.$$

Since $\mathbf{Q}$ is constructed such that the relative order of the rows and columns that are not a part of $\mathbf{P}_{ii}$ are not changed, $\mathbf{P}_{i\star} = \tilde{\mathbf{P}}_{12}$, $\mathbf{P}_i = \tilde{\mathbf{P}}_{22}$, and $\mathbf{P}_{\star i} = \tilde{\mathbf{P}}_{21}$. Thus $\mathbf{C}_{ii} = \tilde{\mathbf{C}}_{11}$. □

Theorem 2.11. *If*

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \ldots & \mathbf{P}_{1k} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \ldots & \mathbf{P}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{k1} & \mathbf{P}_{k2} & \ldots & \mathbf{P}_{kk} \end{bmatrix}$$

is an irreducible, doubly stochastic matrix, then each stochastic complement is also an irreducible, doubly stochastic matrix.

Proof. We need only prove that $\mathbf{C}_{ii}$ is doubly stochastic. For a given i, suppose diagonal block $\mathbf{P}_{ii}$ has been repositioned such that $\tilde{\mathbf{P}}_{11} = \mathbf{P}_{ii}$ as in (2.2) of Lemma 2.10.

Since the permutation matrix $\mathbf{Q}$ only reordered the rows and columns, it could not change row or column sums, and hence both the row and column sums of $\tilde{\mathbf{P}}$ are one.

Allowing the size of $\mathbf{e}$, the column vector of all ones, to be whatever is appropriate for the context, the following four equations are true.

$$\tilde{\mathbf{P}}_{11}\mathbf{e} + \tilde{\mathbf{P}}_{12}\mathbf{e} = \mathbf{e} \tag{2.4}$$

$$\tilde{\mathbf{P}}_{21}\mathbf{e} + \tilde{\mathbf{P}}_{22}\mathbf{e} = \mathbf{e} \tag{2.5}$$

$$\mathbf{e}^T\tilde{\mathbf{P}}_{11} + \mathbf{e}^T\tilde{\mathbf{P}}_{21} = \mathbf{e}^T \tag{2.6}$$

$$\mathbf{e}^T\tilde{\mathbf{P}}_{12} + \mathbf{e}^T\tilde{\mathbf{P}}_{22} = \mathbf{e}^T \tag{2.7}$$

Equations (2.5) and (2.7) can be rewritten to yield

$$\mathbf{e} = \left(\mathbf{I} - \tilde{\mathbf{P}}_{22}\right)^{-1}\tilde{\mathbf{P}}_{21}\mathbf{e} \quad \text{and} \quad \mathbf{e}^T = \mathbf{e}^T\tilde{\mathbf{P}}_{12}\left(\mathbf{I} - \tilde{\mathbf{P}}_{22}\right)^{-1}.$$

As noted earlier $(\mathbf{I} - \tilde{\mathbf{P}}_{22})^{-1} \geq 0$ and hence

$$\tilde{\mathbf{C}}_{11} = \tilde{\mathbf{P}}_{11} + \tilde{\mathbf{P}}_{12}\left(I - \tilde{\mathbf{P}}_{22}\right)^{-1}\tilde{\mathbf{P}}_{21} \geq 0.$$

Multiplying $\tilde{\mathbf{C}}_{11}$ on the right by $\mathbf{e}$ yields

$$\tilde{\mathbf{C}}_{11}\mathbf{e} = \tilde{\mathbf{P}}_{11}\mathbf{e} + \tilde{\mathbf{P}}_{12}\left(I - \tilde{\mathbf{P}}_{22}\right)^{-1}\tilde{\mathbf{P}}_{21}\mathbf{e} = \tilde{\mathbf{P}}_{11}\mathbf{e} + \tilde{\mathbf{P}}_{12}\mathbf{e} = \mathbf{e},$$

while multiplying it on the left by $\mathbf{e}^T$ gives

$$\mathbf{e}^T\tilde{\mathbf{C}}_{11} = \mathbf{e}^T\tilde{\mathbf{P}}_{11} + \mathbf{e}^T\tilde{\mathbf{P}}_{12}\left(I - \tilde{\mathbf{P}}_{22}\right)^{-1}\tilde{\mathbf{P}}_{21} = \mathbf{e}^T\tilde{\mathbf{P}}_{11} + \mathbf{e}^T\tilde{\mathbf{P}}_{21} = \mathbf{e}^T.$$

Therefore, since $\mathbf{C}_{ii} = \tilde{\mathbf{C}}_{11}$, each stochastic complement is doubly stochastic. □

The following corollary is essential for using Simon-Ando theory in cluster analysis.

Corollary 2.12. *If*

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1k} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \mathbf{P}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{k1} & \mathbf{P}_{k2} & \dots & \mathbf{P}_{kk} \end{bmatrix}$$

is an $n \times n$ irreducible, doubly stochastic matrix, then the stationary distribution vector of the $n_i \times n_i$ stochastic complement $\mathbf{C}_{ii}$ is

$$\mathbf{c}_i^T = (\frac{1}{n_i} \quad \frac{1}{n_i} \quad \dots \quad \frac{1}{n_i}).$$

Proof. By definition, $\mathbf{c}_i^T$ must satisfy the equation $\mathbf{c}_i^T = \mathbf{c}_i^T \mathbf{C}_{ii}$, that is $\mathbf{c}_i^T$ must be the left hand eigenvector of $\mathbf{C}_{ii}$ associated with the eigenvalue 1. Since $\mathbf{C}_{ii}$ is irreducible by Theorem 2.9, the Perron-Frobenius theorem guarantees that $\mathbf{c}_i^T$ exists and is unique.

Since $\mathbf{C}_{ii}$ is doubly stochastic, we can also consider the equation

$$\begin{aligned} \left(\mathbf{c}_i^T \mathbf{C}_{ii}\right)^T &= \left(\mathbf{c}_i^T\right)^T \\ \mathbf{C}_{ii}^T \mathbf{c}_i &= \mathbf{c_i} \end{aligned}$$

which means $\mathbf{c_i}$ is the right hand eigenvector of $\mathbf{C}_{ii}$ associated with the eigenvalue 1, which for stochastic matrices is the constant vector. So now $\mathbf{c}_i^T$ must be a nonnegative, constant vector whose elements sum to 1. Therefore,

$$\mathbf{c}_i^T = (\frac{1}{n_i} \quad \frac{1}{n_i} \quad \dots \quad \frac{1}{n_i}).$$

Therefore the stationary distribution vector for each stochastic complement is the uniform

distribution vector. □

That concludes our discussion of stochastic complementation. We will now examine how the elements in a probability distribution vector change over repeated multiplications by the matrix $\mathbf{P}$. As we will see, one of the central results of Simon-Ando theory is that these changes follow a predictable pattern.

2.3 Simon-Ando theory, Part Two

Let $\mathbf{x}_0^T$ be a probability row vector and consider the evolution equation

$$\mathbf{x}_t^T = \mathbf{x}_{t-1}^T \mathbf{P} \tag{2.8}$$

or its equivalent formulation

$$\mathbf{x}_t^T = \mathbf{x}_0^T \mathbf{P}^t. \tag{2.9}$$

Simon-Ando theory asserts that $\mathbf{x}_t^T$ passes through distinct stages as t goes to infinity. Initially, $\mathbf{x}_t^T$ goes through changes driven by the comparatively large values in each diagonal block $\mathbf{P}_{ii}$. Once these changes have run their course, the elements of $\mathbf{x}_t^T$ settle into a short period of stabilization before the small values in the off-diagonal blocks affect small, but predictable changes in $\mathbf{x}_t^T$.[2]

When $\mathbf{P}$ is a stochastic matrix, the structure of $\mathbf{x}_t^T$ during these periods of stabilization and predictable change can be described in terms of each stochastic complements' stationary distribution vectors [59]. In particular, during the short-term stabilization

$$\mathbf{x}_t^T \approx (\alpha_1 \mathbf{c}_1 \;\; \alpha_2 \mathbf{c}_2 \;\; \ldots \;\; \alpha_k \mathbf{c}_k) \tag{2.10}$$

[2]See pages 118 - 127 Simon and Ando's original paper [74] for additional details. For a more modern proof of Simon and Ando's results, see [36].

where each α_i is a constant dependent on the initial probability vector $\mathbf{x}_0^T$. During the period of predictable change, which we will call middle-run evolution,

$$\mathbf{x}_t^T \approx (\beta_1 \mathbf{c}_1 \;\; \beta_2 \mathbf{c}_2 \;\; \ldots \;\; \beta_k \mathbf{c}_k) \tag{2.11}$$

where each β_i is dependent on t.

Since we know the stationary probability distribution of a doubly stochastic matrix, for the matrices considered in this thesis, (2.10) and (2.11) become

$$\mathbf{x}_t^T \approx \left(\frac{\alpha_1}{n_1} \frac{\alpha_1}{n_1} \ldots \frac{\alpha_1}{n_1} \middle| \frac{\alpha_2}{n_2} \frac{\alpha_2}{n_2} \ldots \frac{\alpha_2}{n_2} \middle| \ldots \middle| \frac{\alpha_k}{n_k} \frac{\alpha_k}{n_k} \ldots \frac{\alpha_k}{n_k} \right) \tag{2.12}$$

$$\mathbf{x}_t^T \approx \left(\frac{\beta_1}{n_1} \frac{\beta_1}{n_1} \ldots \frac{\beta_1}{n_1} \middle| \frac{\beta_2}{n_2} \frac{\beta_2}{n_2} \ldots \frac{\beta_2}{n_2} \middle| \ldots \middle| \frac{\beta_k}{n_k} \frac{\beta_k}{n_k} \ldots \frac{\beta_k}{n_k} \right). \tag{2.13}$$

Though $\mathbf{x}_t^T$ is not precisely equal to a vector created by concatenating multiples of each stochastic complement's stationary distribution vector, the bound on the difference between the left and right sides in Equations (2.10) and (2.11) is well-defined [59].

Specifically, if we assume that

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{C}_{kk} \end{bmatrix}$$

and $\mathbf{Z}^{-1}\mathbf{C}\mathbf{Z}$ is a diagonal matrix[3] then

$$||\mathbf{x}_t^T - (\alpha_1\mathbf{c}_1 \;\; \alpha_2\mathbf{c}_2 \;\; \ldots \;\; \alpha_k\mathbf{c}_k)|| \leq t\delta + ||\mathbf{Z}||_\infty||\mathbf{Z}^{-1}||_\infty|\lambda_{k+1}|^t \tag{2.14}$$

where $\delta = 2\max_i ||\mathbf{P}_{i\star}||_\infty$ and $|\lambda_{k+1}|$ is the magnitude of the largest eigenvalue of $\mathbf{C}$ not equal to one.

If during short-term equilibrium $||\mathbf{x}_t^T - (\alpha_1\mathbf{c}_1 \;\; \alpha_2\mathbf{c}_2 \;\; \ldots \;\; \alpha_k\mathbf{c}_k)||$ is always less than ϵ, then throughout middle-run evolution

$$||\mathbf{x}_t^T - (\beta_1\mathbf{c}_1 \;\; \beta_2\mathbf{c}_2 \;\; \ldots \;\; \beta_k\mathbf{c}_k)|| \leq \epsilon \tag{2.15}$$

The fact that $\mathbf{x}_t^T$ has this predictable structure is key to using Simon-Ando theory for data clustering. But, since this theory was developed not for clustering data, but for forecasting long-term economic trends, we need to look at it from a different angle which is the topic of the next section.

2.4 (Reverse) Simon-Ando Theory

Simon and Ando were not interested in clustering data. For them, the importance of stages like short-term stabilization and middle-run evolution lie in the fact that even for small values of t, the structure of $\mathbf{x}_t^T$ reflected the stationary probability vectors of the smaller $\mathbf{P}_{ii}$ matrices. From there, examination of the $\mathbf{x}_t^T$ vector during the relatively stable periods would allow for determination of these smaller stationary probability vectors and facilitate the calculation of the stationary probability vector for $\mathbf{P}$.

[3] $\mathbf{C}$ is not required to be diagonalizable. See pp. 266-267 of [59] for the bound if $\mathbf{C}$ is reduced to Jordan canonical form.

For cluster analysis however, the focus is turned around. Since we will be using doubly stochastic $\mathbf{P}$ matrices, we already know that the stationary probability vector is the uniform probability vector. We also know that each diagonal block $\mathbf{P}_{ii}$ is associated with a uniform probability vector related to its stochastic complement. Identification of the clusters then comes down to examining the entries of $\mathbf{x}_t^T$.

Throughout this chapter we have presented the matrix $\mathbf{P}$ with its rows and columns arranged so that its nearly uncoupled, block diagonal structure was clear. When working with actual data, $\mathbf{P}$ is not going to be in such an easy to interpret form. But the knowledge of the structure of $\mathbf{x}_t^T$ means we need only look for approximately equal values. For example, if

$$\mathbf{x}_t^T = (0.1497 \quad 0.1839 \quad 0.1793 \quad 0.1509 \quad 0.1836 \quad 0.1526),$$

then this suggests that we may want to place the first, fourth, and sixth elements in one cluster, and the second, third, and fifth in another. A more detailed discussion on determining the relevant gaps is given in Section 5.1.

All the development in this chapter assumed a doubly stochastic matrix. The next chapter demonstrates how we can convert a matrix to doubly stochastic form, and that the process does not destroy any of the desirable characteristics of our matrix.

CHAPTER 3

Matrix Scaling

We begin this chapter with two definitions.

Definition 3.1. *(Meyer [60], p. 661) If all the entries of an $m \times n$ matrix $\mathbf{A}$ are positive, that is $a_{ij} > 0$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, then $\mathbf{A}$ is a* positive matrix. *Similarly, if all $a_{ij} \geq 0$, $\mathbf{A}$ is a* nonnegative matrix. *$\mathbf{A} > \mathbf{0}$ denotes a positive matrix, and $\mathbf{A} \geq \mathbf{0}$ denotes a nonnegative matrix.*

Definition 3.2. *(Golub and van Loan [32], pp. 72-74)* Matrix scaling *is the multiplication of a matrix by a diagonal matrix with positive diagonal entries. If $\mathbf{B}$ is an $m \times n$ matrix, and both $m \times m$ matrix $\mathbf{D}_1$ and $n \times n$ matrix $\mathbf{D}_2$ are diagonal matrices with positive diagonals, then*

$$\hat{\mathbf{B}} = \mathbf{D}_1 \mathbf{B} \mathbf{D}_2$$

is a scaling of $\mathbf{B}$. If $\mathbf{D}_2 = \mathbf{I}$ the process is sometimes called row scaling, *and similarly if*

$\mathbf{D}_1 = \mathbf{I}$ *it is called* column scaling.

Matrix scaling is used in numerical linear algebra algorithms for solving linear systems, computing eigenvalues and evaluating matrix functions (see sections 4.5, 7.5, and 11.3, respectively of [32]) as a way to control the effect extremely large or extremely small values have on the solution.

In the context of this paper, scaling a matrix $\mathbf{B}$ will refer to combined row and column scaling, i.e. $\hat{\mathbf{B}} = \mathbf{D}_1\mathbf{B}\mathbf{D}_2$, with the goal of

$$\sum_{j=1}^{n} \hat{b}_{ij} = \rho \quad \text{for} \quad i = 1, 2, \ldots m \tag{3.1}$$

and

$$\sum_{i=1}^{m} \hat{b}_{ij} = \chi \quad \text{for} \quad j = 1, 2, \ldots n, \tag{3.2}$$

where ρ and χ are positive constants.

According to several sources ([10, 52, 68]) the first paper written about this type of matrix scaling is a 1937 article about telephone networks in a German engineering journal [51]. The method described in that paper and in much of the work to follow is primarily concerned with developing iterative methods for finding $\mathbf{D}_1$ and $\mathbf{D}_2$ [11], and the idea seems to have grown independently in different fields, with the result being a variety of algorithms which have much in common [71].

In our application of Simon-Ando theory to data clustering we are interested in changing the consensus similarity matrix $\mathbf{S}$ into doubly stochastic form, that is $\rho = \chi = 1$ in (3.1) and (3.2). This problem has also drawn considerable attention, and in 1964 Sinkhorn showed that any positive square matrix could be scaled to a unique doubly stochastic matrix [75].[1] As we will see shortly, in 1967 Sinkhorn and Knopp extended this result to

[1]It is a testament to the enduring interest in the doubly stochastic scaling problem, that at least six

nonnegative matrices under certain conditions. Though others independently discovered many of the same results, in the numerical linear algebra community today, a method for scaling a matrix into doubly stochastic form is typically called a Sinkhorn-Knopp algorithm [49].

We will save the details of the implementation of the Sinkhorn-Knopp algorithm for Section 3.3 and use the next two sections to focus on two questions:

1. Does the symmetric consensus matrix $\mathbf{S}$ meet the conditions to be scaled into a doubly stochastic matrix?

2. Will the result of this scaling destroy the irreducibility, symmetry, or near uncoupledness of $\mathbf{S}$?

3.1 Scaling S

It turns out that $\mathbf{S}$ will be scalable if its zero entries are in just the right places. The following definitions help describe the zero structure of matrices and are part of the hypotheses of the theorems that describe when Sinkhorn-Knopp scaling is possible.

Definition 3.3. *(Sinkhorn and Knopp [76]) A nonnegative $n \times n$ matrix $\mathbf{S}$ is said to have* total support *if $\mathbf{S} \neq \mathbf{0}$ and if every positive element of $\mathbf{S}$ lies on a positive diagonal, where a* diagonal *is defined as a sequence of elements $s_{1\sigma(1)}, s_{2\sigma(2)}, \ldots, s_{n\sigma(n)}$ where σ is a permutation of $\{1, 2, \ldots, n\}$.*[2]

Definition 3.4. *(Minc [64], p.82) An $n \times n$ matrix $\mathbf{S}$ is* partly indecomposable *if there*

proofs of Sinkhorn's 1964 results, all using different methods, have appeared in the literature, including one as late as 1998 [47].

[2]Notice that by this definition of *diagonal*, the main diagonal of a matrix is the diagonal associated with the permutation $\sigma = (1\ 2\ 3\ \ldots\ n)$.

exist permutation matrices $\mathbf{P}$ *and* $\mathbf{Q}$ *such that*

$$\mathbf{PSQ} = \begin{bmatrix} \mathbf{X} & \mathbf{Z} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix},$$

where $\mathbf{X}$ *and* $\mathbf{Y}$ *square.*

If no such $\mathbf{P}$ *and* $\mathbf{Q}$ *exist, then* $\mathbf{S}$ *is* fully indecomposable.

Figure 3.1 provides examples to explain the meaning of total support and draw the distinction between reducible and partly decomposable matrices.

$$\mathbf{A} = \begin{bmatrix} 0 & 12 & 0 \\ 9 & 0 & 2 \\ 5 & 0 & 8 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 12 & 2 \\ 9 & 0 & 0 \\ 5 & 0 & 8 \end{bmatrix}$$

(a) Matrix $\mathbf{A}$ has total support since the elements 12, 9, and 8 lie on the a_{12}, a_{21}, a_{33} diagonal and the remaining positive elements 2 and 5 along with 12 lie on the a_{12}, a_{23}, a_{31} diagonal. Matrix $\mathbf{B}$, however, does not have total support since the two diagonals that include $b_{13} = 2$, b_{13}, b_{21}, b_{32} and b_{13}, b_{22}, b_{31}, both contain a zero.

$$\mathbf{A} = \begin{bmatrix} 6 & 8 & 1 \\ 0 & 5 & 0 \\ 3 & 4 & 4 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{PAP}^T = \left[\begin{array}{cc|c} 2 & 3 & 4 \\ 1 & 6 & 8 \\ \hline 0 & 0 & 5 \end{array}\right]$$

$$\mathbf{B} = \begin{bmatrix} 0 & 5 & 0 \\ 1 & 8 & 6 \\ 2 & 4 & 3 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{PBQ} = \left[\begin{array}{cc|c} 2 & 3 & 4 \\ 1 & 6 & 8 \\ \hline 0 & 0 & 5 \end{array}\right]$$

(b) The terms reducible (Definition 2.2) and partly decomposable have very similar definitions. The rows and columns of both $\mathbf{A}$ and $\mathbf{B}$ can be permuted to obtain the same matrix ($\mathbf{PAP}^T = \mathbf{PBQ}$), the difference being that only one matrix (and its transpose) is needed to permute reducible matrix $\mathbf{A}$ while two matrices are needed to permute partly decomposable $\mathbf{B}$.

Figure 3.1: Some examples to illustrate the meanings of *total support*, *reducible*, and *partly decomposable*.

Definition 3.5. *(Minc [64], p.82) Two matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *are* permutation equivalent*, or* p-equivalent*, if there exist permutation matrices* $\mathbf{Q}$ *and* $\hat{\mathbf{Q}}$ *such that* $\mathbf{A} = \mathbf{Q}\mathbf{B}\hat{\mathbf{Q}}$*.*

This new terminology is needed to understand the following, nearly identical theorems that were independently proven within a year of each other, the first in 1966 and the second in 1967.

Theorem 3.6. *(Brualdi, Parter, and Schneider [12]) If the* $n \times n$ *matrix* $\mathbf{A}$ *is nonnegative and fully indecomposable, then there exist diagonal matrices* $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *with positive diagonal entries such that* $\mathbf{D}_1\mathbf{A}\mathbf{D}_2$ *is doubly stochastic. Moreover* $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *are uniquely determined up to scalar multiples.*

Theorem 3.7. *(Sinkhorn and Knopp [76]) If the* $n \times n$ *matrix* $\mathbf{A}$ *is nonnegative, then a necessary and sufficient condition that there exists a doubly stochastic matrix of the form* $\mathbf{D}_1\mathbf{A}\mathbf{D}_2$ *where* $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *are diagonal matrices with positive diagonal entries is that* $\mathbf{A}$ *has total support. If* $\mathbf{D}_1\mathbf{A}\mathbf{D}_2$ *exists, then it is unique. Also* $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *are unique up to a scalar multiple if and only if* $\mathbf{A}$ *is fully indecomposable.*

The uniqueness up to a scalar multiple of $\mathbf{D}_1$ and $\mathbf{D}_2$ mentioned in both theorems means that if $\mathbf{E}_1$ and $\mathbf{E}_2$ are also diagonal matrices such that $\mathbf{E}_1\mathbf{A}\mathbf{E}_2$ is doubly stochastic, then $\mathbf{E}_1 = \alpha\mathbf{D}_1$ and $\mathbf{E}_2 = \beta\mathbf{D}_2$ where $\alpha\beta = 1$.

The way that the consensus similarity matrix $\mathbf{S}$ is constructed guarantees its nonnegativity, so the only thing standing in the way of knowing that the scaling matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ exist is showing that $\mathbf{S}$ either has total support or is fully indecomposable. Reviewing the definitions of these terms, neither of these tasks seems inviting. Fortunately, there is a theorem that will simplify the matter.

Theorem 3.8. *(Minc [64], p.86) A nonnegative matrix is fully indecomposable if and only if it is p-equivalent to an irreducible matrix with a positive main diagonal.*

$\mathbf{S}$ is trivially p-equivalent since $\mathbf{S} = \mathbf{ISI}$, and $\mathbf{S}$ is an irreducible matrix with a positive main diagonal. Now that we know $\mathbf{S}$ is fully indecomposable, its symmetry is going to guarantee another excellent result. The proof of the following lemma is included since there was a typographical error in the one included in the original paper. Also the proof below explicitly shows the algebra leading to the relationship between $\mathbf{D}$ and $\mathbf{D}_1$.

Lemma 3.9. *(Csima and Datta [17]) Let* $\mathbf{S}$ *be a fully indecomposable symmetric matrix. Then there exists a diagonal matrix* $\mathbf{D}$ *such that* $\mathbf{DSD}$ *is doubly stochastic.*

Proof. Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be nonnegative diagonal matrices such that $\mathbf{D}_1\mathbf{SD}_2$ is doubly stochastic. Then $(\mathbf{D}_1\mathbf{SD}_2)^T = \mathbf{D}_2\mathbf{SD}_1$ is also doubly stochastic. By the uniqueness up to a scalar multiple from Theorems 3.6 and 3.7, we know $\mathbf{D}_2 = \alpha\mathbf{D}_1$ and $\mathbf{D}_1 = \beta\mathbf{D}_2$. Using the first of these facts

$$\begin{aligned} \mathbf{D}_1\mathbf{SD}_2 &= \mathbf{D}_1\mathbf{S}\alpha\mathbf{D}_1 \\ &= \sqrt{\alpha}\mathbf{D}_1\mathbf{S}\sqrt{\alpha}\mathbf{D}_1 \\ &= \mathbf{DSD} \end{aligned}$$

shows us that $\mathbf{D} = \sqrt{\alpha}\,\mathbf{D}_1$. □

3.2 The structure of DSD

We will use $\mathbf{P}$ as the symbol for the doubly stochastic matrix derived from $\mathbf{S}$, that is $\mathbf{P} = \mathbf{DSD}$. For simplicity of notation, the i^{th} diagonal entry of $\mathbf{D}$ will be denoted d_i. The next three subsections will demonstrate that like $\mathbf{S}$, $\mathbf{P}$ is irreducible, symmetric and nearly uncoupled.

3.2.1 Is P irreducible?

Lemma 3.10. *If* $\mathbf{S}$ *is an* $n \times n$ *fully indecomposable irreducible matrix and* $\mathbf{P} = \mathbf{DSD}$ *is doubly stochastic, then* $\mathbf{P}$ *is irreducible.*

Proof. Since $\mathbf{S}$ is irreducible, there is no permutation matrix $\mathbf{Q}$ such that

$$\mathbf{QSQ}^T = \begin{bmatrix} \mathbf{X} & \mathbf{Z} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix}.$$

where both $\mathbf{X}$ and $\mathbf{Y}$ are square.

Thus the only way that $\mathbf{P} = \mathbf{DSD}$ could be reducible is if the zero structure of $\mathbf{S}$ is changed by the multiplication. But notice that since $p_{ij} = d_i d_j s_{ij}$ and both d_i and d_j are positive, $p_{ij} = 0$ only when $s_{ij} = 0$. So the zero structure does not change, and $\mathbf{P}$ is irreducible. □

3.2.2 Is P symmetric?

Since the number of times elements i and j cluster with one another is necessarily equal to the number of times elements j and i cluster with one another, the symmetry of the consensus similarity matrix $\mathbf{S}$ reflects a real-world property of the consensus clustering problem and so it is important that symmetry is not lost when $\mathbf{S}$ is converted into $\mathbf{P}$.

Let us first note that some people's intuition leads them to believe that a doubly stochastic matrix is necessarily symmetric. However, notice that

$$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{6} & \frac{5}{12} & \frac{5}{12} \end{bmatrix}$$

is doubly stochastic but not symmetric.

Lemma 3.11. *If* $\mathbf{S}$ *is an* $n \times n$ *fully indecomposable symmetric matrix and* $\mathbf{P} = \mathbf{DSD}$ *is doubly stochastic, then* $\mathbf{P}$ *is symmetric.*

Proof.

$$\mathbf{P}^T = (\mathbf{DSD})^T = \mathbf{DS}^T\mathbf{D} = \mathbf{DSD} = \mathbf{P}$$

$\mathbf{P}$ is symmetric. □

3.2.3 Effect on nearly uncoupled form

We wish to prove that if $\mathbf{S}$ is nearly uncoupled, then so is $\mathbf{P}$. To do so we first need a formal definition of near uncoupledness. Then we will show how this uncoupling measure for $\mathbf{P}$ is related to the uncoupling measure of $\mathbf{S}$.

Definition 3.12. *Let* n_1 *and* n_2 *be fixed positive integers such that* $n_1 + n_2 = n$*, and let* $\mathbf{S}$ *be an* $n \times n$ *symmetric, irreducible matrix whose respective rows and columns have been rearranged to the form*

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix}$$

where $\mathbf{S}_{11}$ *is* $n_1 \times n_1$ *and* $\mathbf{S}_{22}$ *is* $n_2 \times n_2$ *so that the ratio*

$$\sigma(\mathbf{S}, n_1) = \frac{\mathbf{e}^T\mathbf{S}_{12}\mathbf{e} + \mathbf{e}^T\mathbf{S}_{21}\mathbf{e}}{\mathbf{e}^T\mathbf{S}\mathbf{e}} = \frac{2\mathbf{e}^T\mathbf{S}_{12}\mathbf{e}}{\mathbf{e}^T\mathbf{S}\mathbf{e}}$$

is minimized over all symmetric permutations of $\mathbf{S}$*. The quantity* $\sigma(\mathbf{S}, n_1)$ *is called the* uncoupling measure *of* $\mathbf{S}$ *with respect to parameter* n_1*. In other words* $\sigma(\mathbf{S}, n_1)$ *is the ratio of the sum of the elements in the off-diagonal blocks to the sum of all the matrix entries.*

Theorem 3.13. *If* $\mathbf{S}$ *is the* $n \times n$ *consensus matrix created from* r *clustering results and* $\sigma(\mathbf{S}, n_1) = \beta$*, then for the doubly stochastic matrix* $\mathbf{P} = \mathbf{DSD}$*,* $\sigma(\mathbf{P}, n_1) \leq \frac{\Sigma}{nr}\beta$*, where* $\Sigma = \mathbf{e}^T\mathbf{Se}$.

Proof. By the way we constructed $\mathbf{S}$, $s_{ii} = r$ for $i = 1, 2, \ldots, n$. Since $p_{ii} = d_i d_i s_{ii}$ and $p_{ii} \leq 1$, it follows that $d_i^2 r \leq 1 \rightarrow d_i \leq \frac{1}{\sqrt{r}}$.

If we impose the same block structure on $\mathbf{D}$ that exists for $\mathbf{S}$, that is

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix},$$

and recall that $\mathbf{P}$ is doubly stochastic,

$$\sigma(\mathbf{P}, n_1) = \frac{2\mathbf{e}^T\mathbf{D}_1\mathbf{S}_{12}\mathbf{D}_2\mathbf{e}}{n}$$

Since each element of $\mathbf{D}_1$ and $\mathbf{D}_2$ is less than $\frac{1}{\sqrt{r}}$,

$$\begin{aligned} \sigma(\mathbf{P}, n_1) &\leq \frac{\left(\frac{1}{\sqrt{r}}\right)^2 (2\mathbf{e}^T\mathbf{S}_{12}\mathbf{e})}{n} \\ &= \frac{\Sigma}{nr}\sigma(\mathbf{S}, n_1) \\ &= \frac{\Sigma}{nr}\beta \end{aligned}$$

The bound is found. □

3.3 The Sinkhorn-Knopp algorithm

If one were to naively approach writing a computer program to convert any nonnegative matrix $\mathbf{A}$ to doubly stochastic form the code would probably look like this:

1. Divide each element of the matrix by its row sum.
2. Divide each element of the matrix by it column sum.
3. Repeat until that last two iterations of steps 1 and 2 yield two matrices within a certain tolerance of each other.

It turns out that such an approach works. In terms of matrix multiplication this repetition of row/column scaling would be

$$\mathbf{P} = \mathbf{R}_k\mathbf{R}_{k-1}\dots\mathbf{R}_2\mathbf{R}_1\mathbf{A}\mathbf{C}_1\mathbf{C}_2\dots\mathbf{C}_{k-1}\mathbf{C}_k$$

where $\mathbf{R}_i$ is the row-scaling diagonal matrix used the i^{th} time through the loop and $\mathbf{C}_i$ is the respective column-scaling matrix. This equation can be simplified by multiplying all of the row and column-scaling matrices together to get

$$\mathbf{P} = \mathbf{D}_1\mathbf{A}\mathbf{D}_2.$$

Since in the end all that is needed are the two vectors that make up the diagonals of $\mathbf{D}_1$ and $\mathbf{D}_2$, many programmers just find them through an iterative method. Let $\mathbf{r}$ represent the diagonal of $\mathbf{D}_1$ and $\mathbf{c}$ the diagonal of $\mathbf{D}_2$, set $\mathbf{r} = \mathbf{e}$ and repeat the following two MATLAB command within a loop:

```
c = 1./(A'*r);
r = 1./(A*c);
```

In our case where the input matrix is symmetric, these two commands collapse into one

```
x = 1./(A*x);
```

though in this case $\mathbf{x}$ alternates between $\mathbf{c}$ and $\mathbf{r}$, so the vectors from two consecutive iterations have to be taken and then the $\sqrt{\alpha}$ from Lemma 3.9 is computed to find $\mathbf{d}$, the vector that becomes the diagonal of $\mathbf{D}$.

Whether using one or two MATLAB commands, the loop continues until a stopping criteria is met. Two typical stopping criteria that involve a tolerance ϵ and a norm $||\cdot||$ are

1. Stop when successive values of $\mathbf{c}$ and $\mathbf{r}$ differ by less than the tolerance, i.e.

$$||\mathbf{c}_{n+1} - \mathbf{c}_n|| < \epsilon \quad \text{and} \quad ||\mathbf{r}_{n+1} - \mathbf{r}_n|| < \epsilon.$$

2. Stop when

$$||\mathbf{D}_1\mathbf{A}\mathbf{D}_2\mathbf{e} - \mathbf{e}|| < \epsilon \quad \text{and} \quad ||\mathbf{e}^T - \mathbf{e}^T\mathbf{D}_1\mathbf{A}\mathbf{D}_2|| < \epsilon.$$

In general, using these stopping criteria, the Sinkhorn-Knopp algorithm converges linearly [78], that is

$$||\mathbf{d} - \mathbf{x}_{k+2}|| \leq \gamma||\mathbf{d} - \mathbf{x}_k|| \tag{3.3}$$

where $0 < \gamma < 1$. If $\gamma \approx 1$, convergence will be slow.

Knight has shown [49] that if $\mathbf{A}$ is symmetric, nonnegative, and fully indecomposable (which our $\mathbf{S}$ is), there exists a norm and there exists a point in the the iterative process after which

$$||\mathbf{d} - \mathbf{x}_{k+2}|| \leq |\lambda_2|^2||\mathbf{d} - \mathbf{x}_k|| \tag{3.4}$$

where λ_2 is the eigenvalue of $\mathbf{P}$ that is second largest in magnitude. Though this is encouraging, the upcoming discussion in Chapter 4 will show that in most clustering cases, λ_2 is quite close to one. That being said, it is still better than the general case of linear convergence, though a user must be aware of the possibility of slow convergence.

CHAPTER 4

The role of the second eigenvalue

As we now move on to consider the eigenvalues of $\mathbf{P}$, it will be helpful to have some common notation and vocabulary for eigenvalues in general.

Definition 4.1. *(Meyer [60], p. 490) If* $\mathbf{A}$ *is* $n \times n$ *the set of* $s \leq n$ *distinct eigenvalues is called the* spectrum *of* $\mathbf{A}$*. The elements of the spectrum are numbered so that*

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_s|.$$

If context demands we be specific about which matrix we are referring to, the notation $\lambda_i(\mathbf{B})$ *will be used to denote the* i^{th} *eigenvalue of matrix* $\mathbf{B}$*.*

Now that $\mathbf{S}$ has been converted into the doubly stochastic matrix $\mathbf{P}$, it turns out that an examination of the eigenvalues of $\mathbf{P}$ is quite helpful in determining the structure of the underlying data set. Here are some facts about the eigenvalues of $\mathbf{P}$ [60].

1. Since $\mathbf{P}$ is stochastic, all of its eigenvalues lie on or inside the unit circle of the complex plane.

2. Since $\mathbf{P}$ is real-symmetric, all of its eigenvalues are real. Combined with the last fact, this means all eigenvalues of $\mathbf{P}$ reside in the interval $[-1, 1]$.

3. The largest eigenvalue of $\mathbf{P}$ is one, and since $\mathbf{P}$ is irreducible, that eigenvalue is simple (i.e. it appears only once).

4. $\lambda_i(\mathbf{P}) \neq -1$ for any i because $\mathbf{P}$ is a primitive matrix. $\mathbf{P}$ is primitive because it is irreducible and has at least one positive diagonal element (p. 678, [60]).

For those who work with stochastic matrices, the magnitude of the eigenvalues other than $\lambda_1(\mathbf{P})$ is also of interest. For example, Markov chain researchers know that the asymptotic convergence rate, that is, the expected number of digits of accuracy gained in each iteration of calculating the chain's stationary distribution vector is $-\log_{10}|\lambda_2(\mathbf{P})|$ [53, 60].

Our aim however is quite different; we want a second eigenvalue near one. Slow convergence is a good thing for us since it allows time to examine the elements of $\mathbf{x}_t$ as it passes through short-term equilibrium and middle-run evolution. Also, $\lambda_2(\mathbf{P}) \approx 1$ *may* indicate that the matrix is nearly uncoupled [81]. Later in this chapter we will show that $\lambda_2(\mathbf{P}) \approx 1$ along with other properties of $\mathbf{P}$ *guarantees* that $\mathbf{P}$ is nearly uncoupled.

4.1 Nearly Uncoupled Form and $\lambda_2(\mathbf{P})$

We have already established that $\lambda_1(\mathbf{P}) = 1$ and that $\lambda_2(\mathbf{P}) < 1$. The goal of this section is to make a concrete connection between the size of $\lambda_2(\mathbf{P})$ and the near uncoupledness of $\mathbf{P}$. Those results will be proven in Theorems 4.4 and 4.5, after two lemmas.

Lemma 4.2. *If* $\{\mathbf{P}_k\}$ *is a sequence of symmetric matrices with limit* $\mathbf{P}_0$*, then* $\mathbf{P}_0$ *is symmetric.*

Proof.

$$\lim_{k\to\infty} \mathbf{P}_k = \mathbf{P}_0 \Rightarrow \lim_{k\to\infty} \mathbf{P}_k^T = \mathbf{P}_0^T$$

but

$$\lim_{k\to\infty} \mathbf{P}_k = \lim_{k\to\infty} \mathbf{P}_k^T \quad \text{since} \quad \mathbf{P}_k = \mathbf{P}_k^T.$$

So $\mathbf{P}_0 = \mathbf{P}_0^T$. □

Lemma 4.3. *If* $\{\mathbf{P}_k\}$ *is a sequence of stochastic matrices with limit* $\mathbf{P}_0$*, then* $\mathbf{P}_0$ *is stochastic.*

Proof. Since $\mathbf{P}_k$ is stochastic, $\mathbf{P}_k\mathbf{e} = \mathbf{e}$. It follows that

$$\begin{aligned} \lim_{k\to\infty} \mathbf{P}_k\mathbf{e} &= \lim_{k\to\infty} \mathbf{e} \\ \left(\lim_{k\to\infty} \mathbf{P}_k\right)\mathbf{e} &= \lim_{k\to\infty} \mathbf{e} \\ \mathbf{P}_0\mathbf{e} &= \mathbf{e} \end{aligned}$$

Therefore, $\mathbf{P}_0$ is stochastic. □

Theorem 4.4. *For a fixed integer* $n > 0$*, consider the* $n \times n$ *irreducible, symmetric, doubly stochastic matrix* $\mathbf{P}$*. Given* $\epsilon > 0$*, there exists a* $\delta > 0$ *such that if* $\sigma(\mathbf{P}) < \delta$*, then* $|\lambda_2(\mathbf{P}) - 1| < \epsilon$*. In other words, if* $\mathbf{P}$ *is sufficiently close to being uncoupled, then* $\lambda_2(\mathbf{P}) \approx 1$*.*

Proof. Let $\epsilon > 0$. Consider a sequence of irreducible, symmetric, doubly stochastic matrices

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{11}^{(k)} & \mathbf{P}_{12}^{(k)} \\ \mathbf{P}_{21}^{(k)} & \mathbf{P}_{22}^{(k)} \end{bmatrix}$$

defined so that $\lim_{k\to\infty} \sigma(\mathbf{P}_k) = 0$. The Bolzano-Weierstrass theorem ([4], p. 155) guarantees that this bounded sequence has a convergent subsequence $\mathbf{P}_{k_1}, \mathbf{P}_{k_2}, \ldots$ which converges to a stochastic matrix $\mathbf{C}$ whose structure is

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{bmatrix}, \quad \mathbf{C}_{11} \neq \mathbf{0}, \mathbf{C}_{22} \neq \mathbf{0},$$

where each $\mathbf{C}_{ii}$ is stochastic. By the well-known but rarely proven continuity of eigenvalues,[1] there exists an M such that for $k_i > M$,

$$\begin{aligned} |\lambda_2(\mathbf{P}_{k_i}) - \lambda_2(\mathbf{C})| &< \epsilon \\ |\lambda_2(\mathbf{P}_{k_i}) - 1| &< \epsilon, \end{aligned}$$

and the theorem is proven. □

Theorem 4.5. *For a fixed integer $n > 0$, consider the $n \times n$ irreducible, symmetric, doubly stochastic matrix $\mathbf{P}$. Given $\epsilon > 0$, there exists a $\delta > 0$ such that if $|\lambda_2(\mathbf{P}) - 1| < \delta$, then $\sigma(\mathbf{P}) < \epsilon$. In other words, if $\lambda_2(\mathbf{P})$ is sufficiently close to 1, then $\mathbf{P}$ is nearly uncoupled.*

Proof. By contradiction. Suppose there is an $\epsilon > 0$ such that for any $\delta > 0$ there is an $n \times n$ irreducible, symmetric, doubly stochastic matrix $\mathbf{P}$ with $|\lambda_2(\mathbf{P}) - 1| < \delta$

[1] The typical method of proof is to use Rouché's Theorem from complex analysis which states that the zeros of a polynomial equation are continuous functions of the polynomial's coefficients. This result is then applied to the characteristic polynomial (see [40] pp. 136-139 or [66] pp. 42-45). Those interested in an approach grounded in perturbation theory should consider the resolvent theory of Kato ([48], Section 1.5) and the exposition in [61].

and $\sigma(\mathbf{P}) > \epsilon$. For $\delta = \frac{1}{k}$ let $\mathbf{P}_k$ be such a matrix. Again, there must be a subsequence $\mathbf{P}_{i_1}, \mathbf{P}_{i_2}, \ldots$ which converges, say to $\mathbf{P}_0$. Then $\mathbf{P}_0$ must have $\lambda_2(\mathbf{P_0}) = 1$ and thus $\sigma(\mathbf{P_0}) = 0$. Yet, $\sigma(\mathbf{P_0}) = \lim_{k\to\infty} \sigma(\mathbf{P_k}) \geq \epsilon$, a contradiction. □

Although we already have an uncoupling measure $\sigma(\mathbf{S})$ for a general matrix, for doubly stochastic matrices this theorem allows us to use λ_2 as an uncoupling indicator with a value near one signifying almost complete uncoupling.

4.2 The Perron cluster

There may be additional eigenvalues of $\mathbf{P}$ that are close to one. This group of eigenvalues is called the *Perron cluster*, and in the case where all eigenvalues are real the Perron cluster can be defined as follows.

Definition 4.6. *Let* $\mathbf{P}$ *be an* $n \times n$ *symmetric, stochastic matrix with eigenvalues, including multiplicities, of* $1 = \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n$. *If the largest difference between consecutive eigenvalues occurs between* λ_k *and* λ_{k+1}, *the set* $\{1, \ldots \lambda_k\}$ *is called the* Perron cluster *of* $\mathbf{P}$. *The larger the gap, the more well-defined the cluster is.*

Some researchers use the number of eigenvalues in the Perron cluster as the number of clusters they search for [28, 18]. This inference is a natural extension of Theorems 4.4 and 4.5, that is if $\mathbf{P}$ had k eigenvalues sufficiently close to 1, then $\mathbf{P}$ is nearly uncoupled with k dominant diagonal blocks emerging after an appropriate permutation of $\mathbf{QPQ}^T$. This is also the approach we will take with the stochastic clustering algorithm. Unlike with the vast majority of clustering methods, the user will not have to tell the algorithm the number of clusters in the data set unless they explicitly want to override the algorithm's choice. Instead, the stochastic clustering algorithm will set k equal to the size of the Perron cluster.

CHAPTER 5

The Stochastic Clustering Algorithm

5.1 Putting the concept into practice

Now that the theoretical underpinnings are in place, it is time to formally describe the stochastic clustering algorithm.

The algorithm takes as input the consensus similarity matrix $\mathbf{S}$ which the user has created from whatever combination of clustering methods and/or parameter settings they choose. $\mathbf{S}$ is then converted into the doubly stochastic matrix $\mathbf{P}$ using the algorithm from Section 3.3. All eigenvalues are computed using MATLAB's `eig` command, and the Perron cluster of $\mathbf{P}$ is identified. For large-scale problems, the user can direct the program to find only the g largest eigenvalues and then identify the Perron cluster of that subset of the eigenvalues. The stochastic clustering algorithm then separates the data into k clusters, where k is the number of eigenvalues in the Perron cluster.

Stochastic Clustering Algorithm (SCA)

1. Create the consensus similarity matrix $\mathbf{S}$ using clustering ensemble of user's choice.
2. Use matrix balancing to convert $\mathbf{S}$ into a doubly stochastic symmetric matrix $\mathbf{P}$.
3. Calculate the necessarily real eigenvalues of $\mathbf{P}$. The number of clusters, k, is number of eigenvalues in the Perron cluster.
4. Create a random $\mathbf{x}_0^T$.
5. Track the evolution $\mathbf{x}_t^T = \mathbf{x}_{t-1}^T\mathbf{P}$. After each multiplication, sort the the elements of $\mathbf{x}_t^T$ and then separate the elements into k clusters by dividing the sorted list at the $k-1$ largest gaps. When this clustering has remained the same for a user-defined number of iterations, the final clusters have been determined.

Figure 5.1: The Stochastic Clustering Algorithm

Starting with a randomly generated $\mathbf{x}_0^T$, $\mathbf{x}_t^T = \mathbf{x}_{t-1}^T\mathbf{P}$ is evaluated. After each calculation, the entries of $\mathbf{x}_t^T$ are sorted, the $k-1$ largest gaps in the sorted list identified and used to divide the entries into k clusters. When starting the algorithm, the user inputs the number of consecutive identical clusterings needed to bring the algorithm to a close. Once that number is reached, the program stops and the clusters returned as output. Figure 5.1 summarizes the algorithm.

5.2 A Small Example

Consider the small data matrix $\mathbf{A}$ in Figure 5.2. Each column of the matrix contains the career totals in nine statistics for a famous baseball player. Those familiar with baseball history would probably group these six players into singles hitters (Rose and Cobb), power hitters (Mays, Ott, and Ruth), and a great catcher who doesn't necessarily fit into either

group (Fisk).

The consensus similarity matrix was built using the multiplicative update version of the nonnegative matrix factorization algorithm [54]. Since it is not clear whether two or three clusters would be most appropriate, $\mathbf{S}$ was created by running this algorithm 50 times with $k = 2$ and 50 times with $k = 3$.

With a small example like this, especially one where the players that will cluster together have been purposely placed in adjacent columns, it would be simple enough to cluster the players through a quick scan of $\mathbf{S}$. But since the purpose here is to illustrate how the algorithm works, we will continue by applying the Sinkhorn-Knopp algorithm to create a doubly stochastic matrix $\mathbf{P}$.

The eigenvalues of $\mathbf{P}$ are 1.00, 0.88, 0.19, 0.09, 0.06, and 0.03. The largest gap in this ordered list of eigenvalues is between 0.88 and 0.19, so there are two eigenvalues in the Perron cluster, and thus the stochastic clustering algorithm will look for two clusters in the data set.

Table 5.1 shows the results of a run of the stochastic clustering algoritm. The initial probability vector $\mathbf{x}_0^T$ was chosen randomly, and the table shows the value of $\mathbf{x}_t^T$ and the corresponding clusters for the next six steps of the algorithm. Since $k = 2$, the clusters are determined by ordering the entries of $\mathbf{x}_t$, finding the largest gap in this list, and clustering the elements on either side of this gap. For example, when $t = 4$ in Table 5.1 the ordered list would be

$$0.1601,\ 0.1606,\ 0.1612,\ 0.1716,\ 0.1732,\ 0.1733,$$

and the largest gap is between 0.1612 and 0.1716. This leads to the numerical clustering of $\{0.1601, 0.1606, 0.1612\}$ and $\{0.1716, 0.1732, 0.1733\}$, which translates to the player clustering {Rose, Cobb, Fisk} and {Ott, Ruth, Mays}.

$$\mathbf{A}= \begin{array}{c} \\ \text{G} \\ \text{R} \\ \text{H} \\ \text{2B} \\ \text{3B} \\ \text{HR} \\ \text{RBI} \\ \text{SB} \\ \text{BB} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{Rose} & \text{Cobb} & \text{Fisk} & \text{Ott} & \text{Ruth} & \text{Mays} \end{array} \\ \begin{pmatrix} 3562 & 3034 & 2499 & 2730 & 2503 & 2992 \\ 2165 & 2246 & 1276 & 1859 & 2174 & 2062 \\ 4256 & 4189 & 2356 & 2876 & 2873 & 3283 \\ 746 & 724 & 421 & 488 & 506 & 523 \\ 135 & 295 & 47 & 72 & 136 & 140 \\ 160 & 117 & 376 & 511 & 714 & 660 \\ 1314 & 1938 & 1330 & 1860 & 2213 & 1903 \\ 198 & 897 & 128 & 89 & 123 & 338 \\ 1566 & 1249 & 849 & 1708 & 2062 & 1464 \end{pmatrix} \end{array}$$

(a) The games played, runs, hits, doubles, triples, home runs, runs batted in, stolen bases, and bases on balls career totals for Pete Rose, Ty Cobb, Carlton Fisk, Mel Ott, Babe Ruth, and Willie Mays.[5]

$$\mathbf{S}= \begin{array}{c} \\ \text{Rose} \\ \text{Cobb} \\ \text{Fisk} \\ \text{Ott} \\ \text{Ruth} \\ \text{Mays} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{Rose} & \text{Cobb} & \text{Fisk} & \text{Ott} & \text{Ruth} & \text{Mays} \end{array} \\ \begin{pmatrix} 100 & 66 & 77 & 3 & 1 & 4 \\ 66 & 100 & 57 & 1 & 0 & 7 \\ 77 & 57 & 100 & 14 & 7 & 19 \\ 3 & 1 & 14 & 100 & 90 & 80 \\ 1 & 0 & 7 & 90 & 100 & 84 \\ 4 & 7 & 19 & 80 & 84 & 100 \end{pmatrix} \end{array}$$

(b) The consensus matrix.

$$\mathbf{P}= \begin{array}{c} \\ \text{Rose} \\ \text{Cobb} \\ \text{Fisk} \\ \text{Ott} \\ \text{Ruth} \\ \text{Mays} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{Rose} & \text{Cobb} & \text{Fisk} & \text{Ott} & \text{Ruth} & \text{Mays} \end{array} \\ \begin{pmatrix} 0.4004 & 0.2828 & 0.2872 & 0.0112 & 0.0038 & 0.0146 \\ 0.2828 & 0.4584 & 0.2275 & 0.0040 & 0 & 0.0273 \\ 0.2872 & 0.2275 & 0.3474 & 0.0486 & 0.0248 & 0.0645 \\ 0.0112 & 0.0040 & 0.0486 & 0.3465 & 0.3186 & 0.2711 \\ 0.0038 & 0 & 0.0248 & 0.3186 & 0.3618 & 0.2909 \\ 0.0146 & 0.0273 & 0.0645 & 0.2711 & 0.2909 & 0.3316 \end{pmatrix} \end{array}$$

(c) The doubly stochastic, symmetric matrix with entries rounded to four places.

Figure 5.2: The three matrices associated with our small example.

Table 5.1: The Stochastic Clustering Algorithm for the Small Example

t	$\mathbf{x}_t$	Clusters
0	(0.1735 0.1476 0.2110 0.1015 0.1465 0.2198)	{Rose, Cobb, Fisk, Ruth, Mays} {Ott}
1	(0.1767 0.1712 0.1798 0.1542 0.1552 0.1632)	{Rose, Cobb, Fisk, Mays } {Ott, Ruth}
2	(0.1754 0.1743 0.1739 0.1585 0.1579 0.1599)	{Rose, Cobb, Fisk} {Ott, Ruth, Mays }
3	(0.1742 0.1741 0.1724 0.1597 0.1592 0.1605)	{Rose, Cobb, Fisk} {Ott, Ruth, Mays}
4	(0.1732 0.1733 0.1716 0.1606 0.1601 0.1612)	{Rose, Cobb, Fisk} {Ott, Ruth, Mays }
5	(0.1724 0.1725 0.1710 0.1613 0.1609 0.1619)	{Rose, Cobb, Fisk} {Ott, Ruth, Mays }
6	(0.1717 0.1718 0.1704 0.1620 0.1616 0.1625)	{Rose, Cobb, Fisk} {Ott, Ruth, Mays}

The clusters change after the first two iterations, but then remain unchanged. The stochastic clustering algorithm allows the user to decide how many consecutive identical clusterings define a stopping condition. If that number is five, then the final clustering of {Rose, Cobb, Fisk} and {Ott, Ruth, Mays} is determined when $t = 6$. For the reader curious about whether the clustering changes at some later point, the algorithm was run through $t = 1000$, and the same clustering was found at each step.

CHAPTER 6

Some concerns

As is to be expected with a new algorithm, actual implementation of ideas that looked fine on paper can still be problematic. Even before implementation, there may be concerns about perceived weak links in the algorithm. In this chapter we will address some of these concerns. Since this chapter and Chapter 7 are tightly coupled, it will be hard to talk about these issues without highlighting some of the results to come. Hopefully, no great surprises are spoiled, and the turning of pages back and forth is kept to a minimum.

6.1 Impact of initial probability vectors

The fact that the stochastic clustering algorithm depends on a random initial probability vector (IPV) raises the question of whether all random probability vectors will lead to the same clustering. Since $\mathbf{P}$ is irreducible, we are guaranteed that the matrix has a

stationary distribution vector, regardless of the IPV. But for clustering purposes that is not the issue. Instead we would like to have confidence that for a certain IPV, $\mathbf{x}_t^T$ will remain in short-term equilibrium and middle run evolution long enough for us to identify the clusters. Secondly, as we will see soon in Chapter 7, different IPVs can lead to different cluster results.

We will consider the IPV question in two parts, first addressing the rare occurrence of an IPV that does not lead to a clustering at all, and then considering the fact that different IPVs can lead to different clusterings.

6.1.1 IVPs leading to no solution

Clearly not every initial probability vector will help us in data clustering. Suppose, for example, that

$$\mathbf{x}_0^T = \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n} \end{pmatrix}_{1\times n}.$$

Since $\mathbf{P}_{n\times n}$ is doubly stochastic, $\mathbf{x}_0^T$ is its stationary distribution vector. With such a choice for the IPV, $\mathbf{x}_t^T$ never changes and we have lost any ability to group the probabilities in $\mathbf{x}_t^T$ in order to cluster the original data.

A natural follow-up question would be whether small perturbations to the uniform probability vector also lead to values of $\mathbf{x}_t^T$ that make any clustering assignment impossible. For example, construct an IPV of the form

$$\mathbf{x}_0^T = \begin{pmatrix} \frac{1}{n}+\epsilon_1 & \frac{1}{n}+\epsilon_2 & \dots & \frac{1}{n}+\epsilon_n \end{pmatrix}$$

where $\sum_{i=1}^{n} \epsilon_i = 0$ and $0 \leq \frac{1}{n} + \epsilon_i \leq 1$ for $i = 1, \dots, n$.

Examining the first element of $\mathbf{x}_1^T$ after performing the iteration $\mathbf{x}_1^T = \mathbf{x}_0^T\mathbf{P}$ yields,

$$\begin{aligned}
\mathbf{x}_1^T(1) &= \left(\frac{1}{n} + \epsilon_1\right) p_{11} + \left(\frac{1}{n} + \epsilon_2\right) p_{21} + \cdots + \left(\frac{1}{n} + \epsilon_n\right) p_{n1} \\
&= \frac{1}{n}\left(p_{11} + p_{21} + \cdots + p_{n1}\right) + \epsilon_1 p_{11} + \epsilon_2 p_{21} + \cdots + \epsilon_n p_{n1} \\
&= \frac{1}{n} + \epsilon_1 p_{11} + \epsilon_2 p_{21} + \cdots + \epsilon_n p_{n1},
\end{aligned}$$

suggests that if the system

$$\begin{bmatrix} & & \\ & \mathbf{P} & \\ & & \\ 1 & \cdots & 1 \end{bmatrix}_{(n+1)\times n} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}_{n\times 1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{(n+1)\times 1} \qquad (\tilde{\mathbf{P}}\boldsymbol{\epsilon} = \mathbf{0}) \tag{6.1}$$

has a non-trivial solution, then $\mathbf{x}_1^T$ is the uniform probability vector and again any hope of finding clusters by looking at the entries of $\mathbf{x}_t^T$ is lost.

Note that any solution to (6.1) must be in the null space of $\mathbf{P}$, and if $\mathbf{P}$ is nonsingular the only solution is the trivial one. Also, recall the alternate formulation of the Simon-Ando evolution equation

$$\mathbf{x}_t^T = \mathbf{x}_0^T\mathbf{P}^t.$$

If $\mathbf{P}$ is nonsingular then so is any power of $\mathbf{P}$ meaning that any formulation of eqrefeq:linsys with $\mathbf{P}$ replaced by $\mathbf{P}^k$ will also lead to the zero solution.

We cannot limit our concern to initial probability vectors that go to the uniform distribution vector in just one iteration. If the uniform distribution vector is reached before the stochastic clustering algorithm has identified that $\mathbf{x}_t^T$ is in short-term equilibrium or middle-run evolution, a clustering will not be found. If $\mathbf{P}$ is singular, this possibility

exists, and an alert user may want to check the rank of $\mathbf{P}$ whenever the SCA returns unexpected results.

So the remote possibility exists that an IPV may not yield a solution. In the preparation of this thesis the stochastic clustering algorithm was run hundreds, if not thousands, of times and never was a failure due to a pathological IPV, but the thorough user should be aware of this issue nevertheless.

6.1.2 IPVs leading to different solutions

The fact that cluster analysis is an exploratory tool means that getting different solutions depending on the initial probability vector is not the end of the road, but rather an opportunity to examine these solutions in the hope of gaining additional insight into the data set's structure.

That said, it would still be instructive to know as much as possible about the characteristics shared by IPVs that lead to the same solution, how many different solutions are possible, and how often each of them is likely to appear. Probabilistic analysis of random starting vectors has been done in the context of iterative methods for finding eigenvalues and eigenvectors [20, 46], and is a natural area for further research on the stochastic clustering method.

6.2 Using a single similarity measure

The workload in consensus clustering is concentrated at the beginning of the process when the large number of clustering results are computed. Even if a user has access to a multiprocessor environment where this work can be shared, it would be advantageous to find a single similarity measure which is compatible with the stochastic clustering

algorithm.

Since the SCA is inspired by Simon-Ando theory, the underlying matrix must be nearly uncoupled. For a given data set, the problem with most traditional similarity (or dissimilarity) measures is that their values tend to the middle of their range. To illustrate, consider two common similarity measures: Euclidean distance and the cosine measure

$$f(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1^T \mathbf{x}_2}{||\mathbf{x}_1||_2 \, ||\mathbf{x}_2||_2}.$$

The former has the advantage of being familiar to almost everyone, while the latter has been found to be quite useful in data clustering, particularly text-mining [9]. However, as Figure 6.1 shows for the leukemia DNA microarray data set we will cluster in Section 7.2, the distribution of values returned by these two common measures is not the kind of distribution needed to form a nearly uncoupled matrix. There have been attempts to "massage" these kinds of distributions so that they contain more values at the extremes. Such methods often involve changing small values to zero and then performing some arithmetic operation that gives the remaining data a larger variance (for example, if the values are in the interval $[0, 1]$, squaring each value) [87]. These methods, however, are far from subtle and in experiments for use with the SCA, the matrix $\mathbf{P}$ went from dense to too sparse for clustering in one iteration of attempting to adjust its values.

A single measure that has been used with some success involves the idea of nearest neighbors, those data points closest to a given data point using a specific distance measure. For each element g in the data set, the set $\mathcal{N}_g$ consists of the κ nearest neighbors of g, where the user chooses both the positive integer κ and the distance measure used. The s_{ij} element of the consensus matrix is equal to the number of elements in $\mathcal{N}_i \cup \mathcal{N}_j$ [1].

Work with consensus matrices built in this fashion is still in its initial stages. It has

become obvious that the choice of κ and the distance measure greatly affect the results as can be seen in Table 6.1.

Table 6.1: Building a consensus matrix based on the number of shared nearest neighbors can work well or poorly depending on the value of κ, the number of nearest neighbors calculated for each data point. When $\kappa = 15$ the stochastic clustering algorithm detects five clusters. This fifth cluster only has one member, while the rest of the solution is correct.

κ	Clusters	Errors
15	5	1
20	4	0
25	4	18

6.3 Why use the stochastic clustering algorithm?

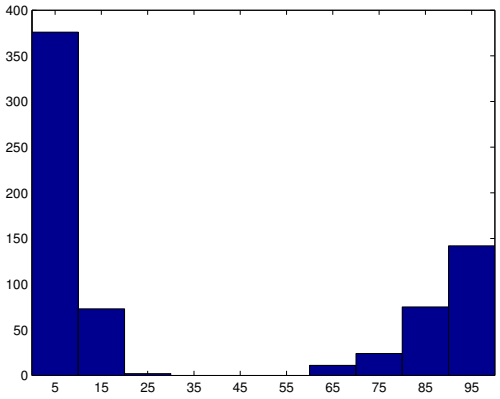
As we conclude our discussion of some of the concerns we have about the stochastic clustering algorithm and prepare to share some results on real-world data sets, it may be time to ask an obvious question: Why should we use this algorithm anyway?

The most compelling reason is that the SCA determines k, the number of clusters. If the clustering ensemble that creates the consensus matrix uses a variety of k-values, how does a user decide which to use for the final clustering of the consensus matrix? And if different clustering methods are used in the ensemble, how do we pick which one to use for the final clustering? And if the user decides to cluster the consensus matrix with a number of different algorithms and values of k, what do they do if they are again presented with a variety of solutions?

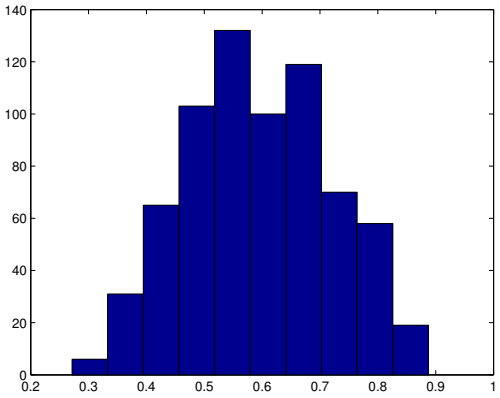
As we saw earlier in this chapter, the stochastic clustering algorithm can return different clusterings depending on the initial probability vector. The SCA has a natural solution to dealing with this problem, namely having the user build a new consensus matrix based on these different clusterings and use it as input for another run of the SCA. This repetitive approach is currently being tested.

Some might wonder why we do not use the consensus matrix $\mathbf{S}$ as input to some graph partitioning algorithm. A short reply to that would be that although there is much known about graph partitioning the work in the field is designed to partition graphs, not cluster data. We cannot assume that just because the structure of the consensus matrix resembles that of a weighted graph, that a partition of the graph is related to the underlying data clustering problem.

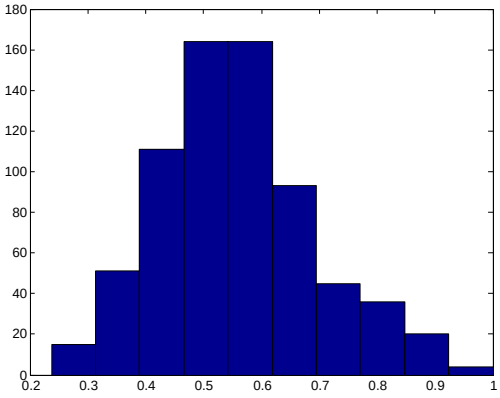
But what of spectral methods like the Fiedler method [23] which have been shown to be effective in clustering consensus matrices (see [6] for one example)? The Fiedler method relies on the Laplacian of the consensus matrix and makes clustering decisions based on sign pattern analysis of one or more the Laplacian's eigenvectors. The Fiedler method can be implemented either as a divide-and-conquer scheme, in which case a poor early division can never be undone, or as a method that examines multiple eigenvectors at once and is thus restricted to looking for a number of clusters equal to a power of two (though some of these may be empty).



(a) Distribution of consensus matrix entries.



(b) Distribution of cosine measure values.



(c) Distribution of Euclidean norm values.

Figure 6.1: Histogram 6.1a shows the distribution of consensus matrix similarity values between the 38 elements in the leukemia DNA microarray data set that will be introduced in Section 7.2. The horizontal axis measures the number of times out of 100 that two elements clustered together. Histogram 6.1b shows the distribution of cosine similarity measures between the same elements, while Histogram 6.1b does the same for Euclidean norm values scaled to the interval $[0, 1]$.

CHAPTER 7

Results

In this chapter we will look at results from using the stochastic clustering algorithm. The data sets will vary in number of elements and dimension of the data. Good and bad results will be shared with the hope that examining both will shed some light on the limitations of the algorithm and suggest areas for future work to improve the algorithm. In each section we also compare the stochastic clustering algorithm's results to those obtained by using a standard clustering algorithm to cluster the columns of the consensus similarity matrix $\mathbf{S}$.

7.1 The Ruspini data set

This thesis began with an "eyeball" clustering of the Ruspini data set, a seemingly innocuous collection of 75 points in the plane. We then saw that some widely-used

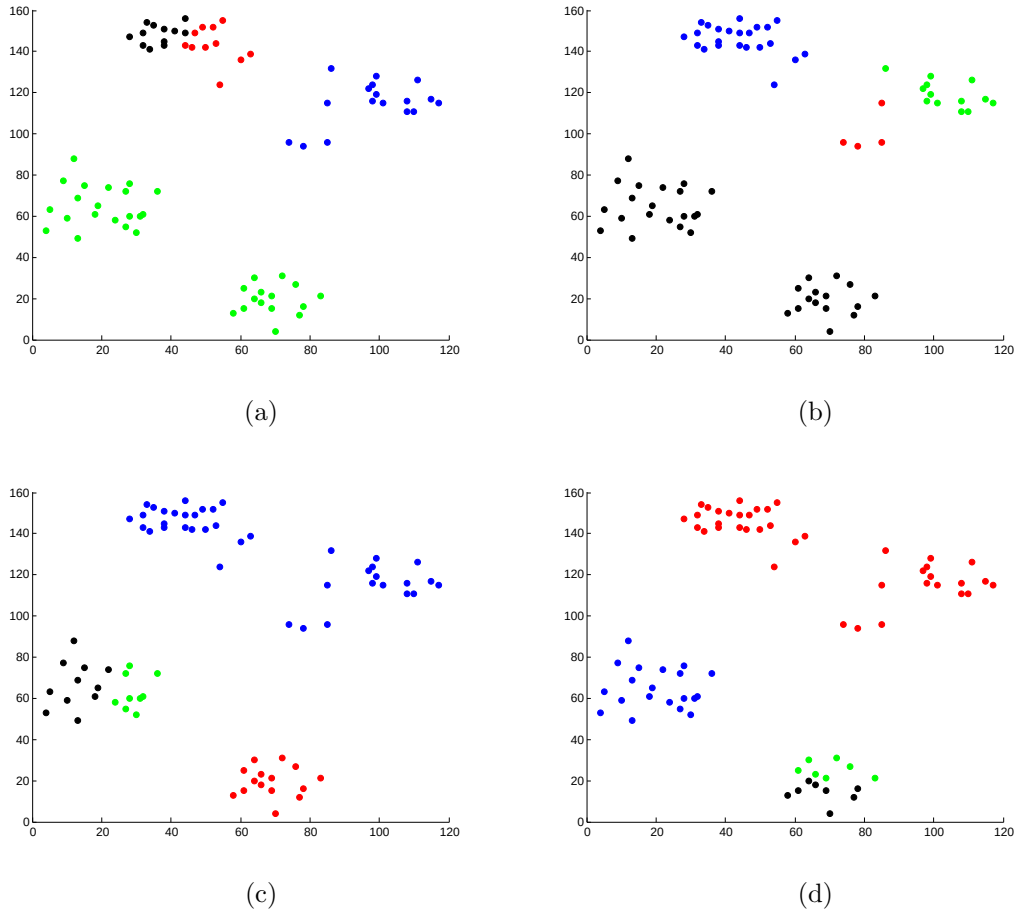


Figure 7.1: The k-means algorithm not only incorrectly clusters the Ruspini data set - but does it in a variety of ways.

clustering algorithms frequently make mistakes clustering this data set. Figure 7.1 shows four different misclusterings supplied by the k-means algorithm.

In a collection of 1000 clustering results found using MATLAB's k-means command, 464 did not match the natural "eyeball" clustering of Figure 1.1. In these instances k-means made between 15 and 26 assignment errors. As further evidence of k-means' unspectacular performance on this data set, in nine of the 1000 instances, k-means returned one empty cluster.

The stochastic clustering algorithm was run using the consensus similarity matrix **S** constructed using the results of these 1000 runs of k-means as input, and the result it returned was perfect.

To see how dependent the stochastic clustering algorithm is on the random initial probability vector, the SCA was run 1000 times with **S** as input. Table 7.1 summarizes the results and compares them to using the k-means algorithm to cluster the columns of **S**. The stochastic clustering algorithm performed far better. In fact, k-means did a better job clustering the original data than it did clustering the consensus matrix.

Table 7.1: The stochastic clustering algorithm performed much better than k-means at clustering the Ruspini data set using the consensus similarity matrix **S** as input.

	Out of 1000 trials	
# of Errors	SCA	k-means
0	924	279
1-10	0	0
11-20	76	333
21-30	0	285
31-40	0	60
41-50	0	42

7.2 DNA microarray data set

In 1999 a paper was published analyzing a DNA microarray data set containing the gene expression values for 6817 genes from 38 bone marrow samples [33]. Five years later, the same 38 samples were examined, though this time only 5000 genes were used [13]. The samples came from leukemia patients who had all been diagnosed with either acute

lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML). Additionally, the AML patients had either the B-cell or T-cell subtype of the disease (ALL-B or ALL-T). This data set is well known in the academic community (Google Scholar reports that the 1999 paper has been cited over 6000 times) and is an excellent test for new clustering algorithms since it can be divided into either two (ALL/AML) or three (ALL-B/ALL-T/AML) clusters. The actual clustering for the leukemia data set is known (see Table 7.2), though the 2004 paper noted that the data "contains two ALL samples that are consistently misclassified or classified with low confidence with most methods. There are a number of possible explanations for this, including incorrect diagnosis of the samples [13]."

Table 7.2: The correct clustering of the leukemia DNA microarray data set.

Diagnosis	Patients
ALL-B	1 – 19
ALL-T	20 – 27
AML	28 – 38

Since the 2004 paper was published to demonstrate the effectiveness of nonnegative matrix factorization in clustering this data set, though results for individual runs varied. So, this data set seems to be an appropriate test for the stochastic clustering algorithm, using NMF with different k values to build the ensemble. The data set was clustered using NMF 100 times each for $k = 2$ and $k = 3$. Additionally, to explore the data set further, the data were clustered an additional 100 times for $k = 4, 5$ and 6.

Table 7.2a shows the number of errors for each clustering used in building $\mathbf{S}_2$, the $k = 2$ consensus similarity matrix. NMF is clearly quite good at clustering this data set into two clusters. When the stochastic clustering algorithm is used to cluster the patients

based on $\mathbf{S}_2$, it mis-clusters Patients 6 and 29. This clustering is extremely reliable, with the same result from 100 consecutive calls of the SCA.

Similar comparisons were done using $\mathbf{S}_3$, the $k = 3$ consensus similarity matrix, and again the stochastic clustering method could not improve on the already excellent results of NMF. NMF made an average of 3.18 errors per clustering compared to 4.76 for the SCA. Even the hope that the SCA would provide a narrower band of errors than NMF is not realized (see Table 7.2b). Perhaps the lesson is that if the original method does a good job of clustering, SCA is not likely to improve on it.

Since cluster analysis is an exploratory tool, consensus matrices $\mathbf{S}_4$, $\mathbf{S}_5$, and $\mathbf{S}_6$ were constructed to see if either the stochastic clustering algorithm or nonnegative matrix factorization could discover some hidden structure in the data set that would indicate one or more undiscovered clusters. If a group of elements all break away from an existing cluster or clusters, there is reason for further investigation regarding a new cluster. Interestingly, when $k = 4$, the results from both NMF and the SCA agree. As Table 7.2c summarizes, they both have identified a fourth cluster made up of four ALL-B patients and two AML patients.

Neither of the methods give any indication of further clusters. When $k = 5$ or $k = 6$ both methods begin to build two or three large clusters with the remaining clusters containing only two or three members.

Before we move on to the next data set, there is one other interesting result to report. If the stochastic clustering algorithm is run using the sum of $\mathbf{S}_2$ and $\mathbf{S}_3$ it identifies two clusters and makes only one clustering mistake, namely Patient 29.[1]

[1]Throughout the research period for this thesis, the Patient 29 sample was misclustered nearly 100 per cent of the time. One of the authors of the 2004 paper verifies that in their work, the Patient 29 sample was also often placed in the wrong cluster [84].

# of Errors	1	2	3	4
# of Instances (NMF)	30	65	3	2
# of Instances (SCA)	0	100	0	0

(a) The leukemia DNA microarray data set was clustered 100 times using NMF with $k = 2$. The number of errors ranged between one and four. When the SCA was used on the consensus matrix created from those 100 NMF clusterings, it mis-clustered Patients 6 and 29 each time.

# of Errors	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
# of Instances (NMF)	0	71	3	9	3	3	1	2	0	3	0	1	0	3	1	0
# of Instances (SCA)	0	67	0	0	0	0	0	0	0	27	0	6	0	0	0	0

(b) Neither the SCA nor NMF shows an advantage over the other when clustering the consensus matrix $\mathbf{S}_3$.

Diagnosis	Patients	Patients
ALL-B	1 – 19	1, 3, 5, 7 – 9, 11 – 14, 16 – 18
ALL-T	20 – 27	10, 20 – 27
AML	28 – 38	28, 30 – 35, 37, 38
New Cluster		4, 6, 19, 29, 36

(c) Both NMF and SCA agree that there may be a new cluster. The third column shows the membership of this new cluster and the patients remaining in the other three.

Figure 7.2: A collection of tables that compare the results of clustering consensus matrices constructed using different k-values. The consensus matrices were clustered by both the SCA and NMF. Table 7.2a compares the results for $k = 2$. Table 7.2b shows very little difference between the two methods when $k = 3$. Table 7.2c shows a possible fourth cluster suggested by both NMF and SCA.

7.3 Presidential election data

The last two sections featured data sets with a known "correct" clustering. It is time to move to a more realistic situation where one might have some intution about the number of clusters and what elements should be in them, but not a definite answer in mind.

We will consider the state-by-state vote counts in each United States presidential election from 1980 to 2008 and cluster the states whose voting behaviors are similar [56]. The data are first stored in a 28×50 candidate-by-state matrix where entry ij is the number of votes candidate i received from state j. Note that a person who ran for president multiple times would have a row for each election. For example Ronald Reagan (1980) is in row one of this data set, while Ronald Reagan (1984) is in row four. The 28 rows represent the two major party candidates from these eight elections, the four well-known third party candidates during this period[2], and eight rows labeled "Other" that include votes cast for all other candidates.

Our knowledge of recent electoral history will tell us if the clustering results are unreasonable. The instances where the results do not fit our a priori opinion should be instructive in demonstrating the clustering method's ability to find hidden structure in the voting records.

The data were first clustered using the three algorithms

1. k-means after normalizing each column to row sum one and using squared Euclidean distance as the similarity measure,

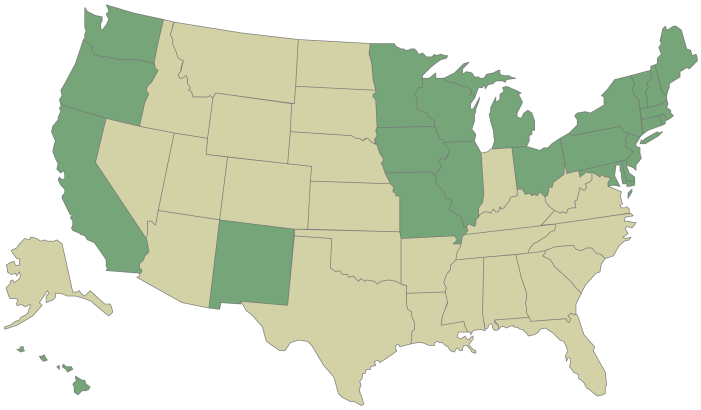
2. k-means using the cosine measure, and

3. nonnegative matrix factorization.

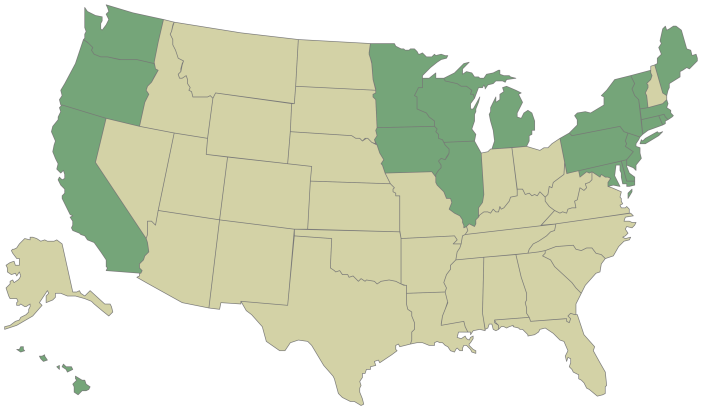
[2]John Anderson (1980), H. Ross Perot (1992, 1996), and Ralph Nader (2000)

Each of the algorithms was run 100 times for each of five k-values ($k = 2, 3, 4, 5, 6$) for a total of 15 consensus matrices.

Using the three $k = 2$ consensus matrices as input to the stochastic clustering algorithm, gives two different solutions as both k-means' inputs led to the same clustering. The maps in Figure 7.3 illustrate the results. The two maps agree on all but four states: Missouri, New Hampshire, New Mexico, and Ohio.



(a) SCA clustering using k-means input



(b) SCA clustering using NMF input

Figure 7.3: The SCA finds two clusters when working with consensus matrices built using $k = 2$, whether the original clustering was done by k-means (7.3a) or NMF (7.3b). There is agreement on all but four states.

Examining the maps it is clear that, in general, the green states are associated more with the Democratic party and the tan states more with the Republican party, and that the size of the "green" cluster is larger when using k-means consensus matrices. It is probably not a surprise that if the stochastic clustering algorithm is run on the sum of the two k-means created consensus matrices that the results are identical to those seen in Figure 7.3a. So even though combining the two k-means consensus matrices does not add to the number of green states, if all three consensus matrices are summed and the resulting matrix used as input to the stochastic clustering algorithm, four more states (Colorado, Florida, Nevada, and West Virginia) join the "green" cluster.

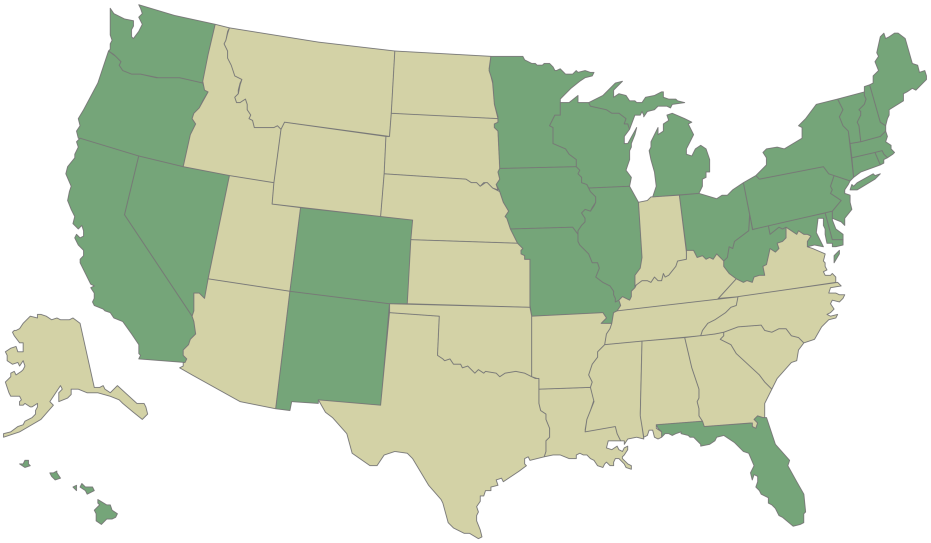
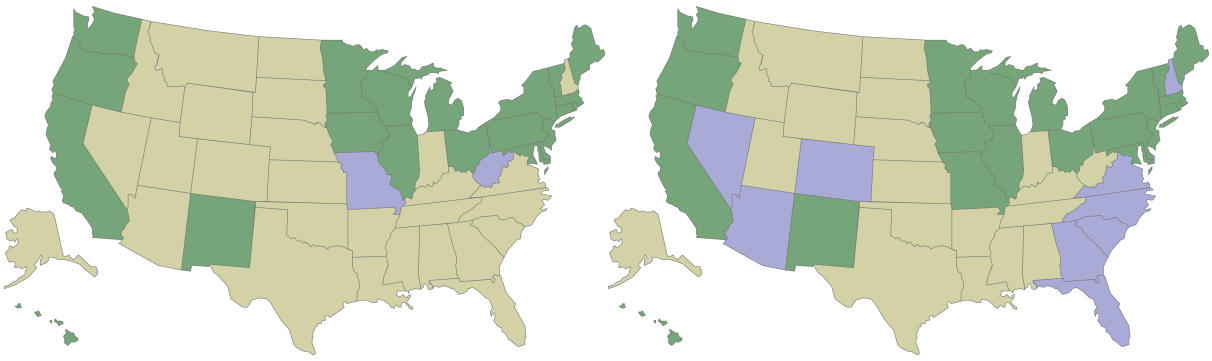


Figure 7.4: Results from the stochastic clustering of the consensus matrix formed by adding all three of the $k = 2$ consensus matrices together.

The results from using the stochastic clustering algorithm with the consensus matrices created when $k = 3$, bear out the common view of two separate groups of states in U. S. politics (i.e. the "red" states and the "blue" states). The third cluster is always small

(see Figure 7.5b) and often contains only one state, usually Missouri or West Virginia. When all three consensus matrices are summed, the SCA finds a third cluster with those two states as its only members (Figure 7.5a). This contrasts with work done examining presidential election data from 1912 to 2008, where a third cluster of eight to twelve states was consistently found [14, 62].

This discovery of only small third clusters is taken to its logical conclusion when the SCA attempts to cluster using the NMF-generated consensus matrix and returns only two clusters. This is because the gap between $\lambda_2(\mathbf{P})$ and $\lambda_3(\mathbf{P})$ is larger than that between $\lambda_3(\mathbf{P})$ and $\lambda_4(\mathbf{P})$. However, the difference in the gaps is only .0022. If the SCA is forced to find a third cluster, it also finds a small one - New Mexico and West Virginia.



(a) SCA clustering using all consensus matrices

(b) SCA clustering using k-means input

Figure 7.5: When the SCA works with consensus matrices built using $k = 3$ the result is often an empty or very small third cluster. The (7.5b) map shows one of the few sizable third clusters found.

As the stochastic clustering algorithm is applied to consensus matrices built with $k = 4$ or more we see the behavior exhibited at $k = 3$ continue. Either a small number of states occupy one or more of the clusters, or the SCA finds fewer clusters than the

original methods were asked to find. Consensus matrices built using NMF never yield more than three clusters, and when they are built with $k = 6$, the SCA finds just one cluster - all 50 states.

Still there is something to be learned by looking at the clusters found by the stochastic clustering algorithm as k gets larger. Figure 7.6 displays some representative maps and highlights the exploratory nature of cluster analysis. As the value of k grows, we can see small groups of states move into their own clusters. Subtleties that were hidden when $k = 2$ now begin to appear - a distinction between southeastern states that border the Atlantic Ocean and those inland, a difference between northern and southern states in the Mountain time zone, and New Hampshire's place as the state most physically isolated from its political brethren.

7.4 Custom clustering

As we first mentioned in Section 6.1, the fact that the stochastic clustering algorithm uses a random initial probability vector means that it can arrive at different solutions, and when clustering the leukemia and presidential election data sets we found this to be so. While this might be viewed as a weakness of the algorithm, it does give the researcher the ability to answer a very specific question by creating a specific initial probability vector.

In Section 7.2, we noticed that the SCA did not cluster the leukemia data set consensus matrix any better than nonnegative matrix factorization. But what if our primary interest was not in clustering the entire data set, but instead in finding the membership of the cluster of a particular data point. For example, if you are the physician for Patient number 2 you have limited interest in a global view of the leukemia data set. Indeed, rather than knowing which of the three clusters Patient 2 belonged to, it would be of greater use to

you to know a small number of other patients that are most like Patient 2 in the hope that that knowledge would help you tailor the best treatment plan possible.

To create such a custom clustering, we construct an IPV containing all zeros except for a 1 in the place corresponding to our data point of interest. We then ask the stochastic clustering algorithm to find the cluster containing our specific data point. Since we may be interested in a collection much smaller than that cluster, the stochastic clustering algorithm can be modified to ask for a small number data points whose $\mathbf{x}_t$ entries are closest to our target point.

Here again we find hope in a feature of the SCA that seemed to disappoint us in Section 7.2. In that section, the clustering of consensus matrices built from methods using $k = 5$ and $k = 6$ seemed to supply new information. In fact, the small clusters found then are indicative of an especially close relationship between the cluster members.

Incorporating these ideas using the consensus matrix $\mathbf{S}_6$ from Section 7.2 and an initial probability vector of all zeros except for a 1 in the second position gives us the custom cluster of $\{2, 4, 6, 15, 19, 29, 36\}$, a cluster with four other AML-B patients and two AML patients (although one of them, Patient 29, consistently clusters with the AML-B patients in our experience). These results are presented in table 7.3 along with the six nearest neighbors of Patient 2 using Euclidean distance and cosine measure. The SCA's custom cluster for Patient 2 features three patients not found in these nearest neighbor sets and suggests that physicians could learn a great deal by examining these hidden connections between Patient 2 and Patients 15, 29, and 36.

For another example of how we could use this custom clustering idea on data we examined in an earlier section, consider a young mathematician moving to the state of Pennsylvania and wondering about the political climate of that state. In Section 7.3 we created many consensus matrices based on votes cast in each state in the past eight

Table 7.3: Custom Cluster for leukemia Patient 2. This table shows the six other patients most similar to Patient 2. The patients are listed in similarity order, that is the first one is the one most similar to Patient 2. The cluster returned by the SCA differs by three patients with both lists derived from two traditional distance measures.

Method	Other Patients
SCA	29, 19, 4, 15, 36, 6
2-norm	19, 16, 9, 3, 6, 18
cosine	16, 19, 9, 3, 18, 4

presidential elections. Using a consensus matrix that sums the adjacency matrices from all three algorithms used when $k = 6$, the other members of Pennsylvania's custom cluster are

California,
Iowa,
Maine,
Michigan,
Minnesota,
New Jersey,
Oregon, and
Wisconsin.

For our final example of custom clustering we will attempt to make movie recommendations after clustering a data set of movies and ratings from the MovieLens recommender system, a research lab project in the Department of Computer Science and Engineering at the University of Minnesota [35].

The data set used contains over one million ratings for over 3900 movies made by 6040 users. Before clustering the data, movies that were rated fewer than 20 times were deleted leaving a total of 3043 movies with 995,492 ratings. The data were then stored in a 6040×3043 matrix that was clustered using nonnegative matrix factorization five times each for $k = 11, 12, \ldots, 30$, and the clustering results used to build a 3043×3043

consensus matrix $\mathbf{S}$.

This data set has not appeared yet in this chapter because the stochastic clustering method is not appropriate for clustering it. The largest gap in the list of ordered eigenvalues of $\mathbf{P}$, the doubly stochastic form of $\mathbf{S}$, is between $1 = \lambda_1(\mathbf{P})$ and $\lambda_2(\mathbf{P})$ which the SCA interprets to mean there is only one cluster. When the SCA is forced to look for a larger number of clusters, the typical result is a large number of small clusters and one or two clusters with hundreds or thousands of elements.

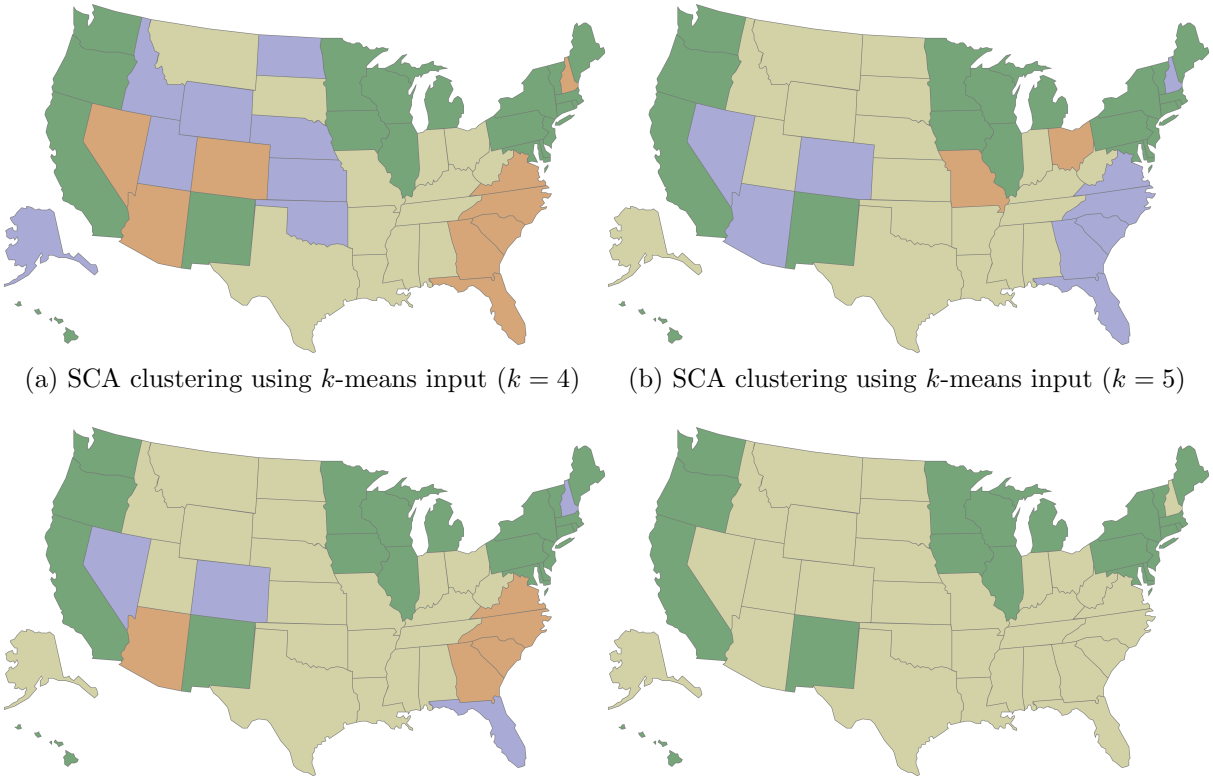
But if the SCA finds many small clusters in this data set, it may be quite useful to use it to find custom clusters for people wanting to see movies that have been rated similarly to their favorites. Table 7.4 shows the custom clustering algorithm's suggestions for three well-known movies. Remember that the algorithm is not saying anything about the similarity of these movies as entertainment. It is just saying that they are rated similarly by the same people. That message is especially important to consider when reading Table 7.5.

Table 7.4: Some Custom Cluster Movie Recommendations

Rebecca	**Die Hard**	**Forrest Gump**
Vertigo	Forbidden Planet	I.Q.
Rear Window	Contact	Dave
Sunset Boulevard	X-Files	That Thing You Do!
Laura	Nineteen Eighty-Four	My Best Friend's Wedding
The Thin Man	2010	The Wedding Singer
The Big Sleep	Star Trek: Insurrection	You've Got Mail
Strangers on a Train	Planet of the Apes	
Shadow of a Doubt	Predator	
The Lady Eve		
The Palm Beach Story		
Double Indemnity		

Table 7.5: Huh?

The Princess Bride
Ghost in the Shell (Kokaku kidotai)
The Hunt for Red October
Lethal Weapon
Conquest of the Planet of the Apes
Robocop 2
Quatermass and the Pit



(a) SCA clustering using k-means input ($k = 4$)

(b) SCA clustering using k-means input ($k = 5$)

(c) SCA clustering input from all $k = 4$ methods

(d) SCA clustering using all 15 consensus matrices.

Figure 7.6: These maps highlight some of the results from applying the SCA to consensus matrices built with k-values of four or greater. Maps 7.6a and 7.6b both come from SCA clustering of k-means created consensus matrices, one using $k = 4$ and the other using $k = 5$. The SCA finds four clusters using both of these inputs, though the clusters are different. Map 7.6c shows the SCA output from using all of the $k = 4$ created consensus matrices. And finally, for 7.6d the SCA was applied to the sum of all 15 consensus matrices. It determined that $k = 2$ and results in a map quite similar to the ones we first saw in Figure 7.3.

Custom Clustering Algorithm (CCA)

1. Create the consensus similarity matrix $\mathbf{S}$ and the doubly stochastic symmetric matrix $\mathbf{P}$ just as in the stochastic clustering algorithm.

2. Construct $\mathbf{x}_0^T$ to contain all zeros except for a one in the place of the element we are interested in creating a custom cluster for.

3. Pass the algorithm values for the minimum and maximum size cluster you desire and the maximum number of iterations the CCA should take trying to find that cluster.

4. After each $\mathbf{x}_t^T = \mathbf{x}_t^T\mathbf{P}$ multiplication, cluster the elements of $\mathbf{x}_t^T$ as in the SCA. If the cluster containing the target element is within the size parameters, output the cluster and end the program.

Figure 7.7: The Custom Clustering Algorithm

CHAPTER 8

Conclusion

8.1 Contributions

- The development and analysis of a new consensus clustering algorithm that does not require the user to decide on the number of clusters. This is significant because nearly all popular techniques require the user to either have a priori knowledge of the number of clusters or to guess at it.
- A new measure, $\sigma(\mathbf{P})$, is introduced that quantifies how nearly uncoupled a matrix is. Unlike earlier measures, $\sigma(\mathbf{P})$ can be applied to both stochastic and non-stochastic matrices.
- A rigorous proof that when the consensus matrix $\mathbf{S}$ is converted to doubly stochastic form near uncoupledness is not lost.

- A rigorous proof that if the second eigenvalue of an irreducible, symmetric, doubly stochastic matrix is close to one, then the matrix has nearly uncoupled form.

- Empirical "proof of concept" results that demonstrate the viability of the new clustering technique. These results have been shared with the research community through conference proceedings and research papers [62, 63].

8.2 Future Research

- Use probabilistic analysis of initial probability vectors to see what we can learn about the number of possible solutions the SCA can return and whether there is any connection between $\sigma(\mathbf{P}, n_1)$ and the tendency of $\mathbf{P}$ to produce multiple solutions.

- Investigate whether in situations where the stochastic clustering algorithm returns multiple answers, if building a consensus matrix from these results, and applying the SCA again will eventually yield a unique solution.

- Examine whether the Sinkhorn-Knopp balancing step can be replaced by a simple scaling to make all row sums equal. Though we lose the results from Markov chain theory, perhaps they are unneeded since all we are looking for is $\mathbf{x}_t^T$ values that are approximately equal.

- Continue the search for a single similarity measure whose values are distributed in a way that can be exploited by the stochastic clustering method.

- Improve the bounds for values of d_i. Numerical results indicate that the upper bound found for Theorem 3.13 can be greatly improved.

- Explore the structure of the spectrum of symmetric, irreducible, nearly uncoupled, doubly stochastic matrices. For this thesis, we were only concerned with the eigenvalues near one, but from examining eigenvalues during the course of this research, there appears to be some structure to the spectrum, especially a large number of eigenvalues near zero.

- Work to find some bounds on the numeric connection between $\lambda_2(\mathbf{P})$ and $\sigma(\mathbf{P}, n_1)$ that Theorems 4.4 and 4.5 establish.

- Establish a precise definition for the Perron cluster and use it to rigorously extend Theorem 4.5 to all the eigenvalues in the Perron cluster.

REFERENCES

[1] R. Abbey. Personal communication, April 28, 2011.

[2] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 51(5):23:1–23:27, 2008.

[3] M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press, 1973.

[4] R. G. Bartle. *The Elements of Real Analysis*. Wiley, New York, 1964.

[5] Baseball-Reference.com. http://www.baseball-reference.com, 2011.

[6] D. Benson-Putins, M. Bonfardin, M. E. Magnoni, and D. Martin. Spectral clustering and visualization: A novel clustering of Fisher's iris data set. *SIAM Undergraduate Research Online*, 4, 2011.

[7] P. Berkhin. Survey of clustering data mining techniques. *Grouping Multidimensional Data: Recent Advances in Clustering*, pages 25–71, 2006.

[8] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, New York, 1979.

[9] M. Berry and M. Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval, Second Edition*. SIAM, 2005.

[10] A. Borobia and R. Cantó. Matrix scaling: A geometric proof of Sinkhorn's theorem. *Linear algebra and its applications*, 268:1–8, 1998.

[11] J. B. Brown, P. J. Chase, and A. O. Pittenger. Order independence and factor convergence in iterative scaling. *Linear algebra and its applications*, 190:1–38, 1993.

[12] R. A. Brualdi, S. V. Parter, and H. Schneider. The diagonal equivalence of a nonnegative matrix to a stochastic matrix. *Journal of Mathematical Analysis and Applications*, 26:31–50, 1966.

[13] J.-P. Brunet, P. Tamayo, T. R. Golub, J. P. Mesirov, and E. S. Lander. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences of the United States of America*, 101(12):4164 – 4169, 2004.

[14] T. Chartier and C. Wessell. A nonnegative analysis of politics. *Math Horizons*, 18(4):10–13, 2011.

[15] F. E. Clements. Use of cluster analysis with anthropological data. *American Anthropologist, New Series*, 56:180 – 199, 1954.

[16] P. J. Courtois. *Decomposability: Queueing and Computer System Applications*. Academic Press, 1977.

[17] J. Csima and B. N. Datta. The DAD theorem for symmetric non-negative matrices. *Journal of Combinatorial Theory (A)*, 12:147–152, 1972.

[18] P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra and its Applications*, 315(1-3):39–59, 2000.

[19] P. Deuflhard and M. Weber. Robust Perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398:161–184, 2005.

[20] J. D. Dixon. Estimating extremal eigenvalues and condition numbers of matrices. *SIAM Journal on Numerical Analysis*, 20(4):812–814, 1983.

[21] R. Dubes and A. K. Jain. Clustering techniques: the user's dilemma. *Pattern Recognition*, 8(4):247–260, 1976.

[22] B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. Arnold, fourth edition, 2001.

[23] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. *Czechoslovak Math. J*, 25(100):619–633, 1975.

[24] V. Filkov and S. Skiena. Heterogeneous data integration with the consensus clustering formalism. In *Proceedings of Data Integration in the Life Sciences*, 2004.

[25] V. Filkov and S. Skiena. Integrating microarray data by consensus clustering. *International Journal on Artificial Intelligence Tools*, 13(4):863 – 880, 2004.

[26] A. Fred and A. K. Jain. Data clustering using evidence accumulation. In *Proceedings of the 16th International Conference on Pattern Recognition*, Quebec, Canada, 2002.

[27] A. L. N. Fred and A. K. Jain. Robust data clustering. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Wisconsin, 2003.

[28] D. Fritzsche, V. Mehrmann, D. B. Szyld, and E. Virnik. An SVD approach to identifying metastable states of Markov chains. *Electronic Transactions on Numerical Analysis*, 29:46–69, 2008.

[29] G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms, and Applications*. ASA-SIAM Series on Statistics and Applied Probability. ASA and SIAM, 2007.

[30] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Trans. Knowl. Discov. Data*, 1, March 2007.

[31] A. Goder and V. Filkov. Consensus clustering algorithms: Comparison and refinement. In *ALENEX*, pages 109–117, 2008.

[32] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 1983.

[33] T. R. Golub et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531 – 537, 1999.

[34] T. Grotkjaer, O. Winther, B. Regenberg, J. Nielsen, and L. K. Hansen. Robust multi-scale clustering of large DNA microarray datasets with the consensus algorithm. *Bioinformatics*, 22(1):58 – 67, 2006.

[35] GroupLens Research. http://www.grouplens.org/node/73, 2011.

[36] D. J. Hartfiel. Proof of the Simon-Ando theorem. *Proceedings of the American Mathematical Society*, 124:67 – 74, 1996.

[37] J. A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, 1975.

[38] P. Hore, L. O. Hall, and D. B. Goldgof. A scalable framework for cluster ensembles. *Pattern recognition*, 42(5):676–688, 2009.

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

[40] H. B. Isaacson, E. andKeller. *Analysis of Numerical Methods*. J. Wiley & Sons, New York, 1966.

[41] M. N. Jacobi. A robust spectral method for finding lumpings and meta stable states of non-reversible Markov chains. *Electronic Transactions on Numerical Analysis*, 37:296 – 306, 2010.

[42] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[43] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

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

[45] A. K. Jain, A. Topchy, M. H. C. Law, and J. M. Buhmann. Landscape of clustering algorithms. *Pattern Recognition*, 1:260–263, 2004.

[46] E. R. Jessup and I. C. F. Ipsen. Improving the accuracy of inverse iteration. *SIAM J. Sci. Stat. Comput.*, 13(2):550–572, 1992.

[47] C. R. Johnson and R. Reams. Scaling of symmetric matrices by positive diagonal congruence. *Linear and Multilinear Algebra*, 57(2):123–140, 2009.

[48] T. Kato. *A Short Introduction to Perturbation Theory for Linear Operators*. Springer-Verlag, 1982.

[49] P. A. Knight. The Sinkhorn-Knopp algorithm: Convergence and applications. *SIAM Journal of Matrix Analysis and Applications*, 30:261 – 275, 2008.

[50] J. Kogan. *Introduction to Clustering Large and High-Dimensional Data*. Cambridge University Press, 2007.

[51] J. Kruithof. Telefoonverkeersrekening. *De Ingenieur*, 52:E15–E25, 1937.

[52] B. Lamond and N. F. Stewart. Bregman's balancing method. *Transportation Research Part B: Methodological*, 15(4):239–248, 1981.

[53] A. N. Langville and C. D. Meyer. Updating Markov chains with an eye on Google's PageRank. *SIAM journal on matrix analysis and applications*, 27(4):968–987, 2006.

[54] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788 – 791, 1999.

[55] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *In NIPS*, pages 556–562. MIT Press, 2001.

[56] David Leip. Dave Leip's Atlas of U.S. Presidential Elections, 2010. http://www.uselectionatlas.org.

[57] D. Leitmann. On one approach to the control of uncertain systems. *Journal of Dynamic Systems, Measurement, and Control*, 115:373 – 380, 1993.

[58] S. A. Levin. The problem of pattern and scale in ecology: the Robert H. MacArthur award lecture. *Ecology*, 73(6):1943–1967, 1992.

[59] C. D. Meyer. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Review*, 31(2):240 – 272, 1989.

[60] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[61] C. D. Meyer. Matrix analysis. Chapter in upcoming book, 2011.

[62] C. D. Meyer and C. D. Wessell. Stochastic consensus clustering. In *Proceedings of the Sixth International Workshop on the Numerical Solutions of Markov Chains*, 2010.

[63] C. D. Meyer and C. D. Wessell. Stochastic data clustering. *in revision, preprint at arXiv:1008.1758*, 2010.

[64] H. Minc. *Nonegative Matrices*. John Wiley & Sons, New York, 1988.

[65] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52:91–118, 2003.

[66] J.M. Ortega. *Numerical analysis: A second course*. Academic Press, New York, 1972.

[67] S. Race. Data clustering via dimension reduction and algorithm aggregation. Master's thesis, North Carolina State University, 2008.

[68] U. G. Rothblum, H. Schneider, and M. H. Schneider. Scaling matrices to prescribed row and column maxima. *SIAM Journal on Matrix Analysis and Applications*, 15(1):1–14, 1994.

[69] E. H. Ruspini. Numerical methods for fuzzy clustering. *Information Science*, 2:319 – 350, 1970.

[70] N. A. Salingaros. Complexity and urban coherence. *Journal of Urban Design*, 5(3):291–316, 2000.

[71] M. H. Schneider and S. A. Zenios. A comparative study of algorithms for matrix balancing. *Operations Research*, 38(3):439–455, 1990.

[72] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer-Verlag, second edition, 1981.

[73] H. A. Simon. Near decomposability and the speed of evolution. *Industrial and Corporate Change*, 11(3):587, 2002.

[74] H. A. Simon and A. Ando. Aggregation of variables in dynamic systems. *Econometrica*, 29(2):111–138, 1961.

[75] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35(2):pp. 876–879, 1964.

[76] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[77] D. Skillicorn. *Understanding Complex Datasets: Data Mining with Matrix Decomposition*. Chapman and Hall, 2007.

[78] G. W. Soules. The rate of convergence of Sinkhorn balancing. *Linear Algebra and its Applications*, 150:3–40, 1991.

[79] H. Spath. *Cluster Analysis Algorithms for Data Reduction and Classification of Objects*. Ellis Horwood Limited, 1980.

[80] O. Sporns, G. Tononi, and G. M. Edelman. Connectivity and complexity: the relationship between neuroanatomy and brain dynamics. *Neural Networks*, 13(8-9):909–922, 2000.

[81] W. J. Stewart. *An Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[82] A. Strehl. *Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining*. PhD thesis, University of Texas at Austin, 2002.

[83] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[84] P. Tamayo. Personal communication, May 9, 2011.

[85] R. M. Tifenbach. On an SVD-based algorithm for identifying meta-stable states of Markov chains. *Electronic Transactions on Numerical Analysis*, 38:17 – 33, 2011.

[86] R. C. Tryon. *Cluster Analysis*. McGraw-Hill, New York, 1939.

[87] S. van Dongen. *Graph clustering by flow simulation*. PhD thesis, University of Utrecht, 2000.

APPENDIX

APPENDIX A

MATLAB code

A.1 Stochastic Clustering Algorithm

```
function [C t k x] = sca(S,ds,num,maxevals,ipv,repclusters,maxits)
%
% sca - Stochastic Clustering Algorithm
%
% INPUT
% S - an n x n similarity matrix
% ds - A flag telling whether S has already been coverted to doubly
%      stochastic form. 0=no, 1=yes. If the user is using the same
%      consensus matrix repeatedly, it is advantageous to use the
%      Sinkhorn-Knopp algorithm just once outside of this program.
% num - the number of clusters. The user can specify a specific number.
%       num=0 lets the sca choose the number of clusters by examining the
%       Perron cluster.
% maxevals - the number of largest eigenvalues of P that are computed. For
%            a large problem the user might want to request less than all
```

```
%            of the eigenvalues to improve performance. maxevals=0 computes
%            all the eigenvalues.
% ipv - a user specified initial probability vector. ipv=0 means a random
%       ipv will be used.
% repclusters - the sca will stop when the same clustering is found for
%               this number of consecutive iterations.
% maxits - the maximum number of iterations the user wants the sca to run.
%
% OUTPUT
% C - an index with the cluster number for each data point.
% t - the number of iterations the sca ran.
% k - the number of clusters found.
% x - all the probability vectors used during the run of the sca.
%     x(1,:) is the ipv. This can use a lot of memory, so don't request it
%     unless you have a need to study each probability vector.
%
%  Author: Chuck Wessell
%  Last Updated: May 15, 2011
%
[m,n] = size(S);
if m ~= n
    fprintf('\nWarning: Input matrix S is not square.\n');
    return
end
%
% If S isn't already doubly stochastic, apply the Sinkhorn-Knopp algorithm
% for symmetric, fully indecomposable matrices. Then the matrix P is
% checked for symmetry.
%
if ds == 0
    [P] = skd(S);
else
    P = S;
end
%
issym = max(max(abs(P-P')))<=1e-16;
if ~issym
    fprintf('\nWarning: Matrix P is not symmetric.\n');
    return
end
%
```

```
% Look at the eigenvalues of P. Although all the eigenvalues should be
% real, MATLAB's eig command occasionally has a +/- .0001i tacked onto an
% eigenvalue. The workaround is to sort only the real parts of the
% eigenvalues.
%
if maxevals == 0
    lambdas = sort(real(eig(P)),'descend');
else
    lambdas = sort(real(eigs(P,maxevals)),'descend');
end
%
% If the parameter num is non-zero, the user wishes to choose the number of
% clusters. If num = 0 the user will let the algorithm decide on the number
% of clusters. To do this, the eigenvalues of P are sorted and the largest
% gap in the sorted list found. This gap divides the eigenvalues into two
% groups. The number of eigenvalues in the group containing 1 is the number
% of clusters.
%
if num == 0
    k = 1;
    maxgap = 0;
    for i=2:length(lambdas)
        if abs(lambdas(i)-lambdas(i-1)) > maxgap
            maxgap = abs(lambdas(i)-lambdas(i-1));
            k = i-1;
        end
    end
else
    k = num;
end
%
% The storage for all the probability vectors and the clustering index is
% created.
%
x = zeros(10,n);
C=ones(n,1);
%
% The clustering loop is entered, and the ipv is either randomly generated
% or set to the user supplied one. Then until a clustering is found or
% maxits reached, the inner while loop calculates a new probability vector,
% find the k-1 largest gaps in it and clusters the probabilities on each
```

```
% side of the gaps. If the clusterings agree for repclusters consecutive
% iterations, clustering is complete.
%
clustered = 0;
while ~clustered
    if ipv == 0
        x(1,:) = rand(1,n);
        x(1,:) = x(1,:)/sum(x(1,:));
    else
        x(1,:) = ipv;
    end
    t = 1;
    done = 0;
    count = 0;
    while ~done && t <= maxits
       x(t+1,:) = x(t,:)*P;
       oldC = C;
       C=ones(n,1);
       [b,idx] = sort(x(t+1,:),'ascend');
       for i=1:length(b)-1
           delta(i)=b(i+1)-b(i);
       end
       [bd,idxd] = sort(delta,'descend');
       for i=2:k
           match = C(idxd(i-1)+1);
           target = idxd(i-1)+1;
           while (target <= n) && (C(target) == match)
               C(target)=i;
               target = target + 1;
           end
       end
       Chat=sortrows([idx' C],1);
       C=Chat(:,2);
       if oldC == C
           count = count + 1;
           if count == repclusters
               done = 1;
           end
       end
       t = t+1;
    end
```

```
    clustered = 1;
end
```

A.2 Matrix Scaling

```
function [P,d] = skd(A)
%
% skd - Sinkhorn-Knopp algorithm for matrices that can be made doubly
% stochastic by left and right multiplication by the same diagonal matrix.
% Function skd converts square matrix A to doubly stochastic from by
% forming the product DAD, where D is a diagonal matrix with positive main
% diagonal entries. The function computes the two diagonal matrices
% typically found by the Sinkhorn-Knopp algorithm and then scales one of
% them to arrive at D.
%
% INPUT
% A - an n x n similarity matrix. In order for this single D approach to be
% appropriate, the input matrix A must be symmetric, fully indecomposable,
% nonnegative and have a positive main diagonal. This will always be the
% case if this function is used in conjunction with the Stochastic
% Clustering Algorithm, but some error checking is done anyway.
%
% OUTPUT
% P - the n x n doubly stochastic matrix converted from A.
% d - the diagonal elements of the matrix D.
%
% Author: Chuck Wessell
% Last Updated: May 15, 2011
%
[m,n]=size(A);
%
if m ~= n
    error('Matrix must be square.');
    return;
end
%
if min(A(:)) < 0
    error('Matrix must be nonnegative.');
    return;
end
%
if sum(diag(A) > 0) < m
    error('Matrix must have a positive diagonal.');
    return;
```

```
end
%
% It is possible to compute the diagonal of D using the single MATLAB
% command x = 1./(A*x); instead of the two statements labeled (1) and (2)
% below. However, alternate iterations of x converge to the same limits
% that c and r do, so this code uses the more intuitive c and r. The loop
% is continued until consecutive iterates of both c and r differ by less
% than 1e-12.
%
iter = 1;
diff = 1; %to force first iteration
tol = 1.0e-12;
c = ones(m,1);
r = ones(m,1);
while (diff > tol)
    oldc = c;
    oldr = r;
    c = 1./(A'*r);  % (1)
    r = 1./(A*c);   % (2)
    iter = iter + 1;
    diff = max(norm(c-oldc,2),norm(r-oldr,2));
end
%
alpha = c(1)/r(1);
d = sqrt(alpha) * r;
P = diag(d) * A * diag(d);
%
end
```

A.3 Custom clustering algorithm

```
function customcluster(S,ds,ipv,maxits,lb,ub,names)
%
% customcluster - attempts to find a cluster within a given size range that
% includes a given element of the data set. customcluster uses the
% Stochastic Clustering Algorithm to cluster the data.
%
% INPUT
% S - an n x n similarity matrix
% ds - A flag telling whether S has already been coverted to doubly
%      stochastic form. 0=no, 1=yes. If the user is using the same
%      consensus matrix repeatedly, it is advantageous to use the
%      Sinkhorn-Knopp algorithm just once outside of this program.
% ipv - a user specified 1 x n initial probability vector containing all
%       zeros except for a 1 in place representing the data point the user
%       wishes to cluster around.
% maxits - the maximum number of iterations the user wants customcluster to
%          search.
% lb - a user-defined lower bound on the size of the cluster (includes the
%      target data point and lb-1 others).
% ub - a user-defined upper bound on the size of the cluster (includes the
%      target data point and ub-1 others).
% names - a list of the names associated with the data points. An array of
%         integers can be used if the data have no names.
%
% OUTPUT
% The cluster elements are printed to the screen.
%
%  Author: Chuck Wessell
%  Last Updated: May 15, 2011
%
[m,n] = size(S);
if m ~= n
    fprintf('\nWarning: Input matrix S is not square.\n');
    return
end
%
% If S isn't already doubly stochastic, apply the Sinkhorn-Knopp algorithm
% for symmetric, fully indecomposable matrices. Then the matrix P is
% checked for symmetry.
```

```
%
if ds == 0
    [P] = skd(S);
else
    P = S;
end
issym = max(max(abs(P-P')))<=1e-16;
if ~issym
    fprintf('\nWarning: Matrix P is not symmetric.\n');
    return
end
%
% The number of clusters is rather arbitratily set to the square root of
% the number of data points. The item of interest (ioi) is identified from
% the ipv and the cluster index C initialized. Then until maxits is reached
% or a suitably sized cluster is found, x=x*P is repeated and the resulting
% vector clustered. Once an appropriate sized cluster is found, the names
% of the cluster members are displayed on the screen.
%
k=floor(sqrt(m));
ioi = find(ipv==1);
x = ipv;
C=ones(n,1);
%
for reps=1:maxits
    x = x*P;
    C = ones(n,1);
    [b,idx] = sort(x,'ascend');
    for i=1:length(b)-1
        delta(i)=b(i+1)-b(i);
    end
    [bd,idxd] = sort(delta,'descend');
    for i=2:k
        match = C(idxd(i-1)+1);
        target = idxd(i-1)+1;
        while (target <= n) && (C(target) == match)
            C(target)=i;
            target = target + 1;
        end
     end
     Chat=sortrows([idx' C],1);
```

```
    C=Chat(:,2);
    if (lb <= length(find(C==C(ioi)))) &&  (length(find(C==C(ioi))) <= ub)
        names(C==C(ioi),:)
        return
    end
end
```