UNIVERSITY OF CALIFORNIA
RIVERSIDE

Automatic Co-Clustering for Social Network and Medical Data

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Juan Ignacio Casse

December 2014

Dissertation Committee:

Dr. Christian Shelton, Chairperson
Dr. Robert Hanneman
Dr. Eamonn Keogh
Dr. Stefano Lonardi

The Dissertation of Juan Ignacio Casse is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am deeply grateful to my adviser, Dr. Christian R. Shelton, for his invaluable mentoring. Without his help I would not have been able to complete this dissertation.

I thank my committee members: Dr. Lonardi for his technical advise, Dr. Hanneman for his time and advice with social network analysis, and Dr. Keogh for his generous support during Fall quarter 2011.

I also like to thank my lab mates, present and past: Teddy Yap, Yu Fan, Antony Lam, Jing Xu, Joon Lee, William Lam, Kevin Horan, Busra Celikkaya, Zhen Qin, Dave Gomboc, Mike Izbicki, Matthew Zarachoff, Kazi Islam and Sepideh Azarnoosh. They made life in the lab more bearable with stimulating conversations and the occasional exploring of local restaurants.

Chapter 2 is, in part, a reprint of the material as it appears in A New Criterion Function for Exploratory Blockmodeling for Structural and Regular Equivalence, published in Social Networks 35(1), 31–50, 2013. The co-author Dr. Christian Shelton listed in that publication directed and supervised the research which forms the basis for this dissertation. The co-author Dr. Robert Hanneman, also listed in the publication, provided technical expertise in social theories and helped analyze the results.

To my parents.

ABSTRACT OF THE DISSERTATION

Automatic Co-Clustering for Social Network and Medical Data

by

Juan Ignacio Casse

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, December 2014
Dr. Christian Shelton, Chairperson

The task of *clustering* is a fundamental task in many important human endeavors. In machine learning parlance, it is an unsupervised learning tool for discovering patterns in data. Specifically, its goal is to find groups of objects in the data that are similar in some sense. Some important fields where clustering is used include medical diagnostics, bioinformatics, social network analysis and market analysis. Clustering is also used "behind the scenes" as a preprocessing step to other tasks, such as Web search and recommender systems.

Co-clustering can be viewed as a generalization of clustering to a wider range of data. While clustering methods work on affinity data (data describing similarity between objects), co-clustering methods can also work on relational data (data describing relationships between objects). An example of affinity data is customers in market analysis, where each customer is described by a set of features (attributes), such as age, gender and income. A similarity measure between pairs of customers can be computed from their features, for example Euclidean distance. An example of relational data is persons in a social network,

where a link between two persons indicate that they are friends. Here persons are compared on their connections to other persons and not on their intrinsic features.

In this dissertation we study the application of co-clustering to social network data and to medical data. In particular, we present a general formulation of co-clustering that fits most methods in the literature and provide solutions to three main problems: (1) clustering relational data under regular equivalence in social network analysis, (2) finding a symmetric clustering of asymmetric data and (3) clustering patients based on high-dimensional, time-varying, sparse physiologic data.

We define implicit similarity measures, by way of criterion functions for co-clustering, that solve the problems we target. We demonstrate and compare our co-clustering methods on real world data sets.

Contents

List of Figures

Chapter 1

Introduction

Finding regularities in data is a universally fundamental and important task in most human endeavors, from the learning of one's mother's voice when we are born to the advancement of science. As an example of its impact to humanity as a whole, consider Johannes Kepler's discovery of the empirical laws of planetary motion, which allowed the development of classical mechanics. These discoveries were possible by the astronomical observations of Tycho Brahe in the 16th century.

In machine learning, clustering is the unsupervised learning task of discovering patterns (or regularities) by finding groups of examples in the data that are similar in some predetermined sense. Clustering can also be viewed as assigning labels to data points, where the labels represent the groups (or classes) to which the data points belong. Clustering is used in a wide variety of problems in many fields. In computer vision, image segmentation is the task of partitioning a digital image into segments by assigning a label to each pixel such that pixels with the same label are spatially close to each other and have the same

color (Shi and Malik, 1997). Image segmentation is typically used to identify objects in a scene, which can have many applications in may fields, from face recognition to medical imaging. In information retrieval, clustering is used to make searches more efficient by scanning only through clusters of similar documents instead of the entire database (Cutting et al., 1992). (In this case, clustering is used as a preprocessing step and not as the end goal.) In market analysis, clustering is used to find groups of customers with similar purchasing behaviors (Punj and Stewart, 1983). In sociology, clustering is used to discover social positions (Lorrain and White, 1971). More recently, clustering of patients based on electronic health records (EHRs) is being studied for the purpose of providing more accurate and timely prognoses and effective treatment (Marlin et al., 2012).

Co-clustering can be viewed as a generalization of clustering to a wider range of data. While clustering methods work on affinity data (data describing similarity between objects), co-clustering methods can also work on relational data (data describing a relationship between objects). An example of affinity data is a data set of customers in market analysis, where each customer is described by a set of features (attributes), such as age, gender and income. A similarity measure between pairs of customers can be designed based on their features. An example of relational data is a data set of persons in a social network, where a link between two persons indicate that they are friends. Here persons are compared by their connections to other persons and not by their features.

Clustering is an exploratory task used usually as a first step in understanding something about the data by discovering patterns (or regularities). Due to this nature of the unknown, clustering has two intrinsic difficulties. First, different notions of similarity

can yield different clusterings. As an example, consider the tomato: fruit or vegetable? Botanically, the tomato is considered a fruit because it has seeds. However, tomatoes have a much lower sugar content than the other fruits and this has led to the culinary use of tomatoes as vegetables. In Nix v. Hedden, 149 U.S. 304 (1893), the U.S. Supreme Court settled a controversy caused by a set of tariff laws that imposed a duty on vegetables, but not on fruits, by declaring the tomato a vegetable, based on the popular definition that classifies vegetables by culinary use—they are generally served with dinner rather than dessert. Therefore, a recurring question when confronted with the task of clustering is what similarity measure to use.

The other difficulty with clustering is determining the number of clusters. As an exploratory endeavor, the number of clusters is just as unknown as the cluster assignments. Several approaches exist that use a separate measure to choose from cluster assignments for different number of clusters (Thorndike, 1953; Goutte et al., 2001; Sugar et al., 2003; Lletí et al., 2004; Sibson, 1973; Defays, 1977; Ankerst et al., 1999). Again, we are faced with the problem of which measure to use.

Kleinberg (2002) proved an impossibility theorem for clustering that states that no clustering algorithm can exist that satisfies the following three properties: (1) scale invariance: changes in the scale of the distance measure between data points should not change the resulting clustering, (2) richness: every possible clustering of the data should be attainable by the algorithm provided the appropriate distance measure be used, and (3) consistency: if the distances between points within clusters are made shorter and distances between points from different clusters are made larger, the output of the algorithm should

not change.

Given all of these difficulties, clustering is inherently tied to its application and no universal clustering method exists. In this dissertation we study the application of co-clustering to social network data and to medical data. In particular, we provide solutions to three main problems: (1) clustering with regular equivalence in social network analysis, (2) finding a symmetric clustering of asymmetric data and (3) clustering patients based on their high-dimensional, time-varying, sparse physiologic data.

1.1 Background

Before we delve into the details, we discuss the type of clustering we perform, co-clustering. Specifically, we review different methods in traditional one-mode clustering and in co-clustering, and discuss why co-clustering is the right method for the applications we target.

1.1.1 Clustering (one-mode)

The traditional way of approaching the task of clustering has been to base the grouping of objects on an explicitly predefined notion of similarity. Two aspects characterize this approach. First, traditional clustering algorithms work on affinity data—data describing the similarity between objects. Data normally consist of a set of feature vectors, each describing one object. An explicit similarity function must be designed to transform the input feature data into affinity data before employing these algorithms. Second, traditional clustering makes sense only when dealing with one-mode data. In one-mode data, all

objects are of the same type. In some applications, such as social network analysis, data can be two-mode since we are dealing with relations between objects, which may be of different types. Two-mode data will be explained in more detail in Section 2.1, where we introduce relational data.

Given the exploratory nature of clustering, a myriad of methods have been studied in the literature, each one discovering different types of structure in the data. They can broadly be placed in three categories based on the type of structure they find: flat clustering, hierarchical clustering and overlapping (non-partitional) clustering. Maimon and Rokach (2005) presents a summary of the most well-known methods.

Flat clustering methods

A flat clustering of a set of objects is a partition of the objects: A division into a set of collectively exhaustive, disjoint subsets.

Centroid-based clustering. These methods assume data are clustered around K centroids (cluster IDs) which, in the case of the K-means algorithm (MacQueen, 1967) (a very popular centroid-based method), are the means of the data in the clusters. The goal is to find the centroids that best cluster the data by optimizing a measure of compactness of the clusters, usually an intra-cluster homogeneity measure. For example, k-means minimizes the sum-of-squares of the distance of all points to their assigned cluster centroids. These methods often employ an alternating optimization scheme: they alternate between assigning each object to the closest centroid and updating the centroids of the newly formed clusters (Lloyd, 1982). This process monotonically reduces the cost function and thus is guaranteed

to converge to a local optimum. An initial set of centroids must be chosen in advanced. Picking different initial centroids can produce different clusterings. The clusters found by centroid-based methods are spherical (convex) in Euclidean space (Jain et al., 2000).

Graph-theoretic clustering. These methods seek to partition a graph by identifying a minimum cut. A cut is a graph partition—a clustering—obtained by the removal of a set of edges, known as the cutset, which disconnects the components (clusters) of a graph. A minimum cut is one whose cutset has a minimum sum of weights. The arc weights are analogous to the similarity measure between pairs of objects. The connected components in the graph are the clusters. The minimum cut method seeks to optimize an inter-cluster separability criterion as opposed to an intra-cluster homogeneity such as in k-means. To sidestep the possibility of skewed partitions—with degenerate clusters of a single node—measures of normalized cuts have been proposed (Shi and Malik, 1997).

Spectral clustering. These methods project the points onto a lower-dimensional space, which preserves most of the information from the original space, formed by the K largest eigenvectors in the range of eigenvectors, or spectra, of the Laplacian (Shi and Malik, 2000; Dhillon et al., 2004). The projected points are then clustered by a simple centroid-based algorithm, such as K-means (Ng et al., 2001). Spectral methods have an advantage over centroid-based methods in that they can discover more complex, non-convex clusters: In spectral methods, objects are clustered together because they are linked and not because they are close to each other.

Hierarchical clustering methods

Hierarchical methods produce hierarchical clusterings—clusters within other clusters. Each level in the hierarchy represents a different clustering. At the lowest level, each object is in its own cluster. Each successive level contains one fewer cluster after merging the two closest clusters from the previous layer. At the top level, all the objects are in the same cluster. This hierarchy is represented by a tree structure, called a dendrogram. Johnson (1967) gives a detailed description. It is possible to output a flat clustering by specifying an additional criterion for selecting among the levels. Ding and He (2002) gives a review of hierarchical methods.

Agglomerative. These methods work from the bottom up by initially placing each object into its own singleton cluster (at the lowest level) and iteratively merging pairs of clusters in order of pairwise distance among clusters until all objects are in one cluster (at the top level) (Guha et al., 1999). In addition to specifying a pairwise similarity measure between objects, a distance measure between pairs of clusters is also required, that is used to determine the next pair of clusters to merge (the closest ones). Various distance measures have been proposed. Single linkage (Sibson, 1973) measures the distance between two clusters as the shortest distance between two elements, one from one cluster. Complete linkage (Defays, 1977) uses the two elements that are farthest away. Complete linkage avoids the problem of single linkage, where otherwise distant clusters are merged because on element happens to be close to the other cluster. Complete linkage tends to find clusters of approximately equal diameters Everitt et al. (2009).

Divisive. These methods work in the opposite direction as agglomerative methods. Starting with all objects in a single cluster, iteratively split clusters one by one (Gunoche et al., 1991). These methods rely on the use of flat clustering methods (as a subroutine) to perform the splitting. Some have argued that divisive methods may perform better than agglomerative methods: In text clustering, documents belonging to different classes may be placed in the same cluster at the earliest stages of agglomerative clustering and these "mistakes" cannot be fixed once they happen (Steinbach et al., 2000).

Overlapping clustering methods

Other methods are based on probabilistic generative models[1]. These methods produce clusterings that are not partitions of the data: Objects may belong to more than one cluster. Instead, the assignment of each point to a cluster is a probability distribution over the clusters. Probabilistic methods assume that each data point is generated by one of K predefined probability distributions—the most popular being Gaussian distributions in Gaussian mixture models (GMMs)—with some probability, also usually following a multinomial distribution. The goal is to find the parameters of the Gaussians and the probability distribution over these Gaussians that had the highest probability (maximum likelihood) of generating the data. Some of the most popular methods used for model parameter estimation include the expectation maximization (EM) algorithm (Dempster et al., 1977) and Markov chain Monte Carlo methods such as Gibbs sampling (Geman and Geman, 1984).

[1]The other previous methods can be classified as discriminative methods, to contrast with generative methods. Generative methods learn a joint probability distribution $p(x, y)$, where x is the data and y is the cluster assignment, which can be used to determine the posterior $p(y|x)$ that assigns a cluster to an item as well as used to generate new samples. On the other hand, discriminative methods learn the posterior directly, that is, the cluster assignments. See Ng and Jordan (2001) and Lasserre and Bishop (2007).

The EM algorithm similar in structure to K-means: The model is initialized with arbitrary parameters and then alternates between calculating the probability for each data point of being generated by each cluster, based on the given parameters, (E-step) and updating the parameters of each cluster using the previously computed probabilities (M-step). The Gibbs sampling algorithm is a randomized algorithm (useful when the joint distribution is not known or explicitly or difficult to sample from) that estimates the model parameters by repeatedly drawing samples from their corresponding full conditional distributions (conditional on the rest of the parameters). The sequence of samples form a Markov chain whose stationary distribution is the sought-after joint distribution (Gelman et al., 1995).

1.1.2 Co-clustering (two-mode)

Co-clustering is a method of clustering that seeks to simultaneously cluster the rows and columns of a two-dimensional data matrix. Data in many applications exist as two-dimensional matrices: word-document co-occurrence tables in text analysis, gene-experimental condition expression data in bioinformatics, consumer-product purchasing data in market analysis, and actor-actor relation data in social network analysis. Different from affinity data, these data represent relationships between pairs of objects: An entry in the matrix relates an object from the rows with an object from the columns.

Instead, of defining an explicit similarity measure as a criterion for clustering two objects together, co-clustering searches for patterns in the data in the form of sub-matrices (or blocks) that conform to some measure of homogeneity. These matrix blocks, in turn, induce a partition on the rows and on the columns. The induced clustering on the rows and

columns can be different and thus, co-clustering can handle data matrices where the rows and columns index two different sets of objects (two-mode data).

Co-clustering can be regarded as a more general method than traditional one-mode clustering because it can cluster both relational data and affinity data (and feature data without an explicit affinity measure). Feature data can be arranged in matrix form, where each row of the matrix is a feature vector describing some object. Applying co-clustering to feature data produces a clustering of the features as as well as a clustering of the objects. By clustering the features, co-clustering performs an implicit dimensionality reduction, so fewer parameters are estimated, resulting in an implicit "regularized" clustering (Dhillon et al., 2003). In other words, co-clustering has the potential for generalizing better than one-mode clustering. This makes co-clustering less susceptible, than traditional one-mode clustering, to missing or corrupted values (Banerjee et al., 2004) and irrelevant features. Co-clustering is considered to perform at least as well as traditional one-mode clustering (Rohe and Yu, 2012).

Co-clustering has proved to be very effective and has been studied in various fields under different names. One of the earliest works (in statistics) calls it direct clustering (Hartigan, 1972) because it uses the data directly instead of a precomputed similarity measure as in traditional clustering. In computer science it has been studied under various names: co-clustering (Dhillon et al., 2003), cross-associations (Chakrabarti et al., 2004), bi-clustering (Cheng and Church, 2000) and box-clustering (Mirkin et al., 1995). In sociology, it is most commonly known as blockmodeling (Batagelj et al., 1992b).

1.2 Contributions

In this dissertation we study co-clustering to solve several problems. In particular, we show how most co-clustering methods in the literature conform to a general formulation and define implicit similarity measures for the following tasks: (1) clustering social network data, (2) finding a symmetric co-clustering for asymmetric one-mode data and (3) clustering medical data.

In social network analysis, sociologists have long been in need of a principled method for blockmodeling under regular equivalence; they have relied on assigning arbitrary penalties in their cost functions to obtain a desired blockmodeling (Doreian et al., 2004a, 2005; Brusco and Steinley, 2009, 2011). We provide a cost function based on compression theory that allows a co-clustering algorithm to automatically select among structural and regular blocks in a matrix.

In some applications in social network analysis we may have data matrices where the rows and columns index the same set of objects, and thus we seek a single clustering for both rows and columns. However, these matrices may be asymmetric, and applying straight up co-clustering will most likely produce a different clustering of the rows and columns. For this, we provide a general framework that can be applied to most co-clustering algorithms to obtain a single clustering.

Clustering medical data presents several challenges. Here, each patient is described by multi-dimensional longitudinal data that are both sparse and incomplete. Our solution tackles two main issues: (1) how to compare patients based on such complex data and (2) how to transform these data to fixed-length vectors for each patient is not obvious.

In the next subsection we present a general formulation for co-clustering that fits most methods in the literature. A description of our work on the targeted applications is presented in subsequent chapters.

1.2.1 General Formulation

Here we articulate a general formulation for co-clustering that fits most methods in the literature. Let $X = [x_{i,j}]$ denote an $M \times N$, binary data matrix, where $i = 1, \ldots, M$ and $j = 1, \ldots, N$ index the rows and columns, respectively. Denote by K and L the number of disjoint row and column clusters, respectively. A co-clustering of $K \times L$ co-clusters is a pair of mappings (ϕ, ψ):

$$\begin{aligned} \phi &: \{1, 2, \ldots, M\} \mapsto \{1, 2, \ldots, K\}, \\ \psi &: \{1, 2, \ldots, N\} \mapsto \{1, 2, \ldots, L\}. \end{aligned}$$

The general co-clustering problem can be formulated as an optimization problem where we are interested in finding a co-clustering $(\phi^\star, \psi^\star)$ that minimizes a cost function:

$$(\phi^\star, \psi^\star) = \underset{\phi,\psi}{\operatorname{argmin}}\ F(\phi, \psi) \tag{1.1}$$

where $F(\phi, \psi)$ is the cost of a co-clustering (ϕ, ψ). In most cases, F takes on a particular form:

$$F(\phi, \psi) = \min_{Z \in \mathcal{Z}} \sum_{i,j} f(X, Z, i, j, \phi(i), \psi(j))\,. \tag{1.2}$$

The auxiliary variable $Z = [z_{kl}]$ encodes some type of information about the co-clusters (matrix blocks) and $f(X, Z, i, j, \phi(i), \psi(j))$ computes a goodness of fit of a matrix element x_{ij} to the co-cluster $z_{\phi(i),\psi(j)}$ to which it has been assigned. For example, Batagelj et al.

(1992b) present a method for blockmodeling binary matrices where each matrix block is described as either being a block of all zeros or a block of all ones, that is, each block is either a zero-block or a one-block, $Z = \{0,1\}^{K\times L}$. The goodness of fit of each matrix element x_{ij} is its difference to the assigned block, $f(X, Z, i, j, \phi(i), \psi(j)) = |x_{ij} - z_{\phi(i),\psi(j)}|$.

In fact, most objective functions for co-clustering in the literature conform to Equation 1.2. Following are a few examples. If X is a real-valued matrix, constant-block co-clustering (Hartigan, 1972) can be formulated as $\mathcal{Z} = \mathbb{R}^{K\times L}$ (the means of the co-clusters) and $f(X, Z, i, j, k, l) = (x_{ij} - z_{kl})^2$.

Information-theoretic co-clustering (Dhillon et al., 2003) assumes X represents a joint probability distribution and approximates it according to its block structure. The approximation, $\mathcal{Z} = \{Z \in [0,1]^{M\times M\times K\times L} \mid z_{ijkl} = z_{kl}z_{ik}z_{jl} \text{ and } \sum_{ijkl} z_{ijkl} = 1\}$, is a distribution over row-index, column-index, row-cluster, and column-cluster that obeys a particular factorization. The goodness of fit, $f(X, Z, i, j, k, l) = x_{ij}\ln(x_{ij}/z_{ijkl})$, is the KL-divergence between X and Z. As an example, consider the data matrix below that corresponds to the joint probability distribution of random variables A and B:

$$X = p(A, B) = \begin{bmatrix} .15 & .15 & 0 & 0 \\ .15 & .10 & 0 & 0 \\ 0 & 0 & .15 & .15 \\ 0 & 0 & 0 & .15 \end{bmatrix}$$

Looking at the row distributions, it is natural to group the rows into two clusters: $\hat{a}_1 = \{a_1, a_2\}$ and $\hat{a}_2 = \{a_3, a_4\}$. Similarly, a natural clustering of the columns is $\hat{b}_1 = \{b_1, b_2\}$ and $\hat{b}_2 = \{b_3, b_4\}$. Given this co-clustering, let us compute the approximation of element $x_{1,2} = p(a_1, b_2)$ as $q(a_1, b_2)$: The probability of the matrix block of which $x_{1,2}$ is a member is $p(\hat{a}_1, \hat{b}_1) = \sum_{i|\phi(i)=\phi(1),j|\psi(j)=\psi(2)} x_{ij} = .55$. The conditional probability of row

1 given row cluster 1 is $p(a_1|\hat{a}_1) = (\sum_j x_{1,j})/p(\hat{a}_1, \hat{b}_1) = .55$. Similarly, the conditional probability of column 2 given column cluster 1 is .45. The approximation of $x_{1,2}$ based on this co-clustering is then $q(a_1, b_2) = p(\hat{a}_1, \hat{b}_1)p(a_1|\hat{a}_1)p(b_2|\hat{b}_1) = .55 \cdot .55 \cdot .45 = .136$. The approximation of the entire data matrix given the co-clustering is

$$q(A, B) = \begin{bmatrix} .166 & .136 & 0 & 0 \\ .136 & .111 & 0 & 0 \\ 0 & 0 & .099 & .202 \\ 0 & 0 & .049 & .099 \end{bmatrix}$$

The best co-clustering is then the one that minimizes the KL divergence between the original data and its approximation. Banerjee et al. (2004) has a more general treatment of the same style, allowing for different constraints on the set $\mathcal{Z}$ and different distance measures f.

The work of Chakrabarti et al. (2004) considers binary matrices X. $\mathcal{Z} = \{Z \in [0, 1]^{K \times L}\}$ describes each matrix block by the frequency of 1s in the block. $f(X, Z, i, j, k, l) = x_{ij} \lg z_{kl} + (1 - x_{ij}) \lg(1 - z_{kl})$ computes the number of bits, by way of Shannon's entropy (Shannon, 1948), required to encode the matrix element x_{ij} according to the Bernoulli distribution described by the matrix block $z_{\phi(i),\psi(j)}$. The Shannon entropy computes the theoretical minimum number of bits, on average, required to encode discrete data according to the distribution of values. Multiplying the Shannon entropy by the number of elements in a binary matrix sub block gives the formula for f. Their formulation also considers changes to K and L (the number of clusters) through minimum description length (MDL) model selection.

Given an optimization criterion with the form of Equation 1.2, the proposed solution algorithms (for instance in the work cited above) have the same coordinate ascent form. Starting from a randomly selected ϕ, ψ, they repeatedly hold ϕ and ψ fixed and find

Input: $X, \phi^{(0)}, \psi^{(0)}$
Output: (ϕ^*, ψ^*)

1 **begin**
2 $\quad s \leftarrow 0$
3 $\quad$ **repeat**
4 $\quad\quad Z^{(s+1)} \leftarrow \texttt{update}_z(\phi^{(s)}, \psi^{(s)})$
5 $\quad\quad \phi^{(s+2)} \leftarrow \texttt{update}_\phi(\psi^{(s)}, Z^{(s+1)})$
6 $\quad\quad Z^{(s+3)} \leftarrow \texttt{update}_z(\phi^{(s+2)}, \psi^{(s)})$
7 $\quad\quad \psi^{(s+4)} \leftarrow \texttt{update}_\psi(\phi^{(s+2)}, Z^{(s+3)})$
8 $\quad\quad s \leftarrow s + 4$
9 $\quad$ **until** *no more changes to* (ϕ, ψ)

Figure 1.1: Base co-clustering algorithm. The functions $\texttt{update}_z$, $\texttt{update}_\phi$ and $\texttt{update}_\psi$ are specific to each application problem.

an optimal Z, and then hold Z and ψ (or Z and ϕ) fixed and optimize ϕ (or ψ). Because of the additive nature of F, this second optimization can be done separately for each $\phi(i)$ (or $\psi(j)$). That is, in optimizing ϕ (or ψ) we can consider each row (or column) independently and select the best cluster for it. In particular, the algorithms select (for some fixed K, L)

$$\begin{aligned} \phi(i) &= \operatorname*{argmin}_k \sum_j f(X, Z, i, j, k, \psi(j)) \\ \psi(j) &= \operatorname*{argmin}_l \sum_i f(X, Z, i, j, \phi(i), l) \ . \end{aligned} \tag{1.3}$$

For example, in the case of constant-block co-clustering (Hartigan, 1972), row i will be assigned to the row cluster k that minimizes $\sum_j (x_{ij} - z_{k,\psi(j)})^2$.

The differences among the methods are in the choices of f and the resulting methods for minimizing Z given a fixed co-clustering. Figure 1.1 presents the "base" co-clustering algorithm we use in our work. Lines 4 and 6 correspond to optimizing Z given a fixed co-clustering. Lines 5 and 7 correspond to updating the row and column cluster assignments, according to the chosen f.

Chapter 2

Co-clustering for Social Network Data

We present a new criterion function for blockmodeling two-way two-mode relation matrices when the number of blocks as well as the equivalence relation are unknown. For this, we specify a measure of fit based on data compression theory, which allows for the comparison of blockmodels of different sizes and block types from different equivalence relations. We demonstrate that the method reproduces consensual blockings of three real world data sets without any pre-specification. We perform a simulation study where we compare our compression-based criterion to the commonly used criterion that measures the number of inconsistencies with an ideal blockmodel.

2.1 Relational data

Relational data, as the name suggests, encode pairwise relationships between objects. Our goal is to group objects together based on how they relate to each other and not on their intrinsic properties—features. Relational data are also known in social network analysis as *dyadic* data, where a relation is defined for each pair of entities—a *dyad*. Relational data can be binary (either there exists a relation or not between two entities), categorical (relations are categorized by type), or numerical (relations have associated strengths). One example of binary relational data is an "acquaintance" network, where two individuals either know each other or not. An example of numerical data is world trade data, which measure the amount of trade of a specific product between pairs of countries. Relational data can also be symmetric or asymmetric. As an example of symmetric data consider the co-worker network, which encodes the "works with" relation. If employee a "works with" employee b, then employee b necessarily "works with" employee a. A world trade network that encodes the "exports to" relation is an example of asymmetric data. These data are asymmetric because if country a exports product x to country b, it is likely that country b does not export product x back to country a; the relation has a direction associated with it.

Relational data can be represented as a two-dimensional matrix, where a matrix entry relates an object from the rows to an object from the columns. It can also be represented by a graph with either unweighted edges (binary data) or weighted edges (numerical data), where an edge $\{u, v\}$ represents a relationship between objects u and v. A directed graph can also be used to encode asymmetric data. For example, world trade data encoding

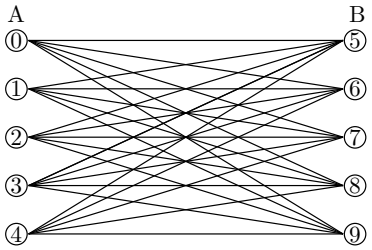
the "exports to" relation can be represented as a graph where a directed edge (u, v) indicates that country u exports to country v, and the weight of the edge denotes the amount of trade.

Dyadic data can also be one-mode or two-mode. Two-mode data are defined for two sets of objects, relating objects in one set to objects in the other. The world trade data is one-mode because the rows and the columns index the same set of objects. The most studied two-mode data set in the social network analysis literature is the Deep South data set, collected by Davis et al. (1941), of a set of women attending social events over a period of nine months. Here the two modes are the women and the social events they attended (or did not attend).

2.1.1 Clustering Relational Data

A reasonable question to ask is why cannot we use traditional one-mode clustering algorithms to cluster relational data? First, traditional one-mode clustering only works for affinity data, which is both one-mode and symmetric. But, assuming the data at hand is one-mode and symmetric, what makes relational data different from affinity data?

The network depicted in Figure 2.1 is an idealized example of one-mode symmetric binary relational data. These data form a bipartite graph with to sets of nodes, A and B, that have edges between them, but not within each set. This is often the case with relational data. In social network analysis, where the data encode pairwise relationships, the nodes in set A are considered similar because they have the same relationship with the same other set of nodes, B, and not because they are connected among themselves, which they are not



		A				B				
		0	1	2	3	4	5	6	7	8	9
A	0	0	0	0	0	0	1	1	1	1	1
	1	0	0	0	0	0	1	1	1	1	1
	2	0	0	0	0	0	1	1	1	1	1
	3	0	0	0	0	0	1	1	1	1	1
	4	0	0	0	0	0	1	1	1	1	1
B	5	1	1	1	1	1	0	0	0	0	0
	6	1	1	1	1	1	0	0	0	0	0
	7	1	1	1	1	1	0	0	0	0	0
	8	1	1	1	1	1	0	0	0	0	0
	9	1	1	1	1	1	0	0	0	0	0

Figure 2.1: Idealized one-mode symmetric binary network with ten nodes and two clusters: A and B. The graph representation, a bipartite graph, on the left and the matrix representation, with complete blocks off the main diagonal, on the right.

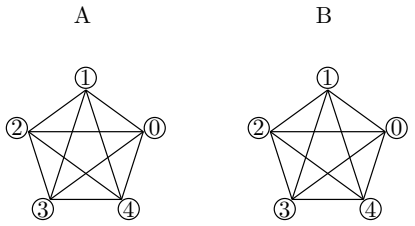
at all in this example. This is different from affinity data, where the criterion for clustering items together is that they must be connected between them.

Given that traditional clustering has a different clustering criterion, the clustering produced by any such algorithms should not be expected to be the desired one. Indeed that is the case in this example. Here we seek a cluster assignment sequence of $AAAAABBBBB$. Applying the Matlab spectral clustering implementation SpectraLIB[1] gives the clustering $BABBABABAA$. Similarly, applying the normalized cut-based graph partitioning algorithm Graclus[2] produces the clustering $BAABBABBBA$. By interpreting graph edges as affinities, graph clustering algorithms will try to cluster together nodes from A and B.

A quick look at the matrix representation in Figure 2.1 and we can see how the data can be easily divided into four sub-matrices (or blocks), and these blocks induce a partition on the rows and the columns equal to A and B in the graph representation. The blocks with all zeros encode the absence of a relationship between the nodes within a group

[1]Obtained from `http://www.stat.washington.edu/spectral/`

[2]Obtained from `http://www.cs.utexas.edu/users/dml/Software/graclus.html`



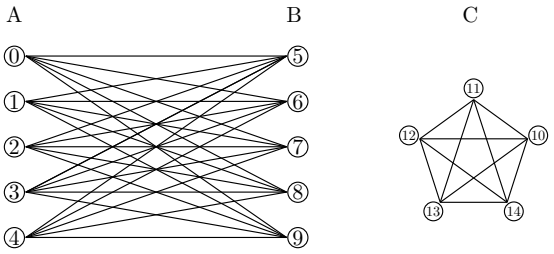
		A				B				
		0	1	2	3	4	5	6	7	8	9
A	0	1	1	1	1	1	0	0	0	0	0
	1	1	1	1	1	1	0	0	0	0	0
	2	1	1	1	1	1	0	0	0	0	0
	3	1	1	1	1	1	0	0	0	0	0
	4	1	1	1	1	1	0	0	0	0	0
B	5	0	0	0	0	0	1	1	1	1	1
	6	0	0	0	0	0	1	1	1	1	1
	7	0	0	0	0	0	1	1	1	1	1
	8	0	0	0	0	0	1	1	1	1	1
	9	0	0	0	0	0	1	1	1	1	1

Figure 2.2: Network obtained by flipping the bits in the network of Figure 2.1. The graph representation, with two connected components, on the left and the matrix representation, with complete blocks on the main diagonal, on the right.

and the blocks with all ones encode the existence of a relationship between nodes of the two groups.

In social network analysis the absence of ties between actors does not necessarily imply that they are not similar. In sociology, similarity is often defined by the concept of structural equivalence (Lorrain and White, 1971). Two nodes in a graph are structurally equivalent if they are adjacent to the same set of nodes, other than each other. Thus in a friendship network, two individuals are structurally equivalent if they have the same friends. Note that two nodes are perfectly equivalent without actually being connected to each other—the existence or not of an edge between them is not important. This condition can be modeled by a bipartite graph (Figure 2.1), where the set of nodes can be divided into two subsets such that for each edge $(u, v) \in E$, node u is in one subset and node v is in the other.

Given that traditional clustering will not produce the correct clustering on relational data, can we transform relational data into an equivalent affinity data? If this



		A					B					C				
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
A	2	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
	3	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
	4	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	6	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
B	7	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	8	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	9	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	11	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
C	12	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	13	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	14	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

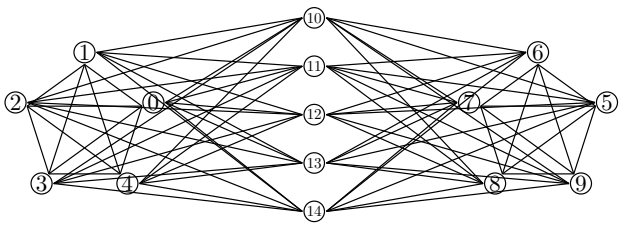
Figure 2.3: Idealized one-mode symmetric binary network of 15 nodes with three complete blocks, both on and off the main diagonal, and three clusters, A, B and C.

transformation could be done cheaply, traditional one-mode clustering may be used.

If we flip the bits in the network of Figure 2.1, we obtain the network in Figure 2.2. The matrix representation in the figure depicts a matrix with blocks along the main diagonal. This type of structure in a matrix corresponds to connected components in a graph representation, such as depicted in the same figure. This is the kind of network for which one-mode clustering is intended.

By flipping the bits we have effectively transformed the relational data in this example to affinity data. The intuition for why this transformation worked is that an edge was placed between any pair of nodes that have an edge to the same other node. By doing this as a first step, the bipartite network can be clustered by a traditional method to obtain the desired clustering of $AAAAABBBBB$.

However, this simple transformation of flipping the bits does not work in the general case. Consider the network in Figure 2.3. Flipping the bits of this network does not get



		A					B					C				
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	0	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
A	2	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
	3	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
	4	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
	5	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	6	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
B	7	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	8	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	9	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	10	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
	11	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
C	12	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
	13	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
	14	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

Figure 2.4: Network obtained by flipping the bits in the network of Figure 2.3.

rid of off-diagonal blocks, which form bipartite sub-graphs that one-mode clustering cannot handle. Here we seek a clustering of $AAAAABBBBBCCCCC$. Applying the Matlab spectral clustering implementation SpectraLIB gives the clustering $AABBBCCCCCBAAAB$. Applying the normalized cut-based graph partitioning algorithm Graclus gives the clustering $AAABABCCCCCABBB$.

To be able to apply one-mode clustering on the network of Figure 2.3, we would need to transform it so that we end up with three connected components. This would require placing edges between nodes according to two opposing criteria: (1) if nodes have edges to the same other nodes in the original network, clusters A and B, and (2) if nodes are connected in the original network, cluster C. Deciding whether to connect two nodes because they are connected to the same other node or because the are connected in the original network would require knowing the desired clustering in the first place. Figure 2.4

	Children					
Mothers	1	0	0	0	1	0
	0	1	0	0	0	0
	0	0	1	0	0	1
	0	0	0	1	0	0

Figure 2.5: Example of a regular block of a binary network that encodes the "mother-of" relation. Persons along the rows are mothers of the persons along the columns.

shows the network after a simple flipping of the bits.

For all the examples above, with the bits flipped or not, co-clustering can determine the correct clusters by discovering the blocked structure of the matrix. Co-clustering is, in this way, more general than traditional one-mode clustering as it can cluster both affinity (blocks along the diagonal) and relational (blocks on and off the diagonal) data.

2.1.2 Regular Equivalence

Regular equivalence is a generalization of structural equivalence (White et al., 1976; Sailer, 1978; Pattison, 1982; White and Reitz, 1983; Kim and Roush, 1984; Everett and Borgatti, 1993; Pattison, 1993; Boyd and Everett, 1999). Under regular equivalence two actors are equivalent to each other if they have ties to similar actors, but not necessarily the same actors. As a concrete example, consider a network that encodes the "mother of" relation (Figure 2.5). Mothers will be grouped together because they are all mothers of children (also grouped together). However, it is clear that two mothers cannot be mothers of the same child.

Regular equivalence can form matrix blocks that are not completely homogeneous, as opposed to structural equivalence. The basic type of regular blocks are 1-covered blocks. A 1-covered block is defined as having at least one 1 in each row and each column—every

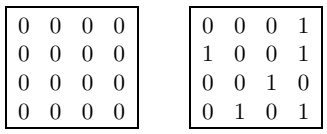


Figure 2.6: Examples of the two basic block types under regular equivalence.

row and column is covered. Figure 2.6 shows examples of the two basic types of blocks under regular equivalence: null and 1-covered. Other types exist, such as row-regular and column-regular, where only the rows are covered and only the columns are covered, respectively. Doreian et al. (1994, 2004a) defines additional types of regular blocks.

While regular equivalence is theoretically appealing, some have questioned its generality. For example, Boyd and Jonas (2001) and Boyd (2002) argue that data should not be assumed to contain regular blocks. They showed, by way of a permutation test, that regular blocks found on three well studied data sets had more errors than regular blocks found by random permutations of the data. Nevertheless, the authors concede that some data in nature exhibit regular equivalence. Situations where an "exclusion principle" holds, such as in the mother-of example in Figure 2.5, have regular equivalence relations.

2.2 Related Work

Blockmodeling tools were developed in the social network literature to partition actors in a network into clusters. One of the earliest works (Batagelj et al., 1992a,b) called it "direct" blockmodeling (uses the data directly) to contrast it to the traditional clustering approach, which operates on the data "indirectly" via a dissimilarity measure of the actual

network values.

The task of clustering two-mode data has been well studied in the sociology literature and a number of methods have been proposed (refer to Krolak-Schwerdt (2003); Mechelen et al. (2004) for two surveys), with the relocation algorithm with multiple restarts being the predominant approach (Brusco and Steinley, 2011). Mirkin et al. (1995) introduced additive box clustering as an improvement over a previous error-variance box clustering by Eckes and Orlik (1993), which sequentially builds matrix blocks by adding matrix elements, guided by a criterion function based on sum-of-squares. Hansohm (2002) proposed a genetic algorithm and compared it to an alternating exchanges algorithm (a local search in the same spirit of the relocation algorithm) by Gaul and Schader (1996). The study showed that the genetic algorithm was not superior for two-mode partitioning problems. Trejos and Castillo (2000) presented a simulated annealing version of the alternating exchanges algorithm by Gaul and Schader (1996). Their simulated annealing version provided superior solutions than the alternating exchanges algorithm at the expense of a substantial increase in running time, as expected (Aarts and Korst, 1989). Brusco and Steinley (2011) presented a tabu search heuristic that provided better solutions than the relocation heuristic over an extensive study where the two methods were allowed the same amount of computation time. In a comparative study by van Rosmalen et al. (2009), a two-mode k-means algorithm outperformed implementations of various other methods, including simulated annealing and tabu search. Brusco and Steinley (2007) did a simulation study comparing a two-mode variable neighborhood search algorithm with a more efficient, but less exhaustive, k-means-based search, finding that the results given by the two were comparable when both algorithms

were allowed to run for the same amount of time.

As stated by Doreian et al. (2004a), many blockmodeling tools in the literature have required that the block types and/or their locations on the network be pre-specified by the analyst. In more recent works, such as Doreian et al. (2005); Brusco and Steinley (2007, 2011), exploratory methods have been presented that do not require the image matrix[3] to be pre-specified. Brusco and Steinley (2009) presented an integer program that guarantees an optimal solution for a given image matrix. They suggest a two-stage procedure where a heuristic method is first used to obtain an image matrix, and then the integer program provides the optimum solution for that image matrix. In Brusco and Steinley (2007), one of the methods implemented was a two-mode k-means with 500 restarts. This approach can determine the block placements, but still requires the pre-specification of the number of row and column clusters.

Let $X(\phi^{-1}(k), \psi^{-1}(l))$ denote the matrix block induced by the co-clustering (ϕ, ψ) and indexed by k, l, where $\phi^{-1}(k)$ denotes the set of matrix row indices that map to row cluster k (similarly, $\psi(\cdot)$ for the columns). A common criterion function in the literature (Doreian et al., 2005; Batagelj et al., 2004, 1992a,b; Borgatti and Everett, 1992) is one that measures the deviation of each block $X(\phi^{-1}(k), \psi^{-1}(l))$, induced by the co-clustering (ϕ, ψ), from the ideal block $B \in \mathcal{B}$, where $\mathcal{B}$ is the set of ideal blocks defined by the equivalence relation. The total deviation of the co-clustering (ϕ, ψ) is expressed as the sum of block deviations across all blocks induced by the co-clustering:

$$F(\phi, \psi) = \sum_{kl} d(X(\phi^{-1}(k), \psi^{-1}(l))), \tag{2.1}$$

[3]The image matrix is the matrix of block descriptions, Z in Equation 1.2

where $d(X(\phi^{-1}(k), \psi^{-1}(l)))$ is the block deviation determined by

$$d(X(\phi^{-1}(k), \psi^{-1}(l))) = \min_{B \in \mathcal{B}} \delta(X(\phi^{-1}(k), \psi^{-1}(l)), B), \tag{2.2}$$

and $\delta(\cdot, \cdot)$ measures the deviation between a block and an ideal block prototype.

As an example, for structural equivalence (Batagelj et al., 1992b; Doreian et al., 2005), the deviation of a block can be measured as

$$\delta(X(\phi^{-1}(k), \psi^{-1}(l)), B) = \sum_{i|\phi(i)=k, j|\psi(j)=l} |x_{ij} - b_{ij}|. \tag{2.3}$$

This counts the number of 1s in the block $X(\phi^{-1}(k), \psi^{-1}(l))$ if the ideal block B is null, and counts the number of 0s in $X(\phi^{-1}(k), \psi^{-1}(l))$ if the ideal block B is complete. The criterion function in Equation 2.1 adds the deviation of all blocks to their respective closest ideal blocks.

The most common optimization technique in the social networks literature is the relocation algorithm (Batagelj et al., 1992b,a; Borgatti and Everett, 1992; Doreian et al., 2005; Batagelj et al., 2004). It starts from an initial co-clustering and searches the space locally. The local neighborhood is determined by two transformations: (1) moving a unit (row or column) from one cluster to another, and (2) swapping two units from two different clusters. The procedure moves through the search space by selecting the neighbor for which the criterion function produces the smallest result. When no more neighbors produce a better result, the algorithm has reached a local minimum. The procedure is repeated from various starting points and the best solution is selected.

This formulation of blockmodeling requires that the number of clusters be specified *a priori*. If the algorithm is allowed to change K and L (the number of row and column

clusters), it could achieve zero error by setting $K = M$ and $L = N$ and placing each cell into its own block that could be perfectly described by a complete or null block. Therefore, the analyst must have *a priori* knowledge of how many clusters there should be. For many real-world problems the number of clusters is as unknown as the assignment of units to clusters.

To the best of our knowledge, none of the methods in the sociology literature can automatically determine the number of row and column clusters for two-mode blockmodeling. Moreover, additional fine tuning may be required such as defining different penalties on some inconsistencies relative to others. These requirements make the blockmodeling tools brittle.

Various methods outside the sociology literature have been proposed for determining the number of clusters. Methods such as those proposed by Tibshirani et al. (2000) measure intra-cluster scatter: For each cluster, an average pairwise distance is computed and the sum over all clusters is monitored as the number of clusters is incremented. This sum decreases rapidly as clusters are added and then remains somewhat stable after some K. Other methods combine the minimization of intra-cluster scatter with the maximization inter-cluster scatter to avoid a potential undesirable split of a distinct cluster (Ray and Turi, 1999). Other criteria are based on probability theory: Criteria such as BIC (Bayesian Information Criterion), MML (Minimum Message Length) and MDL (Minimum Description Length) add a penalty proportional to the number of clusters. It is this last approach, MDL, that we adopt in our solution, presented next.

2.3 Automatic Co-clustering for Regular Equivalence

The notion of regular equivalence is an important theoretical concept, yet sociologists have struggled to find a principled method for analyzing regular equivalence in data. Some have questioned its generality. For example, Boyd and Jonas (2001) and Boyd (2002) argue that data should not be assumed to contain regular blocks. They showed, by way of a permutation test, that regular blocks found on three well studied data sets had more errors than regular blocks found by random permutations of the data. Nevertheless, the authors concede that some data in nature exhibit regular equivalence. Situations where an "exclusion principle" holds, such as in the "mother of" example in Figure 2.5, have regular equivalence relations.

Goodness of fit under regular equivalence is defined as the number of covered rows/columns for null blocks and the number of uncovered rows/columns for 1-covered blocks (Batagelj et al., 1992a). The problem with this definition is that it admits multiple equally well-fitting 1-covered blocks of varying sizes and densities (Doreian et al., 2004a). To cope with this difficulty, works in the literature employ criterion functions that assign arbitrary penalties (Doreian et al., 2005; Brusco and Steinley, 2007, 2011). We provide a method that can discover both regular equivalence and structural equivalence in binary relational data and automatically determine the number of clusters. To achieve this, we implement a compression-based criterion function that captures the differences between structural and regular equivalence and can be used to compare co-clusterings of different sizes.

2.3.1 Compression as an Optimization Criterion

Clustering can also be viewed as a form of data compression where each cluster summarizes a group of similar units. We present a criterion function that evaluates a co-clustering in terms of how well it compresses the data. That is, if we were to transmit the entire data matrix, how short would the transmission be if the receiver also assumed that the matrix had either a regular or a structural blocking? The receiver does not know the block types (or even the number of clusters), but is expecting a block-structured matrix. This allows us to agree on an efficient language for describing such data.

This criterion function has the property that the closer the co-clustering of the data is to an ideal blockmodel, the lower the result of the criterion function is. From a communications theory point of view, we devise a measure of fit that computes the number of bits required to transmit the compressed data such that the original data can be reconstructed without loss at the receiving end. For example, if the data conforms to structural equivalence, that is, the underlying structure of the data forms co-clusters (blocks) that are homogeneous, that is, either null or complete, we can compress the data inside each block by just sending the type of the block instead of every cell. In general, most real-world problems do not have a perfect structure, so additional information must be sent that encodes the location of the matrix cells that are inconsistent with the ideal block type. Co-clusterings that produce lower costs of encoding are therefore better methods of compression and (in some sense) better describe the data by extracting structure.

The sender and receiver of the data have a predetermined communication protocol where the receiver expects the incoming data in a certain order and format. In particular,

we assume three parts to the protocol. First, the sender transmits the number of blocks and which rows and columns are assigned to which blocks. We call this the *overhead* encoding. Second, the sender transmits per block information specifying the ideal structure of each block. We call this the *block type* encoding. Lastly, the sender transmits any deviations from these ideal types. We call this the *error* encoding. Below we describe the encodings and how to calculate their lengths for a particular blocking.

2.3.2 Overhead Encoding

The overhead contains the number of clusters per way and the assignments of each way element to its cluster. We assume that the number of rows and columns are already known.[4] The number of clusters for a way with M elements can be encoded in $\lg(M) = \log_2(M)$ bits: We need a number between 1 and M. Each of the M items in this way can be assigned to its cluster by encoding the cluster's index in $\lg(K)$ bits (if there are K clusters). Given two ways of size M (with K clusters) and N (with L clusters), the total overhead is

$$\xi = \lg(M) + \lg(N) + M\lg(K) + N\lg(L)\ . \tag{2.4}$$

2.3.3 Block Type Encoding

The block type encoding describes ideal pattern for each block. Each block's encoding begins with an integer that encodes the type of the block. As we are considering 3 blocks (complete, null, 1-covered), this takes lg(3) bits. If the block is null or complete, this is sufficient to describe the ideal block (all cells should be either 0 or 1, respectively).

[4] If not, they can also be encoded using constant extra space, so it does not enter into the optimization.

However, if the block is 1-covered, this is not sufficient. We must further specify how the 1s are distributed. For a block of size m-by-n, we do this by computing the total number of possible 1-covered blocks and taking the logarithm (base 2) as the number of bits required to encode this ideal block. Thus, the number of additional bits required to specify a regular block of size m-by-n is

$$\zeta_{reg}(B) = \lg \left[\sum_{i=0}^{m} \sum_{j=0}^{n} (-1)^{(i+j)} \binom{m}{i} \binom{n}{j} 2^{(m-i)(n-j)} \right]. \tag{2.5}$$

The term inside the brackets is the total number of 1-covered blocks and is computed by subtracting from the total number of possible blocks, 2^{mn}, the total number of blocks that are not covered by at least one row or column. If we let S_i^C be the set of blocks in which column i is uncovered and let S_i^R be the same for row i, then we wish to compute the size of the union of all of these sets: $\bigcup_i S_i^C \cup \bigcup_i S_i^R$. This can be done by applying the inclusion-exclusion principle which results in Equation 2.5.

Thus the cost of encoding an ideal block B is

$$\zeta(B) = \begin{cases} \lg(3) & \text{if } B \text{ is null or complete} \\ \lg(3) + \zeta_{reg}(B) & \text{if } B \text{ is 1-covered.} \end{cases} \tag{2.6}$$

The total cost of encoding the ideal blocks is the sum of ζ over all blocks.

2.3.4 Error Encoding

Finally, we encode the location of any cells that do not conform to the ideal blocks transmitted above. For null blocks, these will be any 1s. For complete blocks, these will be any 0s. And for regular blocks, these will be any 0s that are in locations that a minimal cover would have dictated as 1s. If we let $\delta(X(\phi^{-1}(k), \psi^{-1}(l)))$ be the number of deviations

for a particular block $X(\phi^{-1}(k), \psi^{-1}(l))$, then this error cost for a given block is

$$\eta = \lg(MN) + \sum_{kl} \delta(X(\phi^{-1}(k), \psi^{-1}(l))) \lg(MN), \tag{2.7}$$

where the first term is the size of the encoding of the number of errors and the second part is the size of the encoding of the locations of each error (specified as a row-column pair).

2.3.5 Total Cost

The total cost of a blockmodel is the sum of the overhead ξ plus the cost ζ of each block plus the cost of the final errors:

$$\psi = \xi + \sum_{kl} \zeta(X(\phi^{-1}(k), \psi^{-1}(l))) + \eta \ . \tag{2.8}$$

Importantly, this evaluation scheme allows for the comparison of blockmodels that have different numbers of clusters. Note that increasing the number of clusters will reduce the value of η (because they better model the data), but it will increase ξ and $\sum_{kl} \zeta(X(\phi^{-1}(k), \psi^{-1}(l)))$. This criterion function defines a solution to the blockmodeling problem that balances the number of clusters versus the number of deviations. This trade-off between the complexity of the model and the complexity of the data given that model is described by the minimum description length (MDL) principle (Grünwald, 2005). For example, at one end of the spectrum, each element in the data matrix is assigned to its own block with zero error, but at the expense of the most complex model. At the other end of the spectrum, all elements are assigned to the same block, which is the simplest model possible, but with the maximum number of errors. We seek a solution between these two extremes that minimizes the criterion.

Note that if the number of clusters is fixed, only the block costs depend on the assignments of units to clusters (the overhead cost remains constant for all blockings) and it is (up to a positive scaling) $\sum_{kl} d(X(\phi^{-1}(k), \psi^{-1}(l)))$ in Equation 2.1, where $d(\cdot) = \eta$ in Equation 2.7 (number of bits required to encode the locations of the matrix elements that do not agree with their respective blocks). In this sense, this is a generalization of the former blocking optimization methods.

2.4 Optimization Algorithm

In Section 2.3.1 we presented the criterion function, which defines a solution to the blockmodeling problem as a minimization. Here we describe an algorithm for finding this solution by optimizing this criterion function efficiently. Different from standard blockmodeling algorithms, this algorithm searches over the space of all blockmodels (with different number of clusters) that admit the three block types: null, 1-covered and complete.

The algorithm is inspired by schemes presented in the information-theoretic co-clustering literature (Dhillon et al., 2003; Banerjee et al., 2004). It consists of a "base" alternating optimization scheme (presented in Figure 1.1) and a search over all possible K, L. The base alternating optimization scheme is similar in structure to the leader algorithm (Batagelj et al., 2004) or k-means (MacQueen, 1967).

The base co-clustering algorithm alternates between optimizing the block types, Z, while holding the row and column cluster assignments, (ϕ, ψ), fixed and optimizing ϕ (or ψ) while holding Z, ψ (or Z, ϕ) fixed. The block types are optimized by selecting the type (null, complete or regular) that minimizes the number of errors. For example, a block

of mostly ones will be best described by a complete block and a block of mostly zeros by a null block. When comparing among block types for a particular matrix block, only the number of deviations, $\delta(\cdot,\cdot)$, is considered since the $\lg(MN)$ terms in Equation 2.7 and the $\lg(3)$ terms in Equation 2.6 are constant over all block types. For null and complete blocks, the number of deviations is simply the number of block elements that do not agree with the block type. For a regular block, the number of deviations is the minimum number of 1s needed to make every row and column covered (to have at least one 1). Additionally, regular blocks incur a cost of encoding which regular block they are (which arrangement of 1s and 0s), shown in Equation 2.5.

Note that for structural (null and complete) blocks the choice of where to place each row (or column) is independent of where to place the other rows (or columns). The optimization criterion is the sum of the errors for each row (or column). For regular blocks, this is no longer the case. However, it is still approximately true in many circumstances and the performance is much faster than other optimization methods.

The alternating optimization finds a solution quickly given a fixed number of clusters. To find K, L, we added two additional transformations to change the number of clusters: (1) a transformation that potentially adds a cluster, and (2) a transformation that deletes a cluster. To add a new cluster, the algorithm finds the largest group of rows that are similar and puts them together in a new cluster. If the resulting blockmodel has a lower cost, it is accepted. The algorithm keeps adding clusters as long as the resulting blockmodel has a lower cost. The process is repeated on the columns. To delete a cluster, the algorithm tries deleting each row cluster in turn (and reassigning its rows) and selecting the one that

```
   Input: X
   Output: (φ*, ψ*)
1  [K, L] ← [√M, √N]
2  (φ, ψ) ← random_{φ,ψ}(K, L)
3  begin
4      repeat
5          Optimize cluster assignments // base co-clustering, Figure 1.1
6          try adding row clusters and update block types
7          try adding column clusters and update block types
8          try deleting clusters and update block types
9          try deleting column clusters and update block types
10     until no positive changes are possible
```

Figure 2.7: K, L search algorithm for social network data applications.

produces the lowest costing blockmodel. (If all increase the cost, no clusters are deleted.) Again, this is done until no better solution is attained. The process is repeated on the columns.

Together, the alternating optimization and the cluster addition and deletion provide the algorithm with the tools to search the space of all possible blockmodels that admit any of the three block types described in Section 2.1.2, that is, null, 1-covered, and complete. The algorithm is shown in Figure 2.7.

The algorithm starts from an initial random co-clustering and applies the transformations in a greedy fashion until no more improvement is possible. At this point the algorithm has reached a local optimum and may be restarted from a new random co-clustering. The final solution is the one with the lowest cost.

2.5 Experiments

We performed four types of experiments. First, we compared the solutions selected by our compression-based criterion function on two data sets previously studied in the literature to those provided by other authors. One is actor-event data on the social activities of 18 southern women over a nine-month period in the 1930s. The other is actor-decision data on the voting patterns of the Supreme Court on 26 important issues during the 2000-2001 term. Second, we performed a simulation study to compare our compression-based criterion to the more common error-based criterion that counts inconsistencies with an ideal blockmodel. Third, we tested our criterion on a larger data on economic activities compiled from a database maintained by the private vendor InfoUSA. Fourth, we evaluated the possibility of finding the number of clusters using the number of errors, as an alternative to our method.

2.5.1 Two previously studied data sets

Southern Women data set. The first data set comes from a socioeconomic study of the rural community of Natchez, Mississippi in the 1930s by Davis et al. (1941). We used the data as presented in Doreian et al. (2004a). The two-mode data, displayed in Figure 2.8, show the participation of a group of women in social events. Actor-event data are a staple of social network analysis. The goal of blockmodeling is to find classes of actors who are similar, based on their co-presence at events, while at the same time, finding classes of events that are similar because they elicited the affiliation of the same sets of actors. Actor-event problems generally have the research hypothesis of some set-wise correspondence, but

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14
Brenda	1		1	1	1	1	1	1						
Charlotte			1	1	1		1							
Dorothy								1	1					
Eleanor					1	1	1	1						
Evelyn	1	1	1	1	1	1		1	1					
Flora									1		1			
Frances			1		1	1		1						
Helen							1	1		1	1	1		
Katherine								1	1	1		1	1	1
Laura	1	1	1		1	1	1	1						
Myra								1	1	1		1		
Nora						1	1		1	1	1	1	1	1
Olivia									1		1			
Pearl			1			1		1	1					
Ruth				1	1		1	1	1					
Sylvia							1	1	1	1		1	1	1
Theresa		1	1	1	1	1	1	1	1					
Verne							1	1	1			1		

Figure 2.8: Southern Women event participation data. A 1 in the matrix denotes the participation of an actor in an event. Zeros in the data are not shown.

often theory does not provide much guidance about the expected number of classes in each model, or the equivalence relations of the blocking. The Southern Women data have been analyzed many times, and there is some consensus about the most meaningful blocking; there is, however, no non-trivial zero-error solution with only the block types considered here.

Figure 2.9 shows the co-clustering selected by our compression-based criterion function. The clusters of women obtained here match those concluded by a meta-analysis of 21 analyses of these data by Freeman (2003). The social events were assigned to the same clusters presented in Table 2 in Doreian et al. (2004a) with the exception of event E6. Note that these results were obtained without pre-specifying any information about the block types or their locations nor the number of clusters to be found.

The participation of each set of women in one set of events is "regular" in that not

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14
Evelyn	1	1	1	1	1	1		1	1					
Laura	1	1	1		1	1	1	1						
Theresa		1	1	1	1	1	1	1	1					
Brenda	1		1	1	1	1	1	1						
Charlotte			1	1	1		1							
Frances			1		1	1		1						
Eleanor					1	1	1	1						
Pearl			1			1		1	1					
Ruth				1	1		1	1	1					
Verne							1	1	1			1		
Myra								1	1	1		1		
Katherine								1	1	1		1	1	1
Sylvia							1	1	1	1		1	1	1
Nora						1	1		1	1	1	1	1	1
Helen							1	1		1	1	1		
Dorothy								1	1					
Olivia									1		1			
Flora									1		1			

Figure 2.9: Southern women data set. Solution selected by our compression-based criterion function.

every woman attended every event, but each attended at least one (that is, the upper left and lower right blocks). The women are also shown as having near-perfect null blocks for the other block of events (that is, upper right and lower left blocks). In addition, there is a third class of events that are regularly equivalent and attended by both factions of women (that is, E7, E8, and E9).

While the ultimate decision about whether a block should be regarded as structural or regular is a matter of the theory of the underlying process generating affinities or clusterings, the information-cost approach provides the solution that is the simplest complete description of the data. Such an efficient description that is agnostic with regard to both the numbers of blocks in each mode, and the nature of the equivalences may suggest patterns that, in turn, suggest richer theories of the underlying process. In the current case, each faction of women must not attend one set of events, and must attend one, but not

	Presidential Election	Illegal Search 1	Illegal Search 2	Illegal Search 3	Seat Belts	Stay of Execution	Federalism	Clean Air Act	Clean Water	Cannabis for Health	United Foods	New York Times Copyrights	Voting Rights	Title VI Disabilities	PGA vs. Handicapped Player	Immigration Jurisdiction	Deporting Criminal Aliens	Detaining Criminal Aliens	Citizenship	Legal Aid for the Poor	Privacy	Free Speech	Campaign Finance	Tobacco Ads	Labor Rights	Property Rights
Rehnquist	1				1		1	1	1	1	1	1		1	1				1			1		1	1	1
Thomas	1			1	1		1	1	1	1	1	1		1					1			1		1	1	1
Scalia	1			1	1		1	1	1	1	1	1		1					1			1		1	1	1
Kennedy	1	1	1		1	1	1	1	1	1	1	1		1	1	1		1	1	1	1	1		1	1	1
O'Connor	1	1	1			1	1	1	1	1		1	1	1	1		1				1	1	1	1	1	1
Breyer		1	1	1		1		1					1		1	1	1	1		1	1	1	1			
Ginsburg		1	1	1		1		1		1		1	1		1	1	1	1		1	1		1			
Souter		1	1	1	1	1		1		1	1	1	1		1	1	1	1		1	1		1			
Stevens		1	1		1	1		1		1	1		1		1	1	1	1	1	1	1		1			

Figure 2.10: Supreme Court votes on 26 cases. A 1 in the matrix indicates that a Justice voted with the majority on a case. Zeros in the data are not shown.

necessarily all of another set. There is also another class of bridging events, and it appears normative that all members of either faction must attend at least one of these.

Supreme Court voting data set. The second data set comes from a study by Doreian and Fujimito (2003) on the Supreme Court's decision on twenty six cases. The data set is presented here as in Doreian et al. (2004a), shown in Figure 2.10. The rows correspond to the nine Justices. The columns represent twenty six cases decided by the Supreme Court during the 2000-2001 term. A more detailed description of these cases can be found in Greenhouse (2001) and in Doreian and Fujimito (2003).

The solution selected by our compression-based criterion function is shown in Figure 2.11. The swing voters, Kennedy and O'Connor, were put in their own group. The more liberal Justices, Breyer, Ginsburg, Souter and Stevens, were grouped together. This

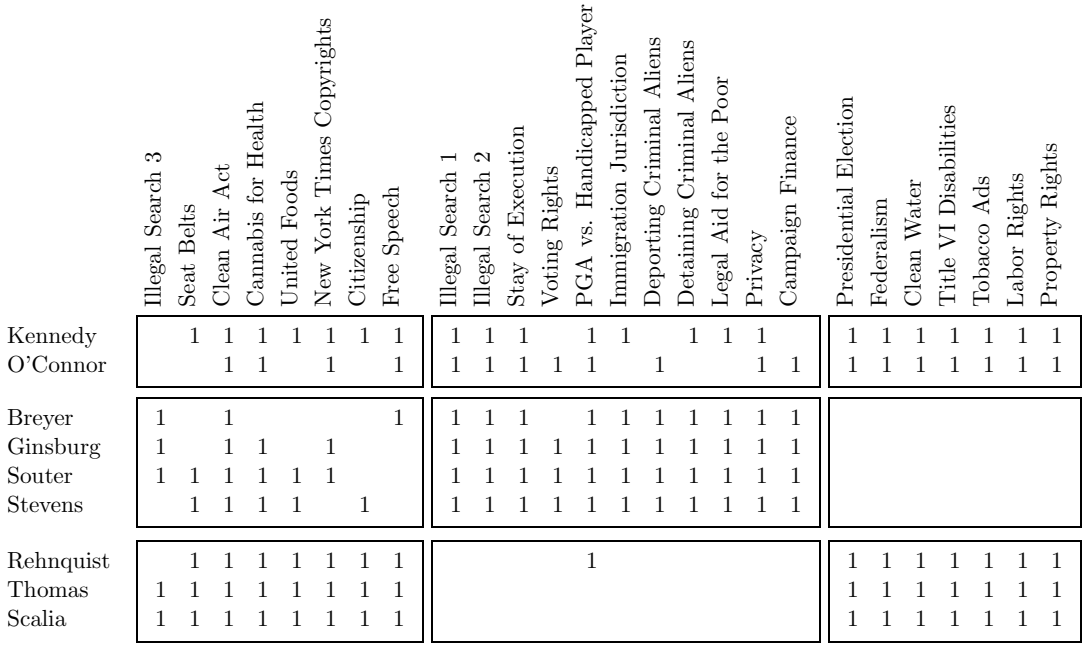


Figure 2.11: Supreme Court voting data set. Solution selected by our compression-based criterion function.

left the more conservative Justices, Rehnquist, Thomas and Scalia, in the last group. This is slightly different from the clustering offered by Doreian et al. (2004a) and Doreian and Fujimito (2003), where they split up the swing voters each into its own singleton group.

The cases were grouped into three clusters, in contrast to the seven clusters of Doreian et al. (2004a). From this blocking we identify almost four perfect complete blocks and two perfect null blocks. The remaining three are 1-covered blocks. From this, we can clearly see where the more conservative Justices differ from the more liberal ones. The swing voters voted with the majority in all three groups of cases; they were the two Justices to dissent the least number of times during the term (Greenhouse, 2001). Again, our algorithm was not given the number of blocks and types *a priori*.

2.5.2 Simulation study

In this study we compare the quality of our compression-based criterion function to that of the standard error-based criterion for direct blockmodeling. The error-based criterion counts the total number of inconsistencies across all blocks.

We compared the two criterion functions on six synthetic data sets of varying sizes, with different image matrices and noise. We assessed the quality of a criterion based on the Rand index (Rand, 1971) of the solution that it selects (the solution with the lowest cost) with respect to the designed solution for each data set. The experiment consisted of running the alternating optimization algorithm five hundred times, each from a different starting point, on each data set for each criterion function. The Rand index of the 500 generated blockmodels, for each criterion function, was computed.

The error-based criterion was provided with the number of clusters in one test and with the image matrix in another test. Our compression-based criterion was not given any such information.

The study revealed that the compression-based criterion is particularly useful when the data contain a combination of structural and regular blocks. After noise was added, the compression-based criterion selected better blockmodels than the error-based even after providing the error-based with the image matrix.

Inconsistencies in regular equivalence are defined as the number of covered rows/columns for null blocks and the number of uncovered rows/columns for 1-covered blocks (Batagelj et al., 1992a). The problem with this definition is that it admits multiple equally well-fitting 1-covered blocks of varying sizes and densities (Doreian et al., 2004a). We tried using this

definition and the Rand indices of the generated blockmodels were not much better than 0.5. One work-around to this problem, which was used by Doreian et al. (2004a), is to use the error-based criterion for structural equivalence, but penalize inconsistencies in null blocks by 100 times more than inconsistencies in complete blocks. This has the effect of finding null blocks with few or no errors, but allowing complete blocks with many 0s (a pseudo regular block).

Thus, we define the error-based criterion as follows.

$$
\begin{aligned}
f_s(B) &= \begin{cases} \#1\text{s} & \text{if } B \text{ is null} \\ \#0\text{s} & \text{if } B \text{ is complete} \end{cases} \\
f_r(B) &= \begin{cases} \#1\text{s} \times 100 & \text{if } B \text{ is null} \\ \#0\text{s} & \text{if } B \text{ is 1-covered} \end{cases} \\
f_m(B) &= \begin{cases} \#1\text{s} \times 100 & \text{if } B \text{ is null} \\ \#\text{uncovered rows and columns} & \text{if } B \text{ is 1-covered} \\ \#0\text{s} & \text{if } B \text{ is complete} \end{cases}
\end{aligned}
\tag{2.9}
$$

where $f_s(B)$ is used if structural equivalence is expected, f_r is used if regular equivalence is expected, and f_m is used if a mix of both structural and regular blocks are expected in the data. Equation 2.9 is compatible (sensitive) to both structural and regular equivalence, that is, it is zero for ideal blocks (Batagelj et al., 1992b; Batagelj, 1997).

The study is divided into three parts. First, we simulated two two-mode binary matrices of different sizes and image matrices, with and without noise, containing a combination of structural and regular (1-covered) blocks. That is, these data contain the three types of blocks discussed in this work: (1) null, (2) 1-covered, and (3) complete blocks. The results are shown in Figure 2.12.

Each plot shows the Rand index for each resulting co-clustering from the 500 runs

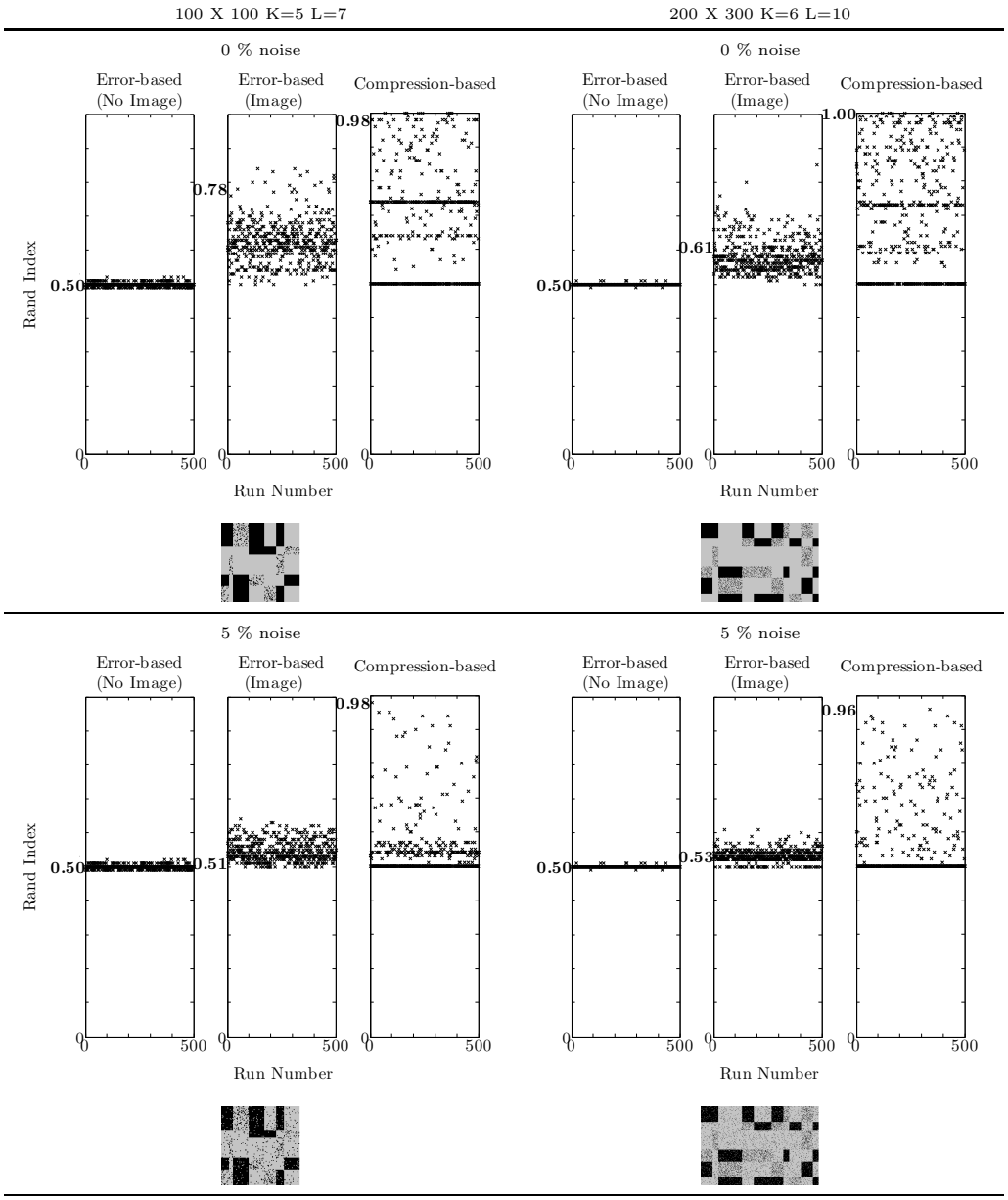


Figure 2.12: Matrices with both structural and regular blocks. The plots show the distribution of the solutions by each criterion for each run, as a function of the Rand index. The $\times$s in the plots denote each run of the algorithm. The Rand index for the solution selected (the lowest cost) by the criterion function is indicated on the y-axis. The matrices show the designed solution used to compute the Rand index.

guided by each criterion. The Rand index is computed with respect to the designed solution, shown in the figure in matrix form. The solution selected by the criterion function, that is, the solution with the smallest value, is indicated by its Rand index along the y-axis. The first row shows the results for the matrices without noise. The data sets with five percent noise added are shown on the second row.

A criterion function well suited for blockmodeling will have two properties: (1) it guides the algorithm toward solutions having a Rand index close to 1 and (2) the lowest costing solution is also the one with the highest Rand index. From Figure 2.12 we see that the compression-based criterion was able to: (1) drive the algorithm toward the designed solution and (2) select the best solution (in some cases it missed by a small margin). On the other hand, the error-based criterion failed to guide the algorithm toward the designed solution, nor did it select the best solution from the candidates by a much larger margin than the compression-based criterion, and this is while the image matrix had been supplied to the error-based criterion. When no image matrix is supplied to the error-based criterion (only the number of clusters), the algorithm gets stuck at Rand index of 0.5, which is no better than randomly guessing a solution. When noise is added, the error-based criterion completely fails, even when the image matrix is provided. Whereas the compression-based criterion still performs very well.

Second, we simulated two matrices with only regular blocks (Figure 2.13). As can be seen, both criterion functions fared poorly. The compression-based criterion decided that the designed solution is not the best. It is very possible that the designed solution is not the best; it is very difficult to visually design solutions for regular equivalence for matrices

of these sizes. The error-based criterion did not fare that much better, even after it was given the image matrix.

The last comparison was done on two structural matrices, shown in Figure 2.14. The error-based criterion proved an excellent choice for structural equivalence blockmodeling. The error-based criterion perfectly selected the designed solution in all four instances. One interesting thing to note is that for the noiseless data, it better guided the algorithm when it was not further constrained by an image matrix.

The compression-based criterion also correctly selected the designed solution when no noise was present. However, it failed to drive the algorithm to the designed solution in many of the restarts. This is due to the criterion having to determine the number of clusters and the block types among structural and regular candidates. When noise is added, the compression-based criterion does worse as it may be forming 1-covered blocks to account for the noise. Also, smaller complete blocks may be formed with fewer errors.

In general, the compression-based criterion should do well at trading off expressibility against over-fitting to the noise. However, when 1-covered blocks are allowed (but do not actually exist), it may find them even when not applicable.

2.5.3 Economic activity in communities

The next test was done on a larger data set. Here we tested our criterion's performance in finding a useful co-clustering of complex real-world data. Does the most efficient complete description of more complex data result in large numbers of clusters and patterns that are difficult to interpret? The data set examined here is indexed by 297 communities

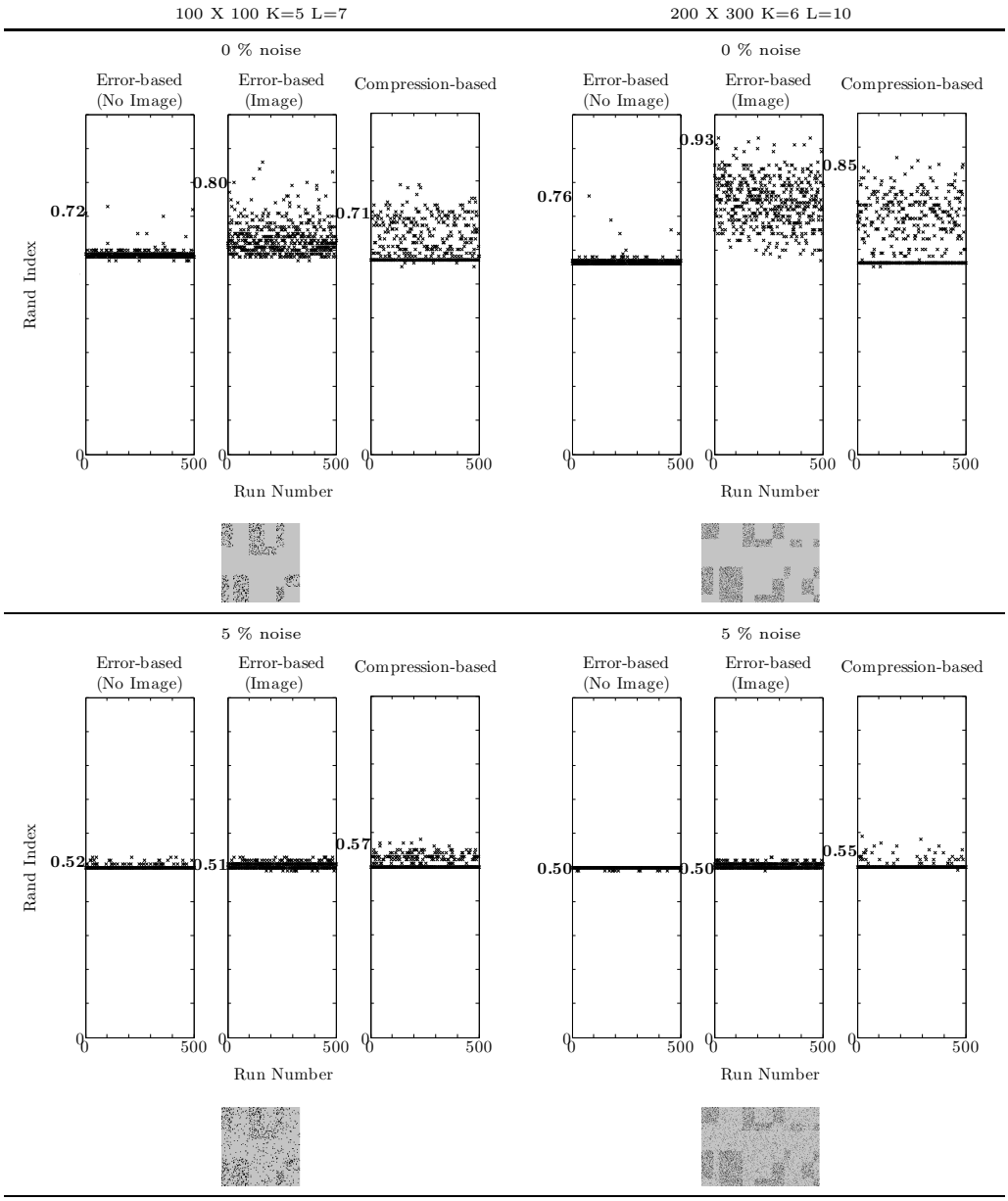


Figure 2.13: Matrices with regular blocks. The plots show the distribution of the solutions by each criterion for each run, as a function of the Rand index. The ×s in the plots denote each run of the algorithm. The Rand index for the solution selected (the lowest cost) by the criterion function is indicated on the y-axis. The matrices show the designed solution used to compute the Rand index.

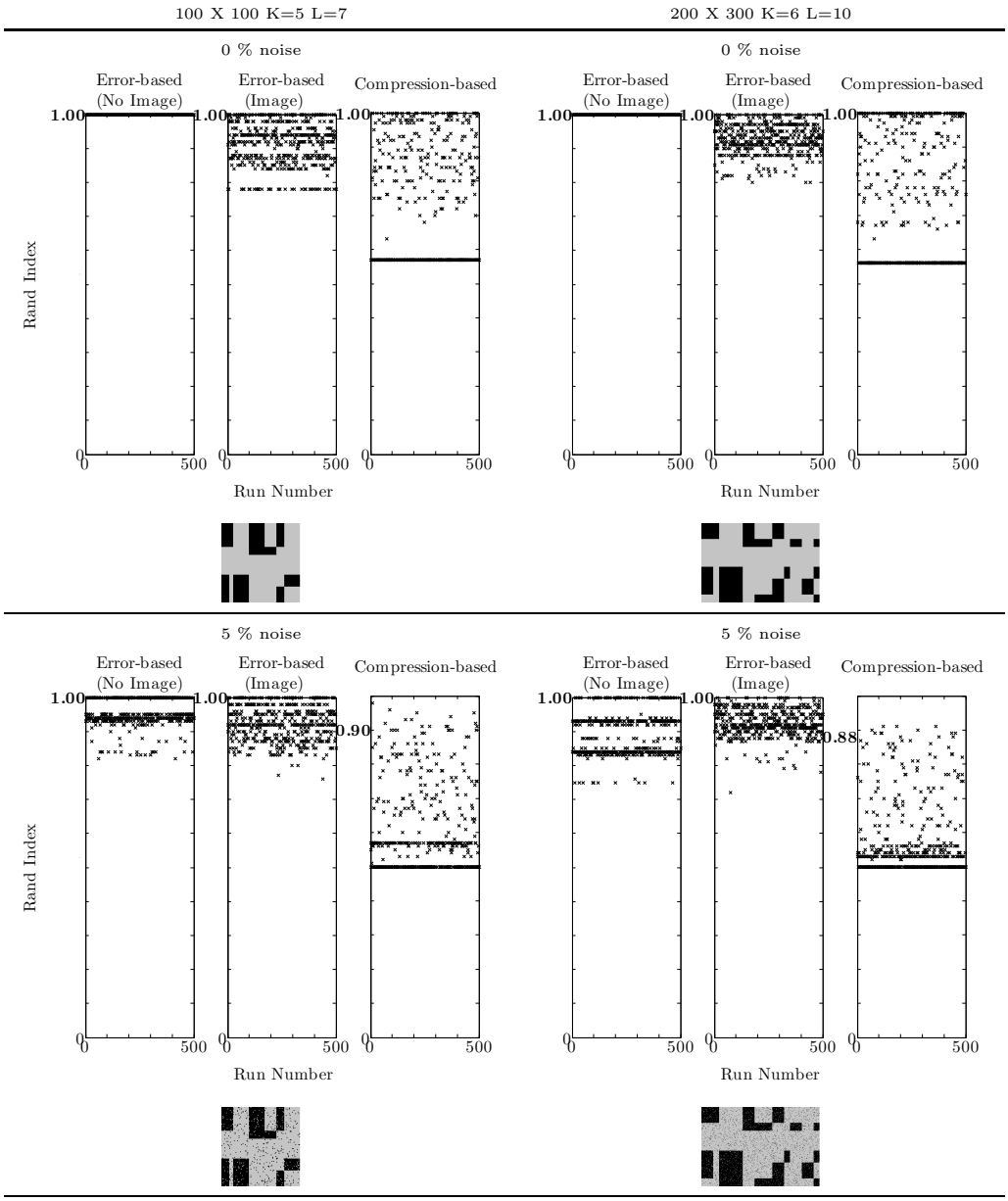


Figure 2.14: Matrices with structural blocks. The plots show the distribution of the solutions by each criterion for each run, as a function of the Rand index. The ×s in the plots denote each run of the algorithm. The Rand index for the solution selected (the lowest cost) by the criterion function is indicated on the y-axis. The matrices show the designed solution used to compute the Rand index.

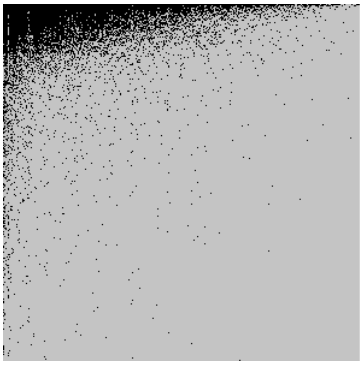


Figure 2.15: New Mexico economic data set after binarization. The rows are indexed by 297 communities and the columns by 296 types of economic activity. Black dots on the figure indicate that a community has at least one business of the corresponding type. The rows and columns in this figure have been rearranged in order of decreasing density.

and 296 economic activities.

The data are the counts of the numbers of organizations performing each of 296 activities as their primary function (as measured by 4-digit NAICS codes) in each of 297 places with organizations in New Mexico in 2004. The data were supplied by InfoUSA, publisher of a widely used business directory. We binarized the data by assigning a 1 to each cell if there were any organizations in the community that performed a particular function. The pre-processed data[5] are shown in Figure 2.15.

Theories of place hierarchies and of organizational communities (Eaton and Lipsey, 1982; Losch, 1954) provide little guidance as to how many "types" of communities we might expect to find or how many "types" of organizations. Furthermore, the nature of the equivalences that might be expected to define the blocks are points of theoretical contention. In the clustering of organizations, there are some reasons to expect that some activities are very likely to occur in the same communities. Some activities are complementary to others

[5]The input to the algorithm is a random permutation of the data.

in chains of production, and there may be efficiencies that result from co-location. However, some other organizational types may be competitive with, or substitutable for others—which would suggest likely regular equivalences. In the clustering of communities, one principle is that of a "central place hierarchy" (Christaller, 1966). This hypothesis suggests that there are sets of more central communities that contain super-sets of activities of less central communities. An alternative idea, however, is that some communities may be functionally specialized and have unique sets of activities that are not common in central places. It may also be that there exists a core set of "keystone" functions (Mills et al., 1993) that must be performed in every community, regardless of size. In short, as with many problems of "affinities" and "correspondences" between two or more modes, theory leads us to expect non-randomness, but provides little *a priori* guidance about numbers of classes in each mode, or the patterns of equivalence defining the blocking.

2.5.4 Compression-based Results and Analysis

We ran the alternating optimization algorithm with our compression-based criterion for 500 randomly generated initial co-clusterings. We then ran the alternating optimization algorithm with the error-based criterion for 500 randomly generated initial co-clusterings, using the number of row and column clusters determined by the best run (that is, the one with lowest cost) of the compression-based criterion. We tried the three error-based criterion functions for each of the three equivalence relation combinations in Equation 2.9, in turn.

Figure 2.16 shows the solution selected by our compression-based criterion. Seven

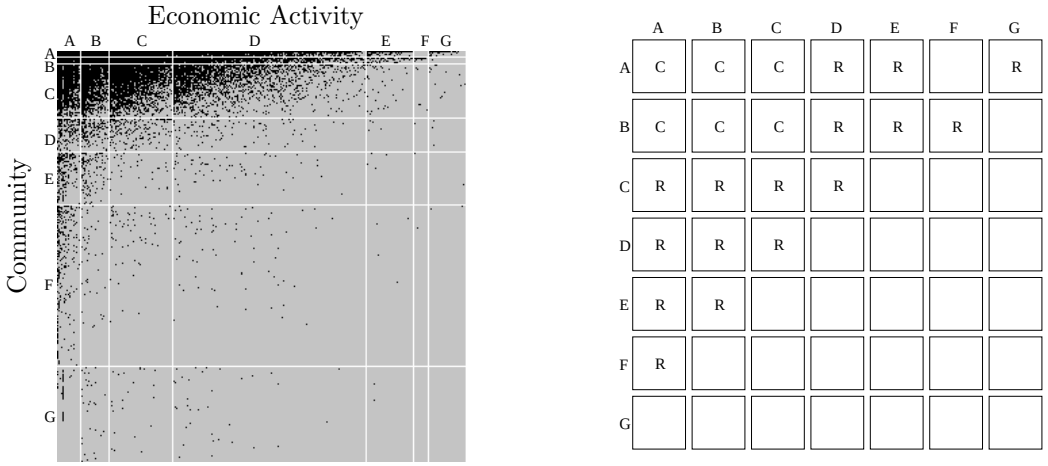


Figure 2.16: New Mexico economic data. Solution selected by our compression-based criterion. Left: The blocked data matrix. Right: The corresponding blockmodel. The letter inside a matrix block indicates the type of block: C for complete, R for regular, and no letter for null blocks.

"types" of communities are suggested, and it is possible to order the types into a hierarchy. Each of the places in community clusters A and B is very likely to contain all of the activities in activities clusters A, B and C. Community clusters A and B differ in the last two economic activity clusters, F and G. The contents of these clusterings are shown in Figures 2.17 and 2.18. The places in clusters A and B comprise eight of the top ten most populated cities in New Mexico (Brinkhoff, 2011). It is reasonable to expect that the more populous cities are more diverse in the types of businesses required to sustain them and thus, must contain many similar business types. Therefore, it is likely that from this economic activity data, clusters separating these large cities from smaller ones will emerge.

Some theories of urban hierarchies suggest that we are likely to observe a single "metropolis." The size distribution of the populations of places in New Mexico is fairly consistent with this idea—Albuquerque is far more populous than the second-ranked city

A
Albuquerque
Las Cruces
Farmington
Hobbs

B
Santa Fe
Roswell
Clovis
Carlsbad

Figure 2.17: Communities found by the compression-based criterion to have many business activities in common: row clusters A and B.

A
Primary activities: (5%)
1129 Other Animal Production
Secondary activities: (24%)
2361 Residential Building Construction
2382 Building Equipment Contractors
2383 Building Finishing Contractors
2389 Other Specialty Trade Contractors
Tertiary activities: (71%)
4842 Specialized Freight Trucking
4911 Postal Service
5617 Services to Buildings and Dwell.
6111 Elementary and Secondary Schools
6244 Child Day Care Services
7139 Other Amusement and Recreation
7212 RV Parks and Recreational Camps
8111 Automotive Repair and Maint.
8121 Personal Care Services
8131 Religious Organizations
9211 Executive, Legislative, and Other
9221 Justice, Public Order, and Safety

B
Primary activities: (5%)
1152 Support Activities for Animal Prod.
Secondary activities: (11%)
2362 Nonresidential Building Construct.
2371 Utility System Construction
Tertiary activities: (84%)
3231 Printing and Related Support
3399 Other Miscellaneous Manufact.
4238 Machinery, Equipment, and Supp.
4249 Miscellaneous Nondurable Goods
4451 Grocery Stores
4452 Specialty Food Stores
4481 Clothing Stores
4483 Jewelry, Luggage, and Leather
4532 Office Supplies, Stationery
4539 Other Miscellaneous Store Retailers
5221 Depository Credit Intermediation
5313 Activities Related to Real Estate
5419 Other Professional, Scientific
6241 Individual and Family Services
7211 Traveler Accommodation
7222 Limited-Service Eating Places

Figure 2.18: Economic activities clustered by our compression-based criterion. The figure shows the business activity types that were in column clusters A and B.

(Las Cruces): 545,852 versus 97,618 (U.S. Census Bureau, 2010). However, the co-clustering suggests that, functionally, Albuquerque can be grouped with a number of other places. All of these places contain all of the cornerstone set of activities. They are also classified together, however, because each contains one (or more) activities from each of three other sets of activities. The appendix contains a complete listing of the community clusters.

The clusterings of economic activities suggest that many activities are part of sets that are mutually absent in many communities (zero blocks). This is consistent with notions of interdependency and complementarity of activities. There are also two relatively large classes of regularly equivalent functions, which would be consistent with theories of competition or substitutability. A look at these details (see the appendix for a complete listing of the economic activity clusters) suggests caution in making a strong interpretation of the regular equivalence clusters. The patterns are messy, and the solution suggests puzzles, as well as patterns.

The simultaneous classification of places in terms of what types of economic functions are performed there, along with the classification of sets of economic functions that commonly occur in the same community does not have a known "correct" solution. The picture that emerges here is broadly consistent with the notions of a central-place hierarchy and "keystone types" of functions. The classes of functions co-occurring in the same or regularly equivalent places (that is, what kinds of economic organizations are present in organizational communities?) involve many regular equivalences—a community may have either this function or that function and still fall in the same class. This is quite reasonable given the detailed level of industrial classification.

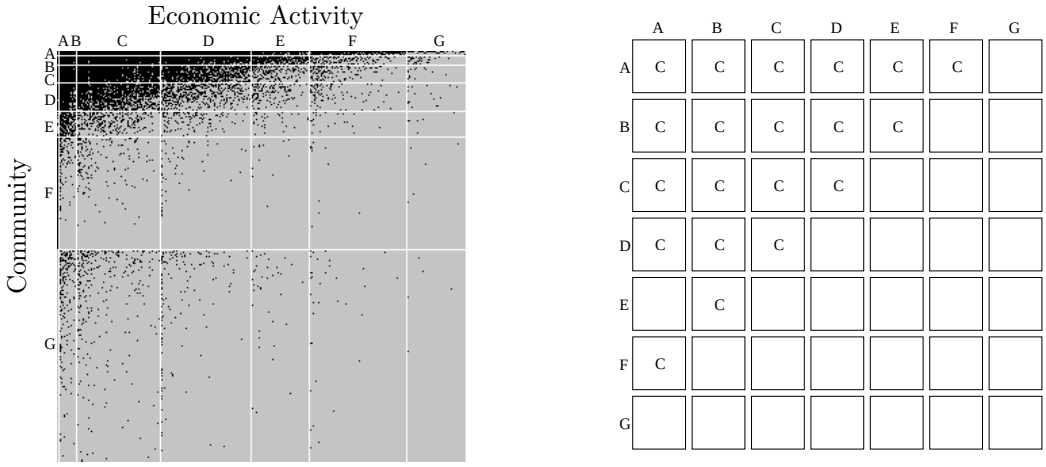


Figure 2.19: New Mexico economic data. Solution selected by the structural error-based criterion, f_s, from Equation 2.9. Left: The blocked data matrix. Right: The corresponding blockmodel. The letter inside a matrix block indicates the type of block: C for complete, and no letter for null blocks.

2.5.5 Error-based Results and Analysis

For the error-based criterion, the purely structural, f_s, gave the best results, so we only show those. Figures 2.19, 2.20, and 2.21 show the results of the error-based criterion. The co-clustering selected by the error-based have similarities and differences to the solution selected by the compression-based criterion. Both coincided in separating the most populous cities from the rest. Given the number of clusters, the error-based criterion placed nine of the top ten (Figure 2.20) in their own two separate clusters. The rest of the community clusters differ. As the data get sparser, the two methods prefer different clusterings. This makes sense as dense blocks are classified as complete by both criterion functions. Whereas, sparse blocks are classified as 1-covered by our compression-based criterion, but null by the error-based criterion. Decidedly null blocks are classified as null by both criterion functions.

A
Albuquerque
Santa Fe
Las Cruces

B
Roswell
Farmington
Hobbs
Clovis
Rio Rancho
Carlsbad

Figure 2.20: Communities clustered by the error-based criterion, f_s. This figure shows row clusters A and B.

A
Primary activities: (0%)
Secondary activities: (0%)
Tertiary activities: (100%)
4911 Postal Service

B
Primary activities: (0%)
Secondary activities: (42%)
2361 Residential Building Construction
2371 Utility System Construction
2382 Building Equipment Contractors
2383 Building Finishing Contractors
2389 Other Specialty Trade Contractors
Tertiary activities: (58%)
8131 Religious Organizations
4451 Grocery Stores
7222 Limited-Service Eating Places
5419 Other Professional, Scientific, and Tech.
8121 Personal Care Services
4539 Other Miscellaneous Store Retailers
8111 Automotive Repair and Maintenance

Figure 2.21: Economic activities clustered by the error-based criterion, f_s. The figure shows the business activity types that were in column clusters A and B.

2.5.6 Automatically choosing the number of clusters

Our compression-based criterion function shows good promise. The criterion selected a solution for the southern women data set that is consistent with previous extensive analyses, e.g., Freeman (2003) and Doreian et al. (2004a). The solution selected for the Supreme Court voting data set correctly identifies the division of ideology among the Justices, democrats, conservatives and swing voters. The solution provided for the New Mexico data set can be explained by socio-economic theories as posited in the previous section. A reasonable question, therefore, is whether other methods might be used to automatically determine the number of clusters. For example, a common object criterion in blockmodeling is minimizing inconsistency with ideal block structure (Brusco and Steinley, 2011; Doreian et al., 2005).

We conducted an experiment to see whether using the number of inconsistencies for a solution is a viable alternative to determine the number of row and column clusters. For example, if we plot the number of errors as a function of the number of row and column clusters we might be able to determine the optimal number of clusters if there is an "elbow" in the graph indicating a change in the percent reduction of errors.

We compared co-clusterings on the economic activities in communities data set for all combinations of row and column cluster counts from 1 to 20 on both ways of the data matrix, by fixing the number of clusters *a priori*. For each row and column cluster combination, we plotted the average of 10 random restarts. The plot is shown in Figure 2.22.

Figure 2.22 needs explaining. First, when the number of both row and column clusters is 1, the number of errors is very small because the single block is best explained

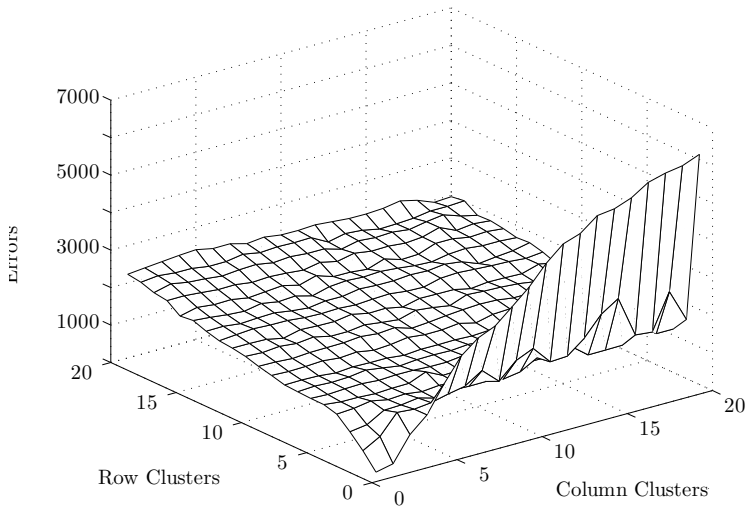


Figure 2.22: Plot of the number of errors (inconsistencies) with the ideal block structure as a function of the number of row and column clusters for the economic activities in communities data set.

by a regular ideal block where each row and column is 1-covered. Second, when the number of row clusters is 1, but the number of column clusters is increased, the number of errors is very high. This is because, in part, there is no clear-cut location where to partition the columns for any number of column clusters when there is only one row cluster (see Figure 2.15). By visual inspection of Figure 2.15, we can see that it is no clear advantage of placing a column partition at one location over another. In addition, the encoding cost of regular 1-covered blocks in Equation 2.5 gets very high as the size of the block increases. The blocks formed when all the rows belong to the same cluster are very large; they span the entire unpartitioned row way of the matrix. So, after the blocks formed by the first couple of column clusters are assigned as regular with very few errors, the rest of the blocks formed by additional column clusters become too costly to be assigned as regular, so they

are assigned as null blocks with many errors.

As soon as the rows are partitioned, smaller blocks can form, toward the top of the matrix, that can be assigned as regular at a reasonable encoding cost, and the larger sparser blocks at the bottom can be assigned as null with fewer errors than before.

From this plot (Figure 2.22) we see that for row clusters of two or more (assigning all rows to one cluster would not give any insight about the data), there is no clear location where there is a change in percent number of errors. The number of errors steadily increases slightly, so there is no indication where to "draw the line". (The reason why the number of errors increases slightly instead of decreasing is that it is very costly to have many regular 1-covered blocks. So, instead of adding regular blocks, null blocks are added. As the number of blocks increases, 1 elements that were previously in a regular block, are now in null blocks, thus increasing the number of errors.)

From this example we see that the number of inconsistencies cannot give an indication as to how many clusters there should be. Moreover, the difficulty is only compounded by the need to select two cut-offs, each of which is dependent on the other.

Chapter 3

Symmetric Clustering of Asymmetric Data

We address the problem of finding a single clustering for both dimensions of a square asymmetric data matrix. Such matrices appear frequently in network analysis, where both the rows and columns represent the same set of actors. We present a framework to impose one clustering on both dimensions of the data matrix by posing the problem as the dual of a constrained co-clustering optimization problem and solving using a subgradient method. Our technique is general to a wide range of co-clustering measures. We employ our method on two real-world data sets, showing that it can effectively discover underlying relationships.

Areas such as in social network analysis have data matrices where both rows and columns represent the same set of objects (for example, actor-actor or organization-organization), that is, they have the same *mode*. In the case of such one-mode data, we would

like to find a single clustering for both rows and columns. For instance in pre-processing for recommender systems, product matching, or general supervised learning, we need a single value for each item. One-mode data may be symmetric, for example, actor-actor relational data where each entry encodes the "coworker" relation. Existing co-clustering methods usually find a single clustering for both rows and columns due to data symmetry. However, other one-mode data are asymmetric, for example, economic country-country data where each entry in the matrix encodes the "exports to" relation or network data representing an asymmetric relationship like "works for."

In sociology and economics, such data have been studied in depth in search of answers to socio-economic questions Smith and White (1992); Moore et al. (2006); Mahutga and Smith (2011). In particular, sociologists and economists often classify countries as members of the core, semi-periphery, or periphery Wallerstein (1972). Thus, given the country-country trade data with the "exports-to" relation, we are interested in identifying each country with one label.

One possible solution is to temporarily make the data matrix symmetric, and then perform two-mode co-clustering on the symmetric matrix. However, this is not guaranteed to yield a clustering that faithfully reflects the data. For example, consider the synthetic two-dimensional one-mode asymmetric binary data in Figure 3.1.

Our one-mode co-clustering method produces a single clustering for both rows and columns (Figure 3.1 (b)). The two-mode co-clustering method prefers a solution where the row and column clusterings are different (Figure 3.1 (c)). The two-mode method on the symmetrized data finds a co-clustering that, although symmetric, is not the best as it

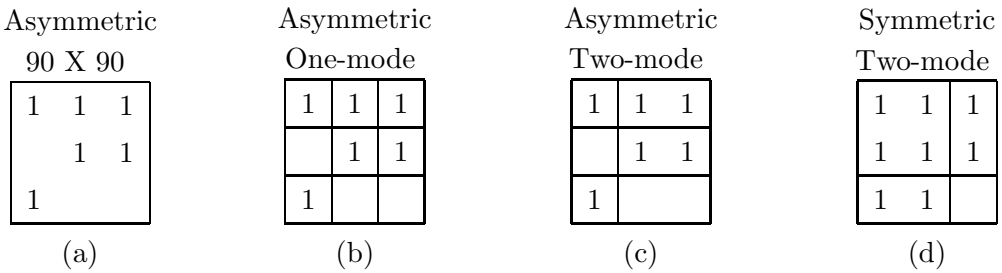


Figure 3.1: Comparison between one- and two-mode co-clustering on one-mode asymmetric data: (a) original data: the 1s represent 30-by-30 sub-matrices of all 1s, (b) our one-mode method result, (c) two-mode method result, (d) two-mode method on symmetrized data result.

groups together the first two thirds of the objects (Figure 3.1 (d)) when in fact they are different in the original data. In making the data symmetric, we have lost information.

Thus, we pursue a method that acts on the asymmetric data directly to produce a symmetric co-clustering. Symmetry here refers to the clustering of the rows and columns, and not to the data. Thus, while row cluster x and column cluster x must contain the same objects, the relationship between elements of row cluster k and column cluster l need not be the same as between row cluster l and column cluster k.

In this chapter, we present a method for finding a symmetric co-clustering of asymmetric one-mode data. We pose the symmetric one-mode co-clustering problem as the dual of a constrained co-clustering optimization problem and demonstrate how standard, existing two-mode co-clustering algorithms can be easily reworked to be one-mode co-clustering algorithms.

We first apply our method to a co-authorship data set, which directly corroborates the hypothetical example in Figure 3.1. Second, we demonstrate results on world trade data and show we can identify countries' economic positions.

3.1 Related Work

Blockmodeling, the dominant technique in social network analysis Doreian et al. (2004b), uses the "relocation" algorithm, which is a hill-climbing local search; each unit is individually tried in every possible cluster. This method has been used for solving the symmetric co-clustering of one-mode data. However, it is impractical as it needs hundreds of restarts before a reasonable solution is obtained (Doreian et al., 2004b). Other methods of blockmodeling that have been tried, such as those by Brusco and Steinley (2009), are used, rather, in confirmatory analysis of *a priori* hypothesized co-clusterings.

Probabilistic blockmodels for one-mode data have also been studied. Fienberg and Wasserman (1981) and Holland et al. (1983) developed pair-dependent stochastic blockmodels for directed graphs. Wasserman and Anderson (1987) and Anderson et al. (1992) built on the p_1 model Holland and Leinhardt (1981) by fitting a separate model to each entity and then grouping the resulting parameters. These algorithms are heuristic. Nowicki and Snijders (2001) proposed a mixture model generalization with Gibbs sampling for Bayesian posterior estimation. It is a probabilistic generalization of the relocation algorithm (sampling, instead of moving, one entity's group at a time). It suffers from similar problems and is not as efficient as the block-optimization methods from co-clustering.

The co-clustering literature has many efficient optimization methods based on block-optimization. However, to this point they operate only on two-mode data. The blockmodeling literature has one-mode and two-mode methods. However, to this point they use the computationally impractical relocation method of moving only one object at a time, and do not automatically find the number of clusters. This paper works with any

underlying optimization method and produces a one-mode version of it. The results we present automatically determine the number of clusters.

3.2 Problem Formulation

To force a one-mode solution (a symmetric co-clustering), we let $K = L$ and recast our problem as

$$\begin{aligned} \underset{\phi,\psi}{\text{minimize}} \quad & F(\phi, \psi) \\ \text{subject to} \quad & \mathbf{1}_k\{\psi(i)\} - \mathbf{1}_k\{\phi(i)\} = 0, \qquad \forall i, k \end{aligned} \tag{3.1}$$

where $F(\phi, \psi)$ is the original co-clustering criterion to optimize and the indicator function $\mathbf{1}_x\{y\}$ is equal to 1 if $y = x$, zero otherwise. Equation 3.1 is the primal problem.

The corresponding Lagrangian is

$$L(\phi, \psi, \mu) = F(\phi, \psi) + \sum_i \mu_{\psi(i),i} - \sum_i \mu_{\phi(i),i}\,, \tag{3.2}$$

where $\mu \in \mathbb{R}^{K \times M}$. The resulting dual is

$$\begin{aligned} \underset{\mu}{\text{maximize}} \quad & q(\mu) \\ \text{where} \quad & q(\mu) = \min_{\phi,\psi} L(\phi, \psi, \mu)\,. \end{aligned} \tag{3.3}$$

3.3 Algorithm

We solve the dual optimization problem of Equation 3.3 by subgradient ascent using the update

$$\mu^{t+1} = \mu^t + \gamma^t g(\mu)\,. \tag{3.4}$$

where γ^t is the step size at time step t, and subgradient

$$g(\mu) = \nabla_\mu|_{\phi=\phi_\mu,\psi=\psi_\mu} L(\phi, \psi, \mu)\,. \tag{3.5}$$

For a given μ, let g_{ki} be the ki-th element:

$$g_{ki} = \left.\frac{\partial L(\phi, \psi, \mu)}{\partial \mu_{ki}}\right|_{\phi=\phi_\mu,\psi=\psi_\mu} = \mathbf{1}_k\{\psi_\mu(i)\} - \mathbf{1}_k\{\phi_\mu(i)\}\ . \tag{3.6}$$

ϕ_μ and ψ_μ denote the optimal co-clustering with respect to the Lagrangian multipliers, μ:

$$(\phi_\mu, \psi_\mu) = \operatorname*{argmin}_{\phi,\psi} L(\phi, \psi, \mu)\,. \tag{3.7}$$

This optimization can be performed by a slightly modified version of the original unconstrained "base" co-clustering algorithm. As mentioned in Section 1.1.2, these algorithms all perform alternating optimizations, switching among z, ϕ, and ψ. The optimization over z is unchanged: The Lagrangian L (Equation 3.2) is the same as F with respect to optimization over z (z only appears inside of F—Equation 1.2—which we have not changed). Therefore, this step can be performed by the base co-clustering algorithm.

The update of ϕ or ψ can be performed again independently for each row or column as the Lagrangian also breaks down into the sum of terms, each involving only one row or column. Therefore the unconstrained updates of Equation 1.3 become

$$\begin{aligned}\phi_\mu(i) &= \operatorname*{argmin}_k \sum_j f(X, z, i, j, k, \psi(j)) - \mu_{k,i} \\ \psi_\mu(j) &= \operatorname*{argmin}_l \sum_i f(X, z, i, j, \phi(i), l) + \mu_{l,j}\ .\end{aligned} \tag{3.8}$$

So, the computation of ϕ_μ and ψ_μ can be performed by the base clustering algorithm with the simple change of adding a penalty or reward to the row-cluster and column-cluster assignments optimization. This addition does not change the overall algorithm.

3.3.1 Step size

Assigning M units into K disjoint clusters is a combinatorial problem, and thus we can do away with line search methods. We determine the step size γ^t in Equation 3.4 as the smallest necessary step size such that μ^{t+1} will cause the base co-clustering algorithm to move one unit to another cluster during its next row or column optimization.

For simplicity, let

$$c_{\psi,i,k} = \sum_j f(X, z, i, j, k, \psi(j)) - \mu_{k,i}$$

and

$$c_{\phi,j,l} = \sum_i f(X, z, i, j, \phi(i), l) + \mu_{l,j},$$

be the targets of optimization in Equation 3.8. We determine γ as the smallest amount by which we can scale g and have an assignment in Equation 3.8 change value:

$$\min\left(\min_{i,k\neq\phi(i)} \frac{c_{\psi,i,k} - c_{\psi,i,\phi(i)}}{g_{k,i} - g_{\phi(i),i}}, \min_{j,l\neq\psi(j)} \frac{c_{\phi,j,l} - c_{\phi,j,\psi(j)}}{g_{\psi(j),j} - g_{l,j}}\right).$$

3.3.2 Optimization

The algorithm proceeds by alternating between two steps: (1) optimizing $\phi_{\mu^t}, \psi_{\mu^t}$ (Equation 3.7) using the base co-clustering algorithm for the current μ^t, and (2) updating μ^t to μ^{t+1} (Equation 3.4). This algorithm is shown in Figure 3.2.

No algorithm is known to exist that guarantees the global minimum to the co-clustering problem (Lagrangian in Equation 3.7) in polynomial time (Anagnostopoulos et al., 2008). Thus, that the base co-clustering algorithm does not guarantee a global minimum is problematic in the subgradient ascent method and is handled by the algorithm

```
   Input: X, K, φ^init, φ^init
   Output: (φ^t, ψ^t)
 1 begin
 2     Set iteration index t = 0, and μ^0 = 0
 3     (φ^0, ψ^0) ← Co-clustering(φ^init, ψ^init, μ^0) // Figure 1.1
 4     repeat
 5         μ^{t+1} ← μ^t + γ^t g(μ^t)
 6         (φ^{t+1}, ψ^{t+1}) ← Co-clustering(φ^t, ψ^t, μ^{t+1}) // Figure 1.1
 7         if L(φ^{t+1}, ψ^{t+1}, μ^{t+1}) ≤ L(φ^t, ψ^t, μ^t) then
 8             if L(φ^{t+1}, ψ^{t+1}, μ^t) < L(φ^t, ψ^t, μ^t) then
 9                 μ^{t+1} ← μ^t
10             else
11                 return (φ^t, ψ^t)
12         t ← t + 1
13     until no increase in L(φ^t, ψ^t, μ^t)
```

Figure 3.2: Symmetric algorithm.

in Figure 3.2 as follows. If the dual function does not increase (Line 7 of Algorithm 3.2), the algorithm checks to see if the newly found co-clustering is better (smaller) than the previous iteration's (Line 8). If so, the algorithm replaces the previous solution with the new one and continues by setting the Lagrange multipliers to the previous ones (Line 9).

The optimization over (ϕ, ψ) is performed by the base co-clustering algorithm, using the updates in Equation 3.8.

3.4 Experimentation

To test our framework, we need a base co-clustering algorithm. Several possible methods are described in Section 1.1.2. We chose the method of Chakrabarti et al. (2004) as our base co-clustering algorithm for its ability to automatically determine the number of

clusters. We modified their method slightly as described below.

3.4.1 A Criterion Function $F(\phi, \psi)$

Chakrabarti et al. (2004) (see Section 1.1.2) define F to be the description length of a co-clustered binary data matrix, applying the minimum description length (MDL) principle (Rissanen, 1978). They first encode the clustering of M units into K clusters by encoding the row permutations plus the number of row clusters and the size of each row cluster; similarly, they encode the column clustering. As we have only one-mode, we modified this to $M \lg K$ bits, that is, for each of the M units, we encode the cluster to which it is assigned in $\lg K$ bits. Then they transmit the block descriptions as their frequencies encoded as the number of ones in each block. Letting n_{kl} be the number of elements in the joint cluster (block) $X(\phi^{-1}(k), \psi^{-1}(l))$, the model length becomes

$$M \lg K + \sum_{k,l} \lg(n_{kl} + 1)\,.$$

Inside the sum, a 1 is added to n_{kl} to account for blocks with no 1s (null blocks). Other information needed in an actual transmission of the data, such as the size of the matrix, is constant across co-clusterings and hence does not factor into the optimization.

Finally, the data are sent using the optimal Hoffman code (Huffman, 1952), given the block frequencies. That is, if z_{kl} is the frequency of ones in joint cluster $X(\phi^{-1}(k), \psi^{-1}(l))$, each element in this block is encoded using $-\lg z_{kl}$ bits if it is a one and $-\lg(1 - z_{kl})$ bits if it is a zero. This results in the total cost of

$$F(\phi,\psi) = \min_{z\in\{0,1\}^{K\times M}} M\lg(K) + \sum_{kl}\lg(n_{kl}+1) \\ - \sum_{ij}\left(x_{ij}\lg z_{\phi(i)\psi(j)} + (1-x_{ij})\lg(1-z_{\phi(i)\psi(j)})\right),$$

alluded to in Section 1.1.2.

3.4.2 Implementation

We adapted the method in Chakrabarti et al. (2004) to work with our framework by making two simple modifications. The first change adds the symmetry penalties (Equation 3.8) to the row and column optimization functions, respectively.

The other change involves the search over the number of clusters K. In their work, starting from $K, L = 1$ they try to increment K and L in alternating steps. They first split the row cluster k with the highest per-row cost to construct the initial co-clustering (ϕ_0^{K+1}, ψ^L) on which to run their alternating optimization algorithm, and then increment the columns in a similar fashion. Since our goal is to have a single clustering for both rows and columns, we maintain $L = K$. We allow the row and column clusterings to be split independently, but run the symmetric co-clustering algorithm on the jointly incremented co-clustering $(\phi_0^{K+1}, \psi_0^{K+1})$.

Finally, it is possible that our algorithm will get stuck in a local optimum that does not satisfy the symmetry constraint, in which case it is re-started from an initial random co-clustering. These restarts occurred in roughly 10% of our experimental runs. The algorithm to search over K is shown in Figure 3.3.

Algorithm: KSearch

Input: X

Output: (ϕ^K, ψ^K)

begin
  Set $K = 1$ and compute $L(\phi^1, \psi^1)$ by Equation 3.2
  repeat
    Increment $K = K + 1$ by splitting most costly cluster
    $(\phi^K, \psi^K) \leftarrow \textsc{Symmetric}(\phi_0^K, \psi_0^K)$
    if $\phi^K \neq \psi^K$ **then**
      Repeat Symmetric with $(\phi^K_{\text{random}}, \psi^K_{\text{random}})$
  until *no decrease in* $L(\phi^K, \psi^K)$

Figure 3.3: KSearch algorithm.

3.4.3 Datasets

For our experiments, we used two asymmetric one-mode data sets. The first is a co-authorship data set that encodes author and co-author relation, obtained from Arnetminer.org (Tang et al., 2009). A 1 in the matrix indicates the row author is the first author of a paper co-authored by the column author. The data set consists of 224 authors, each labeled with one of three topics indicating the author's main area of publication: Data Mining, Bayesian Networks, and Machine Learning. These labels are not supplied to the algorithm.

The second data set is world import and export data compiled by the National Bureau of Economic Research (Feenstra et al., 2005). It consists of traded amounts between 203 countries for various commodities, grouped by Standard International Trade Classification (SITC) code (Dep, 2006), spanning the years 1962–2000. From this, we extracted three data sets covering the years 1990–1999: (1) SITC 0011 (live bovine animals), (2) SITC code 78 (road vehicles), and (3) SITC code 6672 (diamonds). We aggregated the traded amounts

across years and binarized the data by converting non-zero entries into ones to obtain three 203×203 binary data matrices, one for each commodity. A matrix element x_{ij} is 1 if and only if country i exported to country j during 1990–1999.

3.4.4 Method

We compare our one-mode method to three other methods: (1) applying two-mode clustering on the asymmetric data, (2) the two-step process of first symmetrizing the data and then applying the two-mode method and (3) the two-step process of using the two-mode method, then transforming the result to be symmetric by defining clusters in terms of the row-column assignment pairs given by the algorithm. That is, all objects assigned to row cluster a and column cluster b were placed in a new cluster labeled a-b. We call (2) and (3) "pre-process one-mode" and "post-process one-mode", respectively. The post-process method clusters an object with all other objects that share the same row- and column-clusterings from the two-mode method. Method (1), two-mode clustering, is the only one that produces two identities for each object.

Within the co-authorship data set the quality of a co-clustering can be measured by the homogeneity of the clusters with respect to the topics. The quality of the clustering results on world economic trade data can be based on effective discovery of known trade relationships and country categorizations.

3.4.5 Results

We ran each of the four methods on the two real-world data sets.

Our One-mode	**Two-mode**	**Pre-process**	**Post-process**
Cluster 0 (12)	**1st Clstr 0** (19)	**Cluster 0** (20)	**Cluster 0** (1)
0% Dat.Mng.	0% Dat.Mng.	0% Dat.Mng.	0% Dat.Mng.
100% Bayes Net	100% Bayes Net	100% Bayes Net	100% Bayes Net
0% Mach.Lrn.	0% Mach.Lrn.	0% Mach.Lrn.	0% Mach.Lrn.
Cluster 1 (8)	**1st Clstr 1** (107)	**Cluster 1** (75)	**Cluster 1** (15)
0% Dat.Mng.	40% Dat.Mng.	23% Dat.Mng.	0% Dat.Mng.
100% Bayes Net	5% Bayes Net	0% Bayes Net	100% Bayes Net
0% Mach.Lrn.	55% Mach.Lrn.	77% Mach.Lrn.	0% Mach.Lrn.
Cluster 2 (52)	**1st Clstr 2** (98)	**Cluster 2** (129)	**Cluster 2** (1)
0% Dat.Mng.	33% Dat.Mng.	45% Dat.Mng.	0% Dat.Mng.
0% Bayes Net	10% Bayes Net	11% Bayes Net	100% Bayes Net
100% Mach.Lrn.	57% Mach.Lrn.	44% Mach.Lrn.	0% Mach.Lrn.
Cluster 3 (58)	**2nd Clstr 0** (17)	**Cluster 6** (30)	**Cluster 3** (4)
0% Dat.Mng.	0% Dat.Mng.	27% Dat.Mng.	0% Dat.Mng.
2% Bayes Net	100% Bayes Net	3% Bayes Net	100% Bayes Net
98% Mach.Lrn.	0% Mach.Lrn.	70% Mach.Lrn.	0% Mach.Lrn.
Cluster 4 (94)	**2nd Clstr 1** (33)	**Cluster 7** (41)	**Cluster 4** (21)
80% Dat.Mng.	42% Dat.Mng.	27% Dat.Mng.	48% Dat.Mng.
14% Bayes Net	0% Bayes Net	5% Bayes Net	0% Bayes Net
6% Mach.Lrn.	58% Mach.Lrn.	68% Mach.Lrn.	52% Mach.Lrn.
2nd Clstr 3 (103)	**2nd Clstr 2** (71)	**Cluster 8** (55)	**Cluster 5** (12)
41% Dat.Mng.	27% Data Mng.	46% Dat.Mng.	33% Dat.Mng.
13% Bayes Net	4% Bayes Net	5% Bayes Net	0% Bayes Net
46% Mach.Lrn.	69% Mach.Lrn.	49% Mach.Lrn.	67% Mach.Lrn.

Figure 3.4: Co-authorship data set clustering summary. Values in parentheses indicate cluster size. Post-process cluster 9 is omitted; it is similar to cluster 8.

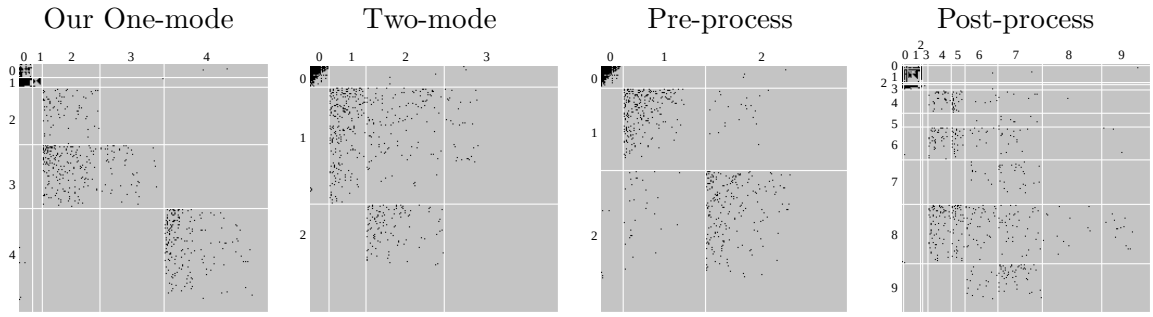


Figure 3.5: Co-authorship results among 224 authors publishing in three different topics. Black indicates an author from the row is a first author in a paper with the author in the column. White lines denote cluster splits.

Co-authorship Dataset. Figure 3.4 measures the homogeneity of each cluster. A graphical representation of the results is displayed in Figure 3.5. For each clustering, we have placed the rows and columns into a "canonical" form: Clusters are ordered by their densities. Individual rows and columns are similarly sorted within clusters.

As expected, the methods produced symmetric and asymmetric clusterings according to their objective functions. The one-mode method produced a symmetric co-clustering with equal clusterings for both rows and columns. The two-mode method on the asymmetric data produced unequal clusterings. Both the pre- and post-process methods produced equal clusterings for the rows and columns. Out of the four methods, the proposed one-mode co-clustering better separated the authors by their topics. Almost four out of five clusters were perfectly homogeneous with only the fifth cluster having a mixture. This simply reflects information in the data encoding collaboration among authors from closely related topics. However, note that all authors labeled with Data Mining were placed in this last cluster. Thus, we can say that clusters 0 and 1 represent Bayes Nets, clusters 2 and 3, Machine Learning, and cluster 4, Data Mining.

The other methods did not fare as well. The two-mode co-clustering on the asymmetric data has all but the smallest cluster mixed. The pre-process method produced three clusters of which only the smallest one was homogeneous. The post-process method suffers cluster count explosion resulting in split and mixed clusters.

Our One-mode	**Two-mode**	**Pre-process**	**Post-process**
Cluster 0	**Exp.Cluster 0**	**Exp.Cluster 0**	**Cluster 0**
Denmark	Australia	Austria	Germany
Germany	Canada	Denmark	Netherlands
Netherlands	Denmark	FmGermanFR	
	Fm German FR	Germany	**Cluster 1**
Cluster 1	Germany	Hungary	Poland
Australia	Netherlands	Italy	
Canada	USA	Netherlands	**Cluster 2**
New Zealand		Poland	Italy
USA	**Exp.Cluster 1**	Spain	Spain
	Austria	Turkey	
Cluster 2	Belgium-Lux		**Cluster 3**
Austria	Bulgaria	**Imp.Cluster 1**	Australia
Belgium-Lux	Czechoslovak	Belgium-Lux	Denmark
Bulgaria	Czech Rep	BosniaHerzg	USA
Czechoslovak	FmGermanFR	Bulgaria	
Czech Rep	Hungary	Czechoslovak	**Cluster 4**
FmGermanFR	Ireland	Czech Rep	FmGermanFR
Hungary	Italy	Egypt	
Ireland	Poland	Greece	**Cluster 5**
+7 more	+5 more	+12 more	Turkey

Figure 3.6: High-activity clusters for live bovine animals.

World Trade Datasets. The countries in the high-activity clusters are shown in Figures 3.6, 3.7, and 3.8, sorted alphabetically. The results of these clustering methods are shown in Figure 3.9. The graphs are sorted in the same manner as the co-authorship graphs.

Our one-mode method resulted in only slightly higher cluster counts when compared to the two-mode result, indicating that the underlying country import and export

Our One-mode	Two-mode	Pre-process	Post-process
Cluster 0	**Exp.Cluster 0**	**Exp.Cluster 0**	**Cluster 0**
Belgium-Lux	Belgium-Lux	Belgium-Lux	Italy
China	China	China	
Germany	Germany	Germany	**Cluster 1**
Italy	India	India	Germany
Japan	Italy	Italy	Netherlands
Korea Rep.	Japan	Japan	UK
Netherlands	Korea Rep.	Korea Rep.	
Spain	Netherlands	Netherlands	**Cluster 2**
Sweden	Spain	Spain	Belgium-Lux
UK	Sweden	Sweden	China
USA	UK	UK	Japan
	USA	USA	Korea Rep.
Cluster 1			Spain
Brazil	**Imp.Cluster 0**	**Imp.Cluster 1**	Sweden
Canada	Italy	Austria	USA
FmGermanFR		Brazil	
India	**Imp.Cluster 1**	Canada	**Cluster 3**
Thailand	Germany	Czech Rep	India
	Netherlands	Denmark	
	UK	+9 more	

Figure 3.7: High-activity clusters for road vehicles.

identities are significantly correlated. Across the world trade data sets our method produced well grouped clusters that easily correlate to real world reasoning.

Geographical affinity is a trade trait that transcends trade commodity as seen in our various data sets. Our method produced clusters representing regionalized partners. In the live bovine data set our algorithm produced clusters corresponding to European countries (0 and 2) and major Pacific Rim countries (1) as seen in Figure 3.6. The other methods did not fair so well. The two-mode result shows some regional ties, such as export and import cluster 1 having differing European countries. Beyond these, the other clusters are less easily mapped to a region. The same is seen in the pre-process method. In the

Our One-mode	Two-mode	Pre-process	Post-process
Cluster 0	**Exp.Cluster 0**	**Exp.Cluster 0**	**Cluster 0**
Belgium-Lux	Belgium-Lux	Belgium-Lux	Belgium-Lux
		India	
Cluster 1	**Imp.Cluster 0**	Switz.Liecht	**Cluster 1**
Germany	Belgium-Lux	UK	USA
India	USA	USA	
Israel		Thailand	**Cluster 2**
Switz.Liecht	**Imp.Cluster 1**		Switz.Liecht
Thailand	Switz.Liecht		Thailand
UK	Thailand		UK
USA	UK		

Figure 3.8: High-activity clusters for diamonds.

post-process method an explosion of clusters divides up many of the discernible geographic regions across many clusters making them difficult to interpret.

Commodity distribution and consumption as well as trade dominance can be identified by clustering, as seen in the road vehicle and diamond data sets. Within the auto industry several countries dominate trade. Maxton and Wormald (2004) identify Germany, Japan and USA as the "core" of the industry. Additionally, South Korea (Korea Republic), Spain, and UK have been established as transplant countries (Biggart and Guillen, 1999; Maxton and Wormald, 2004). In Figure 3.7 our method establishes cluster 0 containing the core as well as other countries with large auto production facilities. Additionally, our method clustered countries of similar consumption. The two-mode and pre-process methods clustered the core but the countries outside the core are less easily mapped to an identity. The post-process method has a significantly higher cluster count, splitting the core across several clusters. Similar trade dominance is seen in the diamonds data set. Spar (2006) and Gupta et al. (2010) describe the economic impact of De Beers as a cartel within the

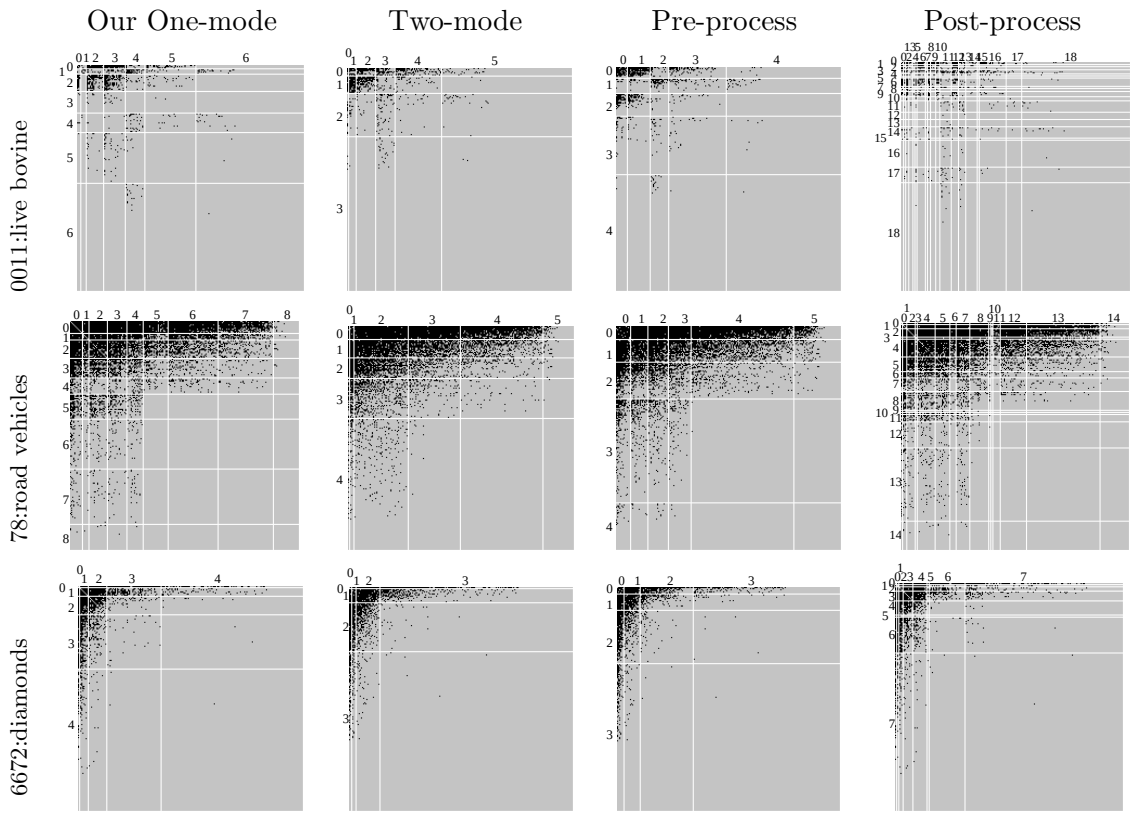


Figure 3.9: World trade results among 203 countries. Three data sets are shown, each covering the 1990s decade. Black indicates an export from the row to the column. White lines denote cluster splits.

diamond industry; De Beers is a registered company in Luxembourg. In Figure 3.8, our method, as well as the two-mode and post-process methods, create a one-country cluster displaying this dominance. The pre-process method fails at this. Our method also creates cluster 1 for countries of similar trade habit. The other methods split the countries into many clusters that are not easily identified as a unit.

A major pitfall of the pre-process method is the loss of information. For example, consider the clustering of the road vehicle and diamond data sets. In each, peripheral countries are added to the core cluster skewing its identity. India is added to the core production

countries found in cluster 0 of our method for road vehicles. Several significant importers are added to the cartel cluster that contains Belgium-Luxembourg in the diamonds data set. Our method does not fall victim to this because data asymmetry is considered while clustering.

Results on the relocation algorithm were not obtained due to it taking too long (too many restarts) on these problems.

Experimental Conclusions. Well-grouped single identities were only discovered when we impose the symmetric clustering constraint; two-mode co-clustering does not naturally produce a single identity on asymmetric data. The pre-process method produced results after making the data symmetric, resulting in mixed clusters as the data asymmetry is not considered while clustering. The post-process method continually fell victim to cluster count explosion. All these hindrances created results that are difficult to interpret. Our method created singular identities without succumbing to these pitfalls.

Chapter 4

Clustering ICU Data using Measurement Timings and Values

We apply co-clustering to automatically discover patient clusters with discernible physiologic patterns and prognostic significance from physiologic data. How to transform multi-dimensional temporal data, such as physiologic measurements from the intensive care unit (ICU), to fixed-length feature vectors for clustering is not obvious. For this, we learn a piecewise-constant conditional intensity model of the data, capturing measurement values and timing dependencies, and extract Fisher-information features for each patient. We demonstrate our method on a real pediatric intensive care unit (PICU) data set of over 10,000 patients, comparing it with other clustering methods. As an advantage, our method can handle a mixture of data types (numeric, categorical and binary) so that in addition to vital signs, we can use binary intervention and drug category data.

4.1 Introduction

Human physiology is complex and until (or unless) all of its pathways are mapped, the health care field remains a data driven one. Physicians place patients into one of a set of predefined categories or diagnoses (clusters of patients with similar characteristics) from which to extrapolate a prognosis. This paradigm is only as good as the defined clusters. Humans cannot properly process all the information contained in these data, and the categories that have been defined do not fit every patient. We seek an automated method that will find, from physiologic data, patient clusters with discernible physiologic patterns and prognostic significance.

Currently, intensive care units (ICUs) are the most instrumented areas in health care. Regular and consistent monitoring and recording of vital signs, administered drugs and interventions give a relatively data-rich picture of the patients' physiologies. We expect increased sensing and recording in other medical venues in the near future, but concentrate on ICU data here to determine what will be possible more broadly in electronic health records (EHRs).

In this work we address the following challenges posed by the task of clustering ICU data. It is not obvious how to compare data points consisting on rich, temporal data. Time window-discretization and averaging obscures temporal detail. The timings of the measurements carry information of the state of the patient; critically ill patients require more attention. By averaging the measurements in equal-sized time windows, the timing variations are lost. Additionally, physiologic data are not "missing at random." The absence of a measurement is indicative of a patient's state.

To alleviate these problems, we build a generative model of the physiology and the measurement process, thus bypassing the "missing at random" assumption: The timing of measurements is part of the model. This model also allows the measurements to be highly irregularly spaced.

Normally, the input to clustering algorithms is a matrix, where each row describes a patient as a fixed-length feature vector. Conversion from multiple irregularly sampled time series to such a vector is not obvious. We address this issue by extracting the implied features of the Fisher kernel for each patient, with respect to the generative model.

We address the problem of automatically finding the number of clusters by defining a compression-based cost function for real-valued data that applies the MDL principle.

We tested our method on a data set from the Pediatric Intensive Care Unit (PICU) at Children's Hospital Los Angeles, consisting of over 10,000 patient episodes collected over ten years. The results show clusters with discernible physiologic patterns and prognostic significance.

4.2 Related Work

Recent work has attempted to utilize the multi-dimensional, temporal data captured in EHRs to cluster patients. Marlin et al. (2012) use a mixture model with an empirical prior distribution that encourages a degree of smoothness over time for each variable's measurements. This model depends on the discretization of time and the assumption that data are "missing at random." Lehman et al. (2008) use a Gaussian mixture model on manually defined features meant to capture temporal information, such as gradients and

trends. While showing success in classification tasks, this method relies on the manual definition of features.

Doshi-Velez et al. (2014) apply hierarchical clustering to manually designed features to investigate the patterns of co-occurrence of medical comorbidities in autism spectrum disorders. Paul and Hoque (2010) present a k-means method that can handle both continuous and categorical data. Their method, however, relies on manually defined features. Chignell et al. (2013) worked on clustering adult ICU patients using data from drugs, labs, vitals, demographics, International Statistical Classification of Diseases and Related Health Problems (ICD) codes and ICU stay tables. They also relied on manually designed features.

Tamang and Parsons (2011) take a similar approach to ours on how to deal with irregularly sampled, variable length temporal data by using a generative model as a way of converting them to fixed-length feature vectors. They learn a separate 3-state (stable, moderate, or unstable) hidden Markov model (HMM) for each patient and apply spectral clustering on the resulting feature vectors formed from each HMM's learned parameters: the initial state probabilities, the transition probabilities and the emission probabilities. The authors attempt this method on binary temporal data of glucose tests (whether a test was ordered or not). Two drawbacks of using HMMs are: (1) the states must be known a priori and (2) high-frequency data may be required to accurately learn the model.

James and Hastie (2001) present a reduced rank mixed effects model that was used for classifying medical time-series data. Their method is an extension of linear discriminant analysis (LDA) to non-linear data sets based on spline functions—functional linear discrim-

inant analysis (FLDA). This method can be used on sparse data, however, it assumes the data to be missing at random.

Other work has focused on the problem of unsupervised discovery of meaningful patterns and features from data. Lin and Li (2009) convert continuous values to discrete symbols and then build "bag-of-words" representations of each physiologic variable. While this approach captures certain structure of physiological time series, it disregards temporal order information. Saria et al. (2010) propose a model, from which features can be constructed to use in other tasks, that directly models the heterogeneous, temporal nature of physiologic time series using a switching latent "topic" model similar to those used in natural language precessing. However, this method relies on the availability of complete, high-frequency time series data, which in most cases is not available: Normally, medical data available in EHRs are uncertain and sparsely sampled (see Section 4.3).

There exists a large body of work on the related subject of analyzing high-frequency time series data. See Esling and Agon (2012) for a survey. The methods developed in that body of work require complete, high-frequency time series data and are inappropriate for the irregularly sampled, sparse temporal data in EHRs.

Lastly, a large body of work exists on using snapshots of physiologic measurements, such as heart rate and blood pressure, to characterize the severity of critical illness in the pediatric critical care setting. They focused mostly on the creation of *severity of illness* (SOI) scores, such as the Pediatric Risk of Mortality III (PRISM III) Pollack et al. (1996) and others. These score were not designed for helping diagnose individual patients, but to benchmark PICU performance.

4.3 Electronic Health Records

EHR data are a collection of electronic health information about a population in a per-patient format. In general, EHR data for each patient may include demographics, medical history, physiologic data, and billing information. For the purpose of this work, we are interested only in physiologic data pertinent to a patient's stay (episode) in the ICU. Specifically, we consider measurements of vital signs, laboratory test results, drugs administered, and interventions performed on the patient while in the PICU.

These data pose several challenges for statistical analysis. First, they are incomplete. Not all patients have the same interventions performed or lab tests done. The data produced depend on the needs of each individual patient. This constitutes a potential source of non-random missing data. Measurements and other interventions are performed at irregular intervals, depending on the patient's need; again, providing another source of missing data. These missing data between intervals are also not missing at random: The timings of measurements and other interventions can be indicative of the patient's state.

Disease progression are not aligned in the data. Very likely, patients arrive at the PICU at different stages in their illnesses. This makes comparing patients based on their illnesses non-trivial. For example, comparing "heart rate in hour 2" across patients might not be meaningful. Finally, the data are subject to various forms of uncertainty, such as errors produced during manual entry by care givers and intrinsic noise in measuring devices and instrument malfunction.

4.4 Method

Our method is a three step procedure. In the first step, we build a conjoint piece-wise constant conditional intensity model (C-PCIM) of the time dependencies in the temporal data (Parikh et al., 2012). By learning the measurement timings, we bypass the "missing at random" assumption; the patient state information encoded in the measurement timings is part of the model. Additionally, with this model there is no time discretization.

Converting irregularly-shaped time series data into fixed-length feature vectors for clustering is handled in the second step. We apply a feature extraction mechanism based on the Fisher information kernel (Jaakkola and Haussler, 1998). With a fixed-length feature vector for each patient, a feature matrix is fed to a co-clustering algorithm, in step three, that uses an MDL-based cost function to automatically determine the number of clusters.

4.4.1 C-PCIM

C-PCIMs (Parikh et al., 2012) are a class of marked point processes [1] where the conditional intensity function is a piecewise constant function of time and history, taking one of a finite number of values. For a given event sequence $y = \{(t_i, l_i)\}_{i=1}^{n}$ with $0 < t_1 < \cdots < t_n$, where $t_i \in [0, \infty)$ is the time of the ith event of type l_i, drawn from a finite set of events $\mathcal{L}$, the history of y at time t is the subsequence $h(t, y) = \{(t_i, l_i) \,|\, (t_i, l_i) \in y, t_i < t\}$. In contrast to its predecessor, PCIM, a C-PCIM shares a single conditional intensity function, $\lambda(l, t, h_t)$, among all event types $l \in \mathcal{L}$. Examples of event types in our data set are "blood pressure was measured" and "the patient was intubated." As a single tree handles all event

[1] A marked point process (see Appendix B) is a point process that distinguishes between event types: Each event is identified (or marked) as being of a specific type.

types, the data in our experiments are first normalized (see Section 4.5).

For any event type l, time t and history h_t, the model gives the instantaneous rate at which it could occur, $\lambda(l, t, h)$. The model is built as a decision tree, representing the conditional intensity function, where each leaf represents a state with a resulting rate.

Given a tree with leaves Σ, the likelihood of an event sequence y is

$$p(y|S, \Theta) = \prod_{s \in \Sigma} \lambda_s^{c_s(y)} e^{-\lambda_s d_s(y)}$$

where $d_s(y)$ and $c_s(y)$ are sufficient statistics of the data: $d_s(y)$ is the total duration spent in state s and $c_s(y)$ is the number of times an event occurs in y when the state function maps to leaf s.

The decision tree is built by greedily choosing the questions, from a given set by the user, that maximize the Bayesian score. More details can be found in Parikh et al. (2012).

Extended C-PCIM

In addition to the timings of the measurements, their values contain important information about the patient's state, for example, elevated blood pressure. We extend the C-PCIM to capture value information available in EHRs. For each state s in the model (leaf in the tree), we add a Gaussian distribution over the values that variables can take while in that state. Including a product of Gaussian distributions to the original product of exponential distributions gives the extended model's likelihood function:

$$p(y|S, \Theta) = \prod_{s \in \Sigma} \Lambda_s \ \lambda_s^{c_s(y)} e^{-\lambda_s d_s(y)}\,, \tag{4.1}$$

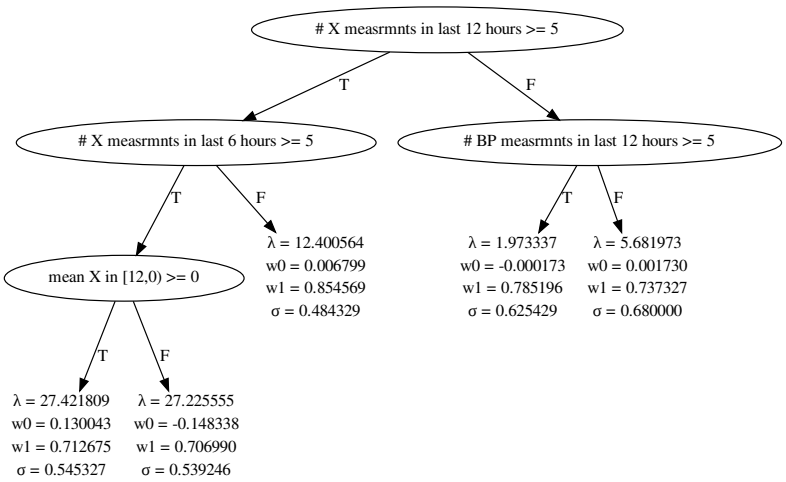


Figure 4.1: Example extended C-PCIM decision tree. The inner nodes represent the various tests selected by the learning algorithm. The parameters of the model are at the leaves. The X is a placeholder for an event type.

where

$$\Lambda_s = \left(\frac{1}{\sigma_s\sqrt{2\pi}}\right)^{c_s(y)} e^{-\frac{1}{2\sigma_s^2}(u_s(y)+w_{1,s}^2 u'_s(y)-2w_{1,s}r_s(y)-2w_{0,s}m_s(y)+2w_{0,s}w_{1,s}m'_s(y)+w_{0,s}^2 c_s(y))}. \tag{4.2}$$

Λ_s is derived from a product of Gaussian $\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{1}{2\sigma^2}(x-\mu)^2}$ distributions for the event values in leaf s, where instead of comparing a value x to the empirical mean μ, we compare it to a linear regression estimate, $w_0 + w_1 x'$, based on its previous value x'. The linear regression parameters (w_0, w_1) capture trend information in the temporal data. $u_s(y) = \sum_{x \lhd s} x^2$, $u'_s(y) = \sum_{x' \lhd s} x'^2$, $r_s(y) = \sum_{x,x' \lhd s} xx'$, $m_s(y) = \sum_{x \lhd s} x$ and $m'_s(y) = \sum_{x' \lhd s} x'$ are sufficient statistics, where $x, x' \lhd s$ indicates that the event that produced the value x occurred while in state s and x' is the value of the event preceding it. The extended model's parameters are $\{\lambda_s, w_{0,s}, w_{1,s}, \sigma_s^2\}_{s\in\Sigma}$.

Figure 4.1 shows an example extended C-PCIM decision tree for a very small

subset of the EHR data set we use for experimentation. The tree consists of three tests (inner nodes) and five leaves containing the model parameters: the rate plus the extended parameters.

The X in the tree is a placeholder for an event type. This placeholder allows the conjoint PCIM to share a single tree among all event types (Parikh et al., 2012): The tests with an X can be applied to any event type. The model firsts asks if there has been at least five occurrences of the event type in questions in the last 12 hours. If not, it checks if the blood pressure has been measured at least five times in the same time period. In other words, if there has not been much activity for this patient, has he had, at least, his blood pressure measured?

On the other hand, if there has been activity for the patient for this type of measurement in the last 12 hours, the model then refines the check to more recent activity (within the last six hours). If there has been recent activity, the model deems this as significant and checks for the actual values of measurements in the mean test at the bottom left inner node.

4.4.2 Feature extraction based on the Fisher kernel

To extract features, we use the Fisher kernel proposed by Jaakkola and Haussler (1998):

$$K(y_i, y_j) = U_{y_i}^{\top} I^{-1} U_{y_j},$$

where I is the Fisher information matrix and $U_y = \nabla_{\Theta} \ln p(y|\Theta)$ is the Fisher score. Given a parameterized probabilistic model $p(y_i|\Theta)$, where Θ is set to $\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{N} \ln p(y_i|\Theta)$,

this corresponds to the feature mapping

$$\phi(y_i) = \nabla_\Theta \ln p(y_i|\Theta)\, I^{-\frac{1}{2}}.$$

We compute the Fisher information matrix as $I_{i,j} = -E_\Theta\left[\frac{\partial^2}{\partial\theta_i \partial\theta_j} \ln p(y|\Theta)\right]$.

Each feature vector for each patient encodes how the model parameters should be "stretched" to better fit that patient. Patients for which the parameters must be changed in similar ways are similar in the context of the probabilistic model.

Given the log likelihood for the extended C-PCIM,

$$l(y;\Theta) = \ln p(y|S,\Theta) = \sum_{s\in\Sigma} \ln \Lambda_s + \sum_{s\in\Sigma} c_s(y) \ln \lambda_s - \sum_{s\in\Sigma} \lambda_s d_s(y),$$

obtained from Equation 4.1, the features for patient i are

$$\phi(y_i) = \left[\frac{\partial l(y_i;\Theta)}{\partial \lambda_1}, \frac{\partial l(y_i;\Theta)}{\partial w_{0,1}}, \frac{\partial l(y_i;\Theta)}{\partial w_{1,1}}, \frac{\partial l(y_i;\Theta)}{\partial \sigma_1^2}, \right.$$
$$\left. \ldots, \frac{\partial l(y_i;\Theta)}{\partial \lambda_{|\Sigma|}}, \frac{\partial l(y_i;\Theta)}{\partial w_{0,|\Sigma|}}, \frac{\partial l(y_i;\Theta)}{\partial w_{1,|\Sigma|}}, \frac{\partial l(y_i;\Theta)}{\partial \sigma^2_{|\Sigma|}}\right] I^{-\frac{1}{2}},$$

where

$$\begin{aligned}
\frac{\partial l}{\partial \lambda_s} &= \frac{c_s(y)}{\lambda_s} - d_s(y)a \\
\frac{\partial l}{\partial w_{0,s}} &= -2m_s(y) + 2w_{1,s}m'_s(y) + 2c_s(y)w_{0,s} \\
\frac{\partial l}{\partial w_{1,s}} &= 2w_{1,s}u'_s(y) - 2r_s(y) + 2w_{0,s}m'_s(y) \\
\frac{\partial l}{\partial \lambda_s^2} &= -\frac{c_s(y)}{2\sigma^2} + \frac{A}{\sigma^4},
\end{aligned}$$

and where $A = (u_s(y) + w_{1,s}^2 u'_s(y) - 2w_{1,s}r_s(y) - 2w_{0,s}m_s(y) + 2w_{0,s}w_{1,s}m'_s(y) + w_{0,s}^2 c_s(y))$.

4.4.3 Co-clustering

We apply the co-clustering algorithm presented in Figure 1.1 to the feature matrix X formed from the Fisher information features. The algorithm is guided by a compression-based cost function that applies the MDL principle to automatically determine the number of clusters. The feature matrix X is real-valued, so we employ Shannon's differential entropy in our cost function.

MDL-based cost function

Our two-part MDL cost function is shown in Equation 4.3. The description length of the model is computed as the number of bits required to encode the assignments of M rows to K row clusters and N columns to L column clusters plus the number of bits to encode the description of each block. The data description length is the number of bits required to encode the data within each block, computed as the Shannon's differential entropy for real-valued data. The encoding cost for a particular co-clustering (ϕ, ψ) is

$$F_{\mathcal{N}}(\phi,\psi) = \overbrace{M \ln K + N \ln L + KL\frac{c}{2}\ln\frac{1}{\rho}}^{\text{model description length}} + \underbrace{\sum_{b}^{KL} n_b \frac{1}{2}\ln(2\pi e(\hat{\sigma}_b^2 + \rho))}_{\text{data description length}}. \tag{4.3}$$

A natural way of maximizing the compression rate is by having homogeneous blocks. We thus model each block as a Gaussian distribution. We approximate the number of bits required to encode each block's Gaussian parameters (μ, σ) by $\frac{c}{2}\ln\frac{1}{\rho}$, where $\ln\frac{1}{\rho}$ is the number of bits, to a certain precision ρ, used to encode each parameter according to

```
   Input: X
   Output: (φ*, ψ*)
 1 begin
 2     [K, L] ← [lg M, lg N]
 3     (φ, ψ) ← randomize_{φ,ψ}(K,L)
 4     (φ, ψ) ← optimize_{φ,ψ}(φ, ψ) // base co-clustering, Figure 1.1
 5     repeat
 6         repeat
 7             increment
 8                 K ← K + 1 by splitting some row cluster
 9                 (φ, ψ) ← optimize_{φ,ψ}(φ, ψ)
10             Repeat for L
11         until no more positive splits
12         repeat
13             decrement
14                 K ← K − 1 by deleting some row cluster
15                 (φ, ψ) ← optimize_{φ,ψ}(φ, ψ)
16             Repeat for L
17         until no more positive deletions
18     until no more positive changes
```

Figure 4.2: KL Search algorithm.

some distribution encoded with c bits. The number of bits used to encode each matrix block element is computed as the Gaussian differential entropy with respect to the block's approximated variance $\hat{\sigma}_b^2$ up to precision ρ.

Optimization

Each row or column is placed in the row cluster or column cluster for which its probability is maximized with respect to the row cluster or column cluster descriptions. The function f is thus

$$f(X, Z, i, j, k, l) = p(x_{ij}|z_{kl}) = \frac{1}{\sqrt{2\pi\hat{\sigma}_{kl}^2}} e^{-\frac{(x_{ij}-\hat{\mu}_{kl})^2}{2\hat{\sigma}_{kl}^2}}$$

where elements of Z are of the form

$$z_{kl} = (\hat{\mu}_{kl}, \hat{\sigma}_{kl}) .$$

where $(\hat{\mu}, \hat{\sigma})$ are the maximum likelihood estimators.

To find the number of clusters, we employ a hill-climbing search algorithm. It starts from a randomly computed initial co-clustering and alternates between adding and removing clusters until a local optimum is reached. (See Figure 4.2.)

4.5 Experiments

We experimented on real EHR data and compared our method to several others on two measures: (1) how well are the clusters enriched with respect to mortality and (2) how well the clusters identify physiologic patterns.

4.5.1 Data set

The data set we use was collected from the PICU EHR archive at Children's Hospital Los Angeles. It contains over 10,000 patient episodes collected over 10 years and includes essentially all PICU episodes that could be reliably extracted and verified. The data set includes demographics, outcomes and other encounter information, vitals, laboratory test results, drugs administered, and interventions performed for each patient. We excluded encounter information such as outcomes and diagnoses. In total, we considered 391 variables for each patient: demographics (1), encounter information (2), vitals (27), labs (121), drugs (194), interventions (46), shown in Appendix C.

As the C-PCIM model shares a single conditional intensity function over all variables, we normalized the values. Age dependent variables such as heart rate were normalized for age by dividing by the age's median value among healthy children of the same sex (according to published tables). All the data were z-normalized. Data likely to be erroneous were removed. Specifically, data falling outside of ± 4 standard deviations from the mean of each variable were removed. As patients from the ICU are very ill, some extreme values may still be correct.

4.5.2 Procedure

Assessing the performance of any unsupervised learning procedure is difficult. Unsupervised learning methods are used mainly in exploratory analysis, where the ground truth is unknown. The goal of our research is to discover clusters with (1) discernible physiologic patterns and (2) with prognostic significance. We thus compared our method (CC-PCIM) to several others based on these two criteria. The other methods were selected on the basis of their applicability to sparse temporal data: the spectral clustering method of Ng et al. (2001) on the features generated by our method. We also tested the co-clustering algorithm of our method and the spectral clustering algorithm on a time-window discretization of the data. Finally, we tested our method on all the data (391 variables).

1. our method (CC-PCIM)
2. spectral clustering on our features (S-PCIM)
3. our co-clustering on time-discretized data (CC-PAA)
4. spectral clustering on time-discretized data (S-PAA)

X = particular variable (blood pressure, heart rate, etc.)
X's most recent value $\geq \tau$
X's mean value (in some interval) $\geq \tau$
X's variance (in some interval) $\geq \tau$
X's number of events (in some interval) $\geq \tau$
$T \in$ specific interval within the hour
$X \in$ specific group (vitals, interventions, etc.)
$X \in$ specific category (cardiac, neurological, etc.)

Figure 4.3: C-PCIM basis binary test functions. X is the variable currently being tested. T is the current time. τ is one of several threshold values.

5. our method on all the data

The spectral clustering methods, S-PCIM and S-PAA, use the number of clusters found by our co-clustering algorithms, CC-PCIM and CC-PAA, respectively.

We want the C-PCIM to learn how events (vitals measurements, interventions performed, etc.) depend on the type, timing, and value of prior events. We supplied the C-PCIM learning algorithm with time, value, and label-specific basis functions (tests) to build the decision tree. Figure 4.3 shows these basis functions. These tests encode information such as whether or not the event sequence y contains at least τ events of a specific type with timestamps in a specified interval and whether the mean value or variance is at least τ, over an interval. The test that directly tests the current time is intended to encode the non-random missingness of data due to the measurement process: For example, in this particular ICU, measurements for particular variables are recorded at the top of the hour every hour.

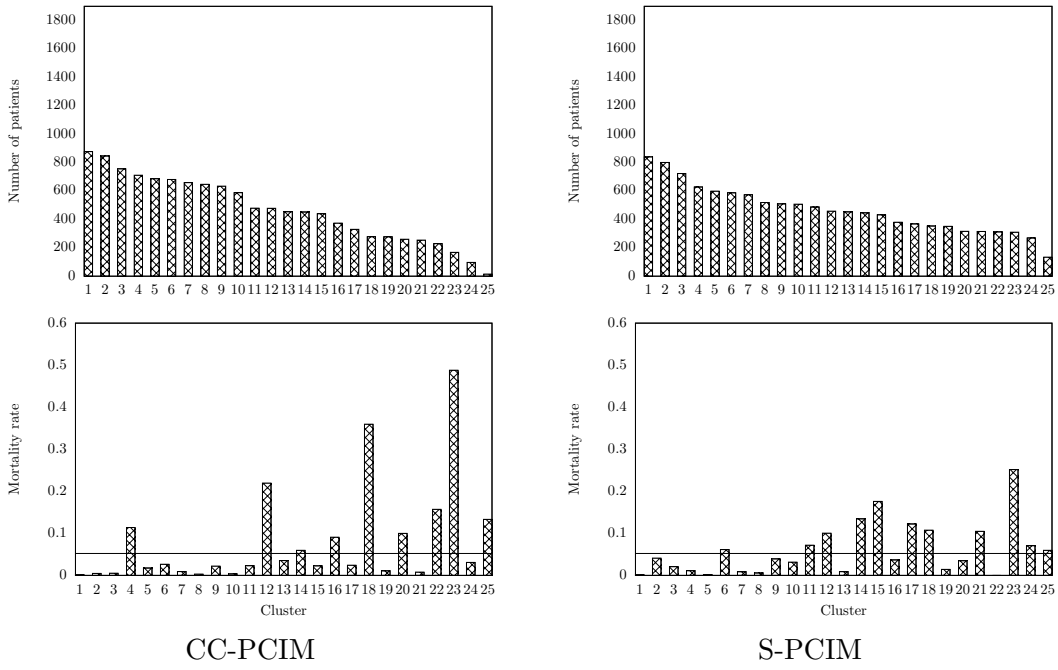


Figure 4.4: Cluster size and mortality enrichment. The number of clusters was automatically determined by our method (CC-PCIM).

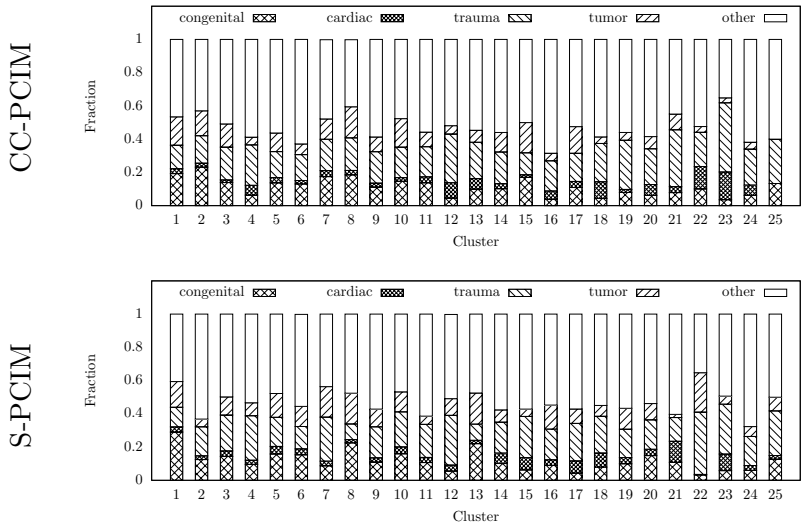


Figure 4.5: Physiologic patterns: diagnosis distribution per cluster.

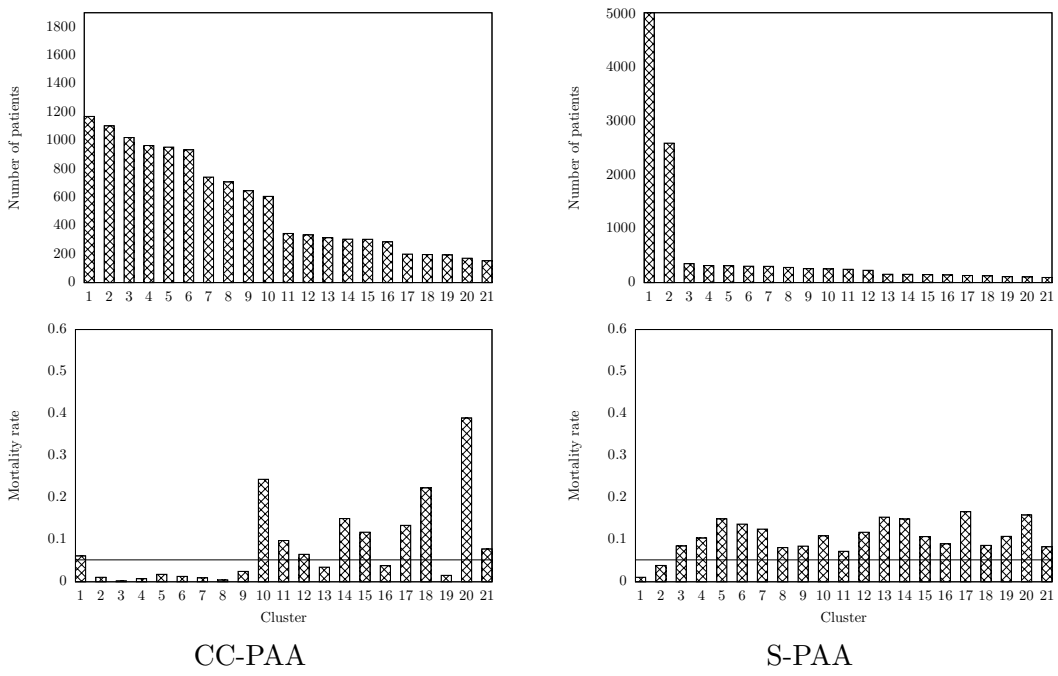


Figure 4.6: Cluster size and mortality enrichment for time-window discretized data.

4.5.3 Results

For the comparison to the other methods, we excluded categorical and binary data, such as labs, drugs, and some interventions. A total of 35 temporal variables were used: 17 from vitals and 18 from interventions (see Appendix C). We also present results from running our method on all 391 temporal variables. The experiments were done using only the first 24 hours of data for each patients. There is utility in being able to predict a patient's outcome within a short time of being admitted to the ICU.

Figure 4.4 shows the mortality enrichment per cluster for the first two experiments. In terms of mortality enrichment per cluster, our method (CC-PCIM) performed significantly better than applying spectral clustering on our features. Figure 4.5 shows the

physiologic patterns within clusters. These patterns are more difficult to determine as patients who die share similar conditions/symptoms as those who survive. The original 300 diagnoses were reduced to 5 categories more amenable to analysis. Our CC-PCIM method found clusters with high mortality that tend to have more diagnoses related to cardiac and fewer related to congenital conditions compared to clusters of lower mortality. This distinction is not so clear in the clusters produced by the spectral clustering method.

Figure 4.6 shows the mortality enrichment for experiments 3 and 4 on time-window discretized data. The performance is significantly worse than using our features. Figure 4.7 shows the results of our method on the entire data. The performance is similar to our method on the subset of the data. Note, however, that the method found an interesting cluster, 26, that is a high-mortality cluster where the rate of congenital conditions is not lower, but higher, than the rate of cardiac conditions as has been seen in all other high-mortality clusters in previous experiments. Another interesting cluster, 27, has no congenital conditions at all, and the rate for cardiac conditions is much higher than any other cluster. Finally, Figure 4.8 compares the performance of the various methods on the basis of mortality prediction. It confirms the better performance of our method.

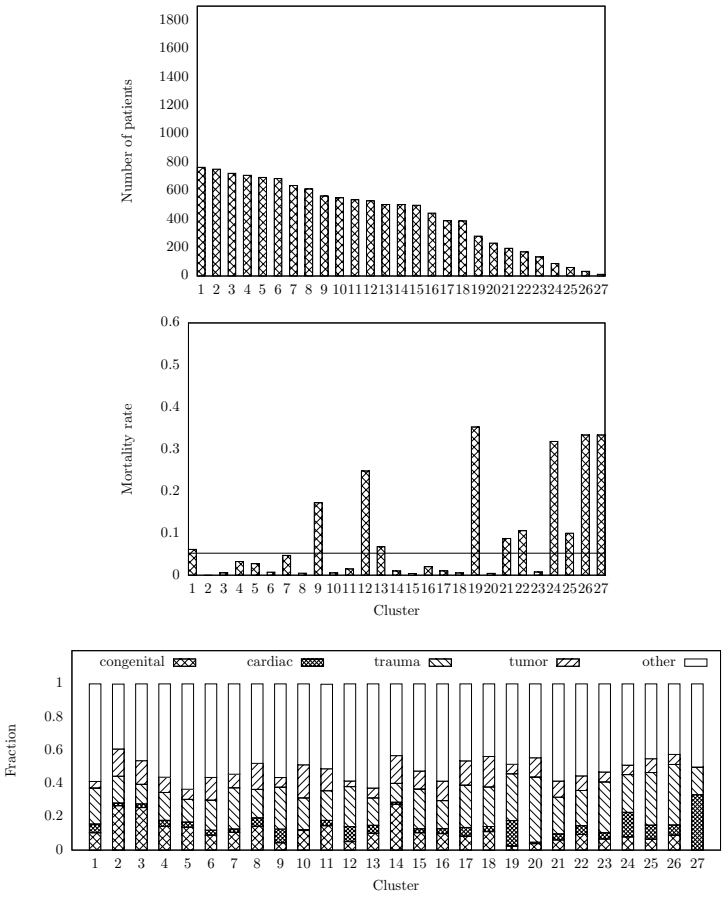


Figure 4.7: Results from running our method on all 391 variables, including numeric, categorical and binary data. Top: cluster sizes, middle: mortality rate per cluster, bottom: diagnosis distribution per cluster.

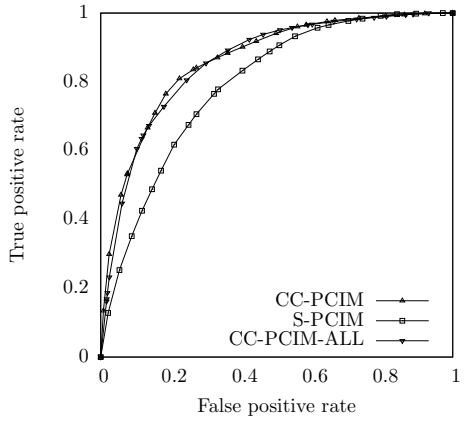


Figure 4.8: ROC curve. For each method, a classifier was constructed by associating each patient with the mortality rate of the cluster they are in. Patients with a mortality rate on or above a specified threshold are then classified as dead. Having constructed such a classifier for each method, we plotted the true positive rate versus the false positive rate for various thresholds between 1 and 0.

Chapter 5

Conclusion

Clustering is the unsupervised learning task of discovering patterns in data by finding groups of objects that are similar according to some predefined measure of similarity. Using different similarity measures may find different groupings. For this and other reasons, the task of clustering is inherently tied to the application problem and no universal clustering exists for all applications.

Co-clustering can be viewed as a generalization of clustering to a wider set of data. It operates on relational data as well as affinity data. Co-clustering clusters both the rows and the columns of a two-dimensional data matrix by discovering the block structure in the matrix. By clustering the columns (features) during the process, objects are no longer compared on a feature-by-feature basis, but on a summary of similar features. This has the effect of performing a regularization, which may produce clusters that better extract information in the data. Another advantage of co-clustering is that by producing feature clusters as well as the object clusters, additional patterns may be discovered, such

as correlated features.

In this dissertation we demonstrated the applicability of co-clustering to social network data and medical data. Specifically, we developed co-clustering methods to address three problems: (1) clustering relational data for regular equivalence in social network applications, (2) finding a symmetric clustering for asymmetric data and (3) clustering ICU patients based on their physiologic data.

In spite of its theoretical significance, sociologists have long been in need of a mathematical model for regular equivalence and only relied on poorly performing ad-hoc methods to analyze such relations in data. We provided such a model in the context of compression theory. Our co-clustering method can automatically differentiate between structural and regular clusters and can determines the number of clusters, which has been a vexing problem in sociology.

In applications such as world trade data, we might be interested in associating each country with a single label denoting its trading ties, yet these data encode a bi-directional relation between countries and thus, co-clustering methods would produce two different labels per country. We provided a framework where a co-clustering method can be made to produce a single clustering.

For medical data applications, we presented a method for clustering multi-dimensional, temporal physiologic data. The method was able to find clusters with discernible physiologic patterns and with diagnostic significance. Our experiments showed that our co-clustering method performed better than state-of-the-art clustering methods such as spectral clustering.

In summary, the results from our experiments on the various data sets show that co-clustering is a competitive algorithm for finding meaningful clusters in data. Indeed, co-clustering performed better than spectral clustering—a powerful clustering algorithm—on the feature data from the medical application.

Finally, as most other research, there is more that can be done. We can identify two possible extensions of our research, both on social network data and medical data applications. The cost function we provided for regular equivalence in social network data deals with the standard definition of a regular block as one whose rows and columns have at least one 1 (a 1-covered block). There have been additional types of regular blocks defined in the literature (Doreian et al., 1994, 2005) for which new cost function definitions may be needed.

In our co-clustering solution of medical data, we did not reap the full benefits of co-clustering: We did not provide an analysis of the clusters of the features extracted form the probabilistic model. Interpreting the extracted Fisher-information features from the C-PCIM is not straight forward. If we could find interpretations for these features, a more complete analysis could be done, which could potentially provide the distinctions in patients from different clusters.

Bibliography

Emile Aarts and Jan Korst. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, Inc., 1989.

Aris Anagnostopoulos, Anirban Dasgupta, and Ravi Kumar. Approximation Algorithms for Co-clustering. In *Proceedings of the twenty-seventh ACM Symposium on Principles of Database Systems*, pages 201–210, 2008.

Carolyn J Anderson, Stanley Wasserman, and Katherine Faust. Building Stochastic Blockmodels. *Social Networks*, 14(1–2):137–161, 1992.

Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM International Conference on Management of Data*, pages 49–60. ACM, 1999.

A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation. In *Proceedings of the Tenth ACM International Conference on Knowledge Discovery and Data Mining*, pages 509–514. ACM, 2004.

Vladimir Batagelj. Notes on Blockmodeling. *Social Networks*, 19:143–155, 1997.

Vladimir Batagelj, Patrick Doreian, and Anuska Ferligoj. An Optimizational Approach to Regular Equivalence. *Social Networks*, 14(1-2):121–135, 1992a.

Vladimir Batagelj, Anuska Ferligoj, and Patrick Doreian. Direct and Indirect Methods for Structural Equivalence. *Social Networks*, 14(1-2):63–90, 1992b.

Vladimir Batagelj, Andrej Mrvar, Anuska Ferligoj, and Patrick Doreian. Generalized Blockmodeling with Pajek. *Metodoloski zvezki*, 1(2):455–467, 2004.

N.W. Biggart and M.F. Guillen. Developing Difference: Social Organization and the Rise of the Auto Industries of South Korea, Taiwan, Spain, and Argentina. *American Sociological Review*, pages 722–747, 1999.

Stephen P. Borgatti and Martin G. Everett. Regular Blockmodels of Multiway, Multimode Matrices. *Social Networks*, 14(1-2):91–120, 1992.

John P. Boyd. Finding and testing regular equivalence. *Social Networks*, 24(4):315–331, 2002.

John P. Boyd and Martin G. Everett. Relations, Residuals, Regular Interiors, and Relative Regular Equivalence. *Social Networks*, 21(2):147–165, 1999.

John P. Boyd and Kai J. Jonas. Are Social Equivalences Ever Regular? Permutation and Exact Tests. *Social Networks*, 23(2):87–123, 2001.

Thomas Brinkhoff. New Mexico (USA): State, Major Cities, Towns & Places, February 2011.

Michael Brusco and Douglas Steinley. A Variable Neighborhood Search Method for Generalized Blockmodeling of Two-mode Binary Matrices. *Journal of Mathematical Psychology*, 51(5):325–338, 2007.

Michael Brusco and Douglas Steinley. A Tabu-Search Heuristic for Deterministic Two-Mode Blockmodeling of Binary Network Matrices. *Psychometrika*, 76(4):612–633, 2011.

Michael J. Brusco and Douglas Steinley. Integer Programs for One- and Two-mode Blockmodeling Based on Prespecified Image Matrices for Structural and Regular Equivalence. *Journal of Mathematical Psychology*, 53(6):577–585, 2009.

D. Chakrabarti, S. Papadimitriou, D. S. Modha, and C. Faloutsos. Fully Automatic Cross-associations. In *Proceedings of the Tenth ACM International Conference on Knowledge Discovery and Data Mining*, pages 79–88. ACM, 2004.

Y. Cheng and G. M. Church. Biclustering of Expression Data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103, 2000.

M. Chignell, M. Rouzbahman, R. Kealey, R. Samavi, E. Yu, and T. Sieminowski. Nonconfidential Patient Types in Emergency Clinical Decision Support. *Security Privacy, IEEE*, 11(6):12–18, 2013.

Walter Christaller. *Central Places in Southern Germany*. Prentice-Hall, 1966.

Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/Gather: a Cluster-based Approach to Browsing Large Document Collections. In *Proceedings of the 15th Annual International ACM Conference on Research and Development in Information Retrieval*, pages 318–329. ACM, 1992.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes Volume I: Elementary Theory and Methods*. Springer, second edition, 2003.

Allison Davis, Burleigh B. Gardner, and Mary R. Gardner. *Deep South*. University of Chicago Press, 1941.

D. Defays. An Efficient Algorithm for a Complete Link Method. *The Computer Journal*, 20(4):364–366, 1977.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society , Series B*, 39(1):1–38, 1977.

Standard International Trade Classification. Department of Economic and Social Affairs, United Nations, 4 edition, 2006.

Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. Information-theoretic Co-clustering. In *Proceedings of the Ninth ACM International Conference on Knowledge Discovery and Data Mining*, pages 89–98. ACM, 2003.

Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Kernel K-means: Spectral Clustering and Normalized Cuts. In *Proceedings of the Tenth ACM International Conference on Knowledge Discovery and Data Mining*, pages 551–556. ACM, 2004.

Chris Ding and Xiaofeng He. Cluster Merging and Splitting in Hierarchical Clustering Algorithms. In *Proceedings of IEEE International Conference on Data Mining*. IEEE Computer Society, 2002.

Patrick Doreian and Kayo Fujimito. Structures of Supreme Court Voting. *Connections*, 25 (3), 2003.

Patrick Doreian, Vladimir Batagelj, and Anuska Ferligoj. Partitioning Networks Based on Generalized Concepts of Equivalence. *The Journal of Mathematical Sociology*, 19(1):1–27, 1994.

Patrick Doreian, Vladimir Batagelj, and Anuska Ferligoj. Generalized Blockmodeling of Two-mode Network Data. *Social Networks*, 26:29–53, 2004a.

Patrick Doreian, Vladimir Batagelj, and Anuska Ferligoj. *Generalized Blockmodeling (Structural Analysis in the Social Sciences)*. Cambridge University Press, 2004b.

Patrick Doreian, Vladimir Batagelj, and Anuska Ferligoj. *Generalized Blockmodeling*. Structural Analysis in the Social Sciences. Cambridge University Press, 2005.

Finale Doshi-Velez, Yaorong Ge, and Isaac Kohane. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics*, 133(1):54–63, 2014.

B. Curtis Eaton and Richard G. Lipsey. An Economic Theory of Central Places. *The Economic Journal*, 92(365):56–72, 1982.

T. Eckes and P. Orlik. An Error Variance Approach to Two-mode Hierarchical Clustering. *Journal of Classification*, 10:51–74, 1993.

Philippe Esling and Carlos Agon. Time-series Data Mining. *ACM Computing Surveys*, 45 (1):12:1–12:34, 2012.

Martin G. Everett and Stephen P. Borgatti. An Extension of Regular Colouring of Graphs to Digraphs, Networks and Hypergraphs. *Social Networks*, 15(3):237–254, 1993.

Brian S. Everitt, Sabine Landau, and Morven Leese. *Cluster Analysis*. Wiley Publishing, 4th edition, 2009.

R. C. Feenstra, R. E. Lipsey, H. Deng, A. C. Ma, and H. Mo. World Trade Flows: 1962-2000. Working Paper 11040, National Bureau of Economic Research, 2005.

S Fienberg and S Wasserman. Categorical Data Analysis of Single Sociometric Relations. *Sociological Methodology*, 12:156–192, 1981.

Linton C. Freeman. Finding Social Groups: A Meta-Analysis of the Southern Women Data. In *Dynamic Social Network Modeling and Analysis*, pages 39–97. National Academies Press, 2003.

Wolfgang Gaul and Martin Schader. A New Algorithm for Two-Mode Clustering. In *Data Analysis and Information Systems*, pages 15–23. Springer, 1996.

Andrew Gelman, Christian Robert, Nicolas Chopin, and Judith Rousseau. Bayesian Data Analysis, 1995.

Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

C. Goutte, L.K. Hansen, M.G. Liptrot, and E. Rostrup. Feature-space Clustering for fMRI Meta-analysis. *Human Brain Mapping*, 13(3):165–183, 2001.

Linda Greenhouse. In Year of Florida Vote, Supreme Court Also Did Much Other Work. *New York Times*, 2001.

Peter Grünwald. *A Tutorial Introduction to the Minimum Description Length Principle*, chapter 1–2, pages 3–79. MIT Press, 2005.

Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. ROCK: A Robust Clustering Algorithm for Categorical Attributes. In *Proceedings of the 15th International Conference on Data Engineering*, pages 512–521. IEEE Computer Society, 1999.

S. Gupta, M. Polonsky, A. Woodside, and C.M. Webster. The Impact of External Forces on Cartel Network Dynamics: Direct research in the diamond industry. *Industrial Marketing Management*, 39(2):202–210, 2010.

A. Gunoche, P. Hansen, and B. Jaumard. Efficient algorithms for divisive hierarchical clustering with the diameter criterion. *Journal of Classification*, 8(1):5–30, 1991.

J. Hansohm. Two-mode Clustering with Genetic Algorithms. In *Classification, Automation, and New Media*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 87–93. Springer Berlin Heidelberg, 2002.

J. A. Hartigan. Direct Clustering of a Data Matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

Paul W. Holland and Samuel Leinhardt. An Exponential Family of Probability Distributions for Directed Graphs. *Journal of the American Statistical Association*, 76(373):33–50, 1981.

Paul W. Holland, Kathryn Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

D. Huffman. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the Institute of Radio Engineers*, 40(9):1098–1101, 1952.

Tommi Jaakkola and David Haussler. Exploiting Generative Models in Discriminative Classifiers. In *Advances in Neural Information Processing Systems 11*, pages 487–493. MIT Press, 1998.

Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

Gareth M. James and Trevor J. Hastie. Functional Linear Discriminant Analysis for Irregularly Sampled Curves. *Journal of the Royal Statistical Society Series B*, 63:533–550, 2001.

Stephen C. Johnson. Hierarchical Clustering Schemes. *Psychometrika*, 32(3), 1967.

K.H. Kim and F.W. Roush. Group Relationships and Homomorphisms of Boolean Matrix Semigroups. *Journal of Mathematical Psychology*, 28(4):448–452, 1984.

Jon Kleinberg. An Impossibility Theorem for Clustering. In *Advances in Neural Information Processing Systems*, pages 446–453. MIT Press, 2002.

Sabine Krolak-Schwerdt. Two-Mode Clustering Methods: Compare and Contrast. In *Between Data Science and Applied Data Analysis*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 270–278. Springer Berlin Heidelberg, 2003.

Julia Lasserre and Christopher M. Bishop. Generative or Discriminative? Getting the Best of Both Worlds. *Bayesian Statistics*, 8:3–24, 2007.

L Lehman, M Saeed, G Moody, and R Mark. Similarity-Based Searching in Multi-Parameter Time Series Databases. *Computing in Cardiology*, 35(4749126):653–656, 2008.

J. Lin and Yuan Li. Finding structurally different medical data. In *22nd IEEE International Symposium on Computer-Based Medical Systems*, pages 1–8, 2009.

R. Lletí, M. C. Ortiz, L. A. Sarabia, and M. S. Sánchez. Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica Acta*, 515(1):87–100, 2004.

S. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions Information Theory*, 28 (2):129–137, 1982.

Francois Lorrain and Harrison C. White. Structural Equivalence of Individuals in Social Networks. *Journal of Mathematical Sociology*, 1:49–80, 1971.

August Losch. *The Economics of Location*. Yale University Press, 1954.

J. B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

M.C. Mahutga and D.A. Smith. Globalization, the Structure of the World Economy and Economic Development. *Social Science Research*, 40(1):257–272, 2011.

Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag New York, Inc., 2005.

Benjamin M. Marlin, David C. Kale, Robinder G. Khemani, and Randall C. Wetzel. Unsupervised Pattern Discovery in Electronic Health Care Data Using Probabilistic Clustering Models. In *Proceedings of the 2Nd ACM International Health Informatics Symposium*, pages 389–398. ACM, 2012.

G.P. Maxton and J. Wormald. *Time for a Model Change: Re-engineering the Global Automobile Industry*. Cambridge University Press, 2004.

I Van Mechelen, H H Bock, and P De Boeck. Two-mode Clustering Methods: A Structured Overview. *Statistical methods in medical research*, 13(5):363–394, 2004.

L. Scott Mills, Michael E. Soul, and Daniel F. Doak. The Keystone-Species Concept in Ecology and Conservation. *BioScience*, 43(4):219–224, 1993.

Boris Mirkin, Phipps Arabie, and Lawrence Hubert. Additive two-mode clustering: The error-variance approach revisited. *Journal of Classification*, 12(2):243–263, 1995.

S. Moore, A.C. Teixeria, and A. Shiell. The Health of Nations in a Global Context: Trade, Global Stratification, and Infant Mortality Rates. *Social science & medicine*, 63(1):165–178, 2006.

Andrew Y. Ng and Michael I. Jordan. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*, pages 841–848. MIT Press, 2001.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On Spectral Clustering: Analysis and an algorithm. In *Advances in Neural Information Processing System*, pages 849–856. MIT Press, 2001.

K. Nowicki and T. A. B. Snijders. Estimation and Prediction for Stochastic Blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

Ankur Parikh, Asela Gunawardana, and Christopher Meek. Conjoint Modeling of Temporal Dependencies in Event Streams. In *Association for Uncertainty in Artificial Intelligence Bayesian Modelling Applications Workshop*, pages 65–73, 2012.

P. Pattison. *Algebraic Models for Social Networks*. Structural Analysis in the Social Sciences. Cambridge University Press, 1993.

Philippa E. Pattison. The Analysis of Semigroups of Multirelational Systems. *Journal of Mathematical Psychology*, 25(2):87–118, 1982.

R. Paul and AS.M.L. Hoque. Clustering medical data to predict the likelihood of diseases. In *Fifth International Conference on Digital Information Management*, pages 44–49, 2010.

Murray M. Pollack, Kantilal M. Patel, and Urs E. Ruttimann. PRISM III: An Updated Pediatric Risk of Mortality Score. *Critical Care Medicine*, 24(5):743–752, 1996.

Girish Punj and David W. Stewart. Cluster Analysis in Marketing Research: Review and Suggestions for Application. *Journal of Marketing Research*, 20(2):134–148, 1983.

William M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

S. Ray and R. H. Turi. Determination of number of clusters in K-means clustering and application in colour image segmentation, 1999.

J. Rissanen. Modeling By Shortest Data Description. *Automatica*, 14:465–471, 1978.

Karl Rohe and Bin Yu. Co-clustering for Directed Graphs; The Stochastic Co-blockmodel and a Spectral Algorithm, April 2012.

Lee Douglas Sailer. Structural Equivalence: Meaning and Definition, Computation and Application. *Social Networks*, 1(1):73–90, 1978.

S. Saria, D. Koller, and A. Penn. Learning individual and population level traits from clinical temporal data. In *Predictive Models in Personalized Medicine Workshop. Twenty-fourth Annual Conference on Neural Information Processing Systems*, pages 112–135, 2010.

C. E. Shannon. A Mathematical Theory of Communication. *The Bell Systems Technical Journal*, 27:379–423, 623–656, 1948.

Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997.

Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

R. Sibson. SLINK: An Optimally Efficient Algorithm for the Single-linkage Cluster Method. *The Computer Journal*, 16(1):30–34, 1973.

D.A. Smith and D.R. White. Structure and Dynamics of the Global Economy: Network Analysis of International Trade 1965-1980. *Social Forces*, pages 857–893, 1992.

D.L. Spar. Markets: Continuity and Change in the International Diamond Market. *The Journal of Economic Perspectives*, 20(3):195–208, 2006.

Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In *In Knowledge Discovery and Data Mining Workshop on Text Mining*, 2000.

Catherine A. Sugar, Gareth, and M. James. Finding the number of clusters in a data set: An information theoretic approach. *Journal of the American Statistical Association*, 98: 750–763, 2003.

Suzanne Tamang and Simon Parsons. Using Semi-parametric Clustering Applied to Electronic Health Record Time Series Data. In *Proceedings of the 2011 Workshop on Data Mining for Medicine and Healthcare*, pages 72–75. ACM, 2011.

Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining*, pages 807–816. ACM, 2009.

Robert L. Thorndike. Who Belongs in the Family? *Psychometrika*, 18(4):267–276, 1953.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a dataset via the Gap statistic. 63:411–423, 2000.

J. Trejos and W. Castillo. Simulated Annealing Optimization for Two-mode Partitioning. In *Classification and Information at the Turn of the Millennium*, pages 135–142. Springer Berlin Heidelberg, 2000.

Joost van Rosmalen, Patrick J. Groenen, Javier Trejos, and William Castillo. Optimization Strategies for Two-Mode Partitioning. *Journal of Classification*, 26(2):155–181, 2009.

I. Wallerstein. Three Paths of National Development in Sixteenth-century Europe. *Studies in Comparative International Development*, 7:95–101, 1972.

S. Wasserman and C. Anderson. Stochastic a posteriori Blockmodels: Construction and assessment. *Social Networks*, 9:1–36, 1987.

D.R. White and K.P. Reitz. Graph and Semigroup Homomorphisms on Networks of Relations. *Social Networks*, 5:193–234, 1983.

Harrison C. White, Scott A. Boorman, and Ronald L. Breiger. Social Structure from Multiple Networks. I. Blockmodels of Roles and Positions. *American Journal of Sociology*, 81(4):730–780, 1976.

Appendix A

Trade Data Clusters

Place clusters A and B and activity clusters A and B produced by both the compression-based criterion and the error-based criterion of the New Mexico economic activity data are given in the main paper. The other clusters are shown here in this appendix.

The other five place clusters by the compression-based criterion

C					
Alamogordo	Cedar Crest	Gallup	Los Alamos	Rio Rancho	Tijeras
Artesia	Clayton	Grants	Los Lunas	Ruidoso	Truth Or
Aztec	Corrales	Kirtland	Lovington	Santa Rosa	Tucumca
Belen	Deming	Kirtland Afb	Placitas	Santa Teresa	Tularosa
Bernalillo	Edgewood	Lamy	Portales	Silver City	
Bloomfield	Espanola	Las Vegas	Ranchos De	Socorro	
Bosque Farms	Flora Vista	Lordsburg	Raton	Taos	

D					
Algodones	Cerrillos	Holloman Afb	Lemitar	Nogal	Sandia Park
Anthony	Chaparral	Jemez Springs	Logan	Peralta	Santa Cruz
Arenas Valley	Dexter	La Luz	Mesilla Pa	Ruidoso Dow	Sunland Park
Bayard	Estancia	La Plata	Mesquite	San Ysidro	Vado

E					
Amistad	Des Moines	Hagerman	Mora	Ramah	Veguita
Animas	El Prado	Hernandez	Moriarty	Ribera	Waterflow
Cannon Afb	Elida	Jemez Pueblo	Mosquero	Roy	Weed
Canoncito	Fence Lake	La Mesa	Mountainair	Sapello	
Cimarron	Floyd	Lincoln	Pecos	Shiprock	
Cloudcroft	Fort Sumner	Magdalena	Pinon	Texico	
Datil	Glorieta	Mayhill	Questa	Timberon	

F					
Alto	Corona	Guadalupita	Mesilla	Red River	Taos Ski Val
Angel Fire	Costilla	High Rls Mtn	Montezuma	Rehoboth	Tererro
Arrey	Crownpoint	Hillsboro	Monument	Reserve	Tesuque
Arroyo Hon	Cuba	Holman	Nageezi	Rociada	Thoreau
Arroyo Seco	Cubero	Hondo	Nara Visa	Rodeo	Tierra Amar
Blanco	Dixon	Hope	Navajo Dam	Rogers	Tohatchi
Bosque	Dora	Hurley	Newcomb	Rowe	Tome
Broadview	Dulce	Isleta	Ojo Caliente	Sacramento	Ute Park
Caballo	Eagle Nest	Jal	Organ	San Fidel	Vadito
Canones	El Rito	Jamestown	Paguate	San Jon	Vanderwagen
Carrizozo	Embudo	Jarales	Pena Blanca	San Jose	Vaughn
Causey	Eunice	Laguna	Penasco	San Juan P	Velarde
Cerro	Fairview	Lake Arthur	Picacho	San Patrici	Villanueva
Chacon	Faywood	Lakewood	Pinehill	San Rafael	Wagon Moun
Chamisal	Folsom	Loco Hills	Polvadera	Sanostee	White Sands
Chimayo	Fruitland	Loving	Ponderosa	Sheep Spri	Williamsburg
Church Rock	Garfield	Luna	Prewitt	Solano	Winston
Cleveland	Gila	Mc Intosh	Pueblo Of Ac	Springer	Zuni
Cliff	Grady	Melrose	Quemado	Stanley	
Cochiti Lake	Grenville	Mescalero	Radium Spri	Tajique	

G					
Alameda	Caprock	Dona Ana	La Jara	Mogollon	Sedan
Alcalde	Carson	Elephant Butte	La Loma	Navajo	Sunspot
Anton Ch	Casa Blanca	Encino	Lindrith	Newkirk	Tatum
Aragon	Chama	Fairacres	Los Ojos	Ocate	Tinnie
Bard	Clines Corners	Flying H	Maljamar	Pep	Tres Piedras
Bell Ran	Cochiti Pueblo	Fort Wingate	Maxwell	Pie Town	Vallecitos
Bent	Columbus	Gallina	Mc Alister	Pojoaque Va	Valmora
Bingham	Conchas Dam	Gladstone	Mc Donald	Regina	Watrous
Bluewater	Continental Div	Glenwood	Mentmore	San Antonio	Willard
Buckhorn	Cordova	Hanover	Milan	San Miguel	Yeso
Buena Vis	Coyote	Hatch	Milnesand	Santo Domi	
Capitan	Cuervo	House	Mimbres	Seboyeta	

The other five activity clusters by the compression-based criterion

C	
Primary economic activities: (0%)	5182 Data Processing, Hosting, and Related
	5222 Nondepository Credit Intermediation
Secondary economic activities: (7%)	5239 Other Financial Investment Activities
2372 Land Subdivision	5311 Lessors of Real Estate
2373 Highway, Street, and Bridge Const	5312 Offices of Real Estate Agents and Brok
2381 Foundation, Structure, and Building	5321 Automotive Equipment Rental
	5323 General Rental Centers
Tertiary economic activities: (93%)	5411 Legal Services
3118 Bakeries and Tortilla Manufacturing	5412 Accounting, Tax Prep., Bookkeeping
3323 Architectural and Structural Metals	5413 Architectural, Engineering
3327 Mach. Shops; Turned Prod.; Screw	5415 Computer Systems Design and Related
4236 Electrical and Electronic Goods	5416 Management, Scientific, and Technical
4239 Miscellaneous Durable Goods	5614 Business Support Services
4247 Petroleum and Petroleum Products	5616 Investigation and Security Services
4411 Automobile Dealers	6116 Other Schools and Instruction
4412 Other Motor Vehicle Dealers	6211 Offices of Physicians
4413 Automotive Parts, Accessories	6212 Offices of Dentists
4431 Electronics and Appliance Stores	6213 Offices of Other Health Practitioners
4441 Building Material and Supplies	6214 Outpatient Care Centers
4442 Lawn and Garden Equipment	8114 Personal and Household Goods Repair
4461 Health and Personal Care Stores	8123 Drycleaning and Laundry Services
4471 Gasoline Stations	8133 Social Advocacy Organizations
4511 Sporting Goods, Hobby, and Music	8134 Civic and Social Organizations
4521 Department Stores	8139 Business, Professional, Labor, Political
4533 Used Merchandise Stores	9241 Administration of Environmental
5111 Newspaper, Periodical, Book	

D	
Primary economic activities: (3%)	4531 Florists
1114 Greenhouse, Nursery, Floriculture	4541 Electronic Shopping and Mail-Order
1119 Other Crop Farming	4542 Vending Machine Operators
1121 Cattle Ranching and Farming	4543 Direct Selling Establishments
1151 Support Activities for Crop Product	4832 Inland Water Transportation
1153 Support Activities for Forestry	4841 General Freight Trucking
	4851 Urban Transit Systems
Secondary economic activities: (5%)	4852 Interurban and Rural Bus Transp
2111 Oil and Gas Extraction	4853 Taxi and Limousine Service
2123 Nonmetallic Mineral Mining, Quar	4855 Charter Bus Industry
2131 Support Activities for Mining	4859 Other Transit and Ground Passen
2211 Electric Power Gen, Transmission	4861 Pipeline Transportation of Crude
2212 Natural Gas Distribution	4879 Scenic and Sightseeing Transp
2213 Water, Sewage and Other Systems	4881 Support Activities for Air Transp
2379 Other Heavy and Civil Engineering	4882 Support Activities for Rail Transp
	4884 Support Activities for Road Transp
Tertiary economic activities: (92%)	4931 Warehousing and Storage
3111 Animal Food Manufacturing	5121 Motion Picture, Video Industr
3112 Grain and Oilseed Milling	5122 Sound Recording Industries
3114 Fruit and Vegetable Preserv, Spec	5151 Radio and Television Broadcasting
3115 Dairy Product Manufacturing	5152 Cable and Other Subscription Prog
3116 Animal Slaughtering and Processing	5171 Wired Telecommunications Carriers
3119 Other Food Manufacturing	5172 Wireless Telecomm. Carriers
3121 Beverage Manufacturing	5179 Other Telecommunications
3141 Textile Furnishings Mills	5191 Other Information Services
3149 Other Textile Product Mills	5223 Activities Related to Credit
3169 Other Leather and Allied Product	5231 Securities, Commodity Contracts
3212 Veneer, Plywood, Engineered Wood	5241 Insurance Carriers
3219 Other Wood Product Mfg	5242 Agencies, Brokerages, Insurance
3222 Converted Paper Product Mfg	5322 Consumer Goods Rental
3241 Petroleum and Coal Products Mfg	5324 Commercial Industrial Mach. Rent
3251 Basic Chemical Manufacturing	5414 Specialized Design Services
3253 Pesticide, Fert, Other Agric Chem	5417 Scientific Research and Develop
3256 Soap, Cleaning Compound, Toilet	5418 Advertising and Related Services
3259 Other Chemical Product, Prep Mfg	5511 Mgmt of Companies, Enterprises
3261 Plastics Product Manufacturing	5613 Employment Services
3271 Clay Product and Refractory Mfg	5615 Travel Arrangement and Reserv
3273 Cement and Concrete Prod Mfg	5619 Other Support Services
3279 Other Nonmetallic Mineral Prod Mfg	5621 Waste Collection
3315 Foundries	5622 Waste Treatment and Disposal
3328 Coating, Engraving, Heat Treating	5629 Remediation, Other Waste Mgmt
3329 Other Fabricated Metal Product Mfg	6113 Colleges, Universities, Prof Schools
3331 Agriculture, Construction, Mining	6114 Business Schools Computer Mgmt

D (continued)	
3333 Commerc Service Industr Mach Mfg	6115 Technical and Trade Schools
3334 Vent Heat Air Commerc Equp Mfg	6117 Educational Support Services
3335 Metalworking Machinery Mfg	6215 Medical and Diagnostic Labs
3336 Engine Turbine Power Equp Mfg	6216 Home Health Care Services
3339 General Purpose Machinery Mfg	6219 Ambulatory Health Care Services
3342 Communications Equipment Mfg	6221 General Medical and Surgical Hosp
3345 Electromedical Control Instru Mfg	6222 Psychiatric Subst Abuse Hosp
3351 Electric Lighting Equipment Mfg	6231 Nursing Care Facilities
3359 Electrical Equipment Comp Mfg	6233 Community Care Facilities for Elderly
3363 Motor Vehicle Parts Manufacturing	6239 Other Residential Care Facilities
3371 House Furnit Kitchen Cabinet Mfg	6242 Community Food Housing Relief
3391 Medical Equipment Supplies Mfg	6243 Vocational Rehabilitation Services
4231 Motor Vehicle Parts Supplies Whsle	7111 Performing Arts Companies
4232 Furniture Home Furnishing Whsle	7112 Spectator Sports
4233 Lumber Other Construc Mat Whsle	7113 Promoters of Perf. Arts, Sports
4234 Prof Commerc Equip Supplies Whsle	7115 Independent Artists, Writers
4235 Metal Mineral (no Petroleum) Whsle	7121 Museums Historical Sites Similar
4237 Hardware Plumb Heat Equip Whsle	7131 Amusement Parks and Arcades
4242 Drugs Druggists' Sundries Whsle	7132 Gambling Industries
4243 Apparel, Piec Goods, Notions Whsle	7221 Full-Service Restaurants
4244 Grocery Related Prod Whsle	7223 Special Food Services
4245 Farm Prod Raw Material Whsle	7224 Drinking Places (Alcoholic Beverages)
4246 Chemical Allied Products Whsle	8112 Electronic Precision Equip Repair
4248 Beer Wine Alcoholic Bev Whsle	8113 Commerc Ind Mach Equip Repair
4251 Whsle Electronic Markets, Agents	8122 Death Care Services
4421 Furniture Stores	8129 Other Personal Services
4422 Home Furnishings Stores	9231 Administration of Human Resource
4453 Beer, Wine, and Liquor Stores	9251 Administ Hous Prog Urban Plan
4482 Shoe Stores	9261 Administ of Economic Program
4512 Book, Periodical, and Music Stores	9281 National Security International
4529 Other General Merchandise Stores	

E	
Primary economic activities: (3%)	3326 Spring, Wire Product Mfg
1133 Logging	3341 Computer, Peripheral Equp Mfg
	3344 Semiconductor, Electr Compnt Mfg
Secondary economic activities: (0%)	3353 Electrical Equipment Manufacturing
	3362 Motor Vehicle Body, Trailer Mfg
Tertiary economic activities: (97%)	3372 Office Furniture (and Fixtures) Mfg
3113 Sugar and Confectionery Prod Mfg	3379 Furniture Related Product Mfg
3132 Fabric Mills	4241 Paper, Paper Product Whlse
3133 Textile Fabric Finishing Coating Mills	4811 Scheduled Air Transportation
3159 Apparel Accessories Mfg	4821 Rail Transportation
3161 Leather Hide Tanning Finishing	4872 Scenic, Sightseeing Transport, Water
3211 Sawmills Wood Preservation	4885 Freight Transportation Arrangement
3254 Pharmaceutical Medicine Mfg	4889 Support Activities for Transport
3255 Paint, Coating, Adhesive Mfg	4921 Couriers
3262 Rubber Product Manufacturing	5173 Telecommunications Resellers
3272 Glass, Glass Product Mfg	5251 Insurance, Employee Benefit Funds
3311 Iron Steel Mills and Ferroalloy Mfg	5611 Office Administrative Services
3312 Steel Prod Mfg from Purchased Steel	7213 Rooming and Boarding Houses
3322 Cutlery Handtool Mfg	8132 Grantmaking and Giving Services
3325 Hardware Manufacturing	

F	
Primary economic activities: (10%)	3151 Apparel Knitting Mills
1142 Hunting and Trapping	3152 Cut and Sew Apparel Manufacturing
	3162 Footwear Manufacturing
Secondary economic activities: (0%)	3252 Resin Synth Rubber Fibers Mfg
	4812 Nonscheduled Air Transportation
Tertiary economic activities: (90%)	5259 Other Investment Pools and Funds
3117 Seafood Prod Prep and Packaging	5612 Facilities Support Services
3122 Tobacco Manufacturing	

G	
Primary economic activities: (11%)	3332 Industrial Machinery Manufacturing
1113 Fruit and Tree Nut Farming	3343 Audio and Video Equipment Mfg
1123 Poultry and Egg Production	3346 Mfg, Repr Magnetic, Optical Media
1141 Fishing	3352 Household Appliance Manufacturing
	3361 Motor Vehicle Manufacturing
Secondary economic activities: (4%)	3364 Aerospace Product and Parts Mfg
2122 Metal Ore Mining	3366 Ship and Boat Building
	3369 Other Transportation Equipment Mfg
Tertiary economic activities: (85%)	4854 School, Employee Bus Transportation
3131 Fiber, Yarn, and Thread Mills	4871 Scenic, Sightseeing Transport, Land
3221 Pulp, Paper, and Paperboard Mills	4883 Support Activities Water Transport
3274 Lime and Gypsum Product Mfg	5112 Software Publishers
3313 Aluminum Production and Prcssng	5181 Internet Service Prov, Web Search
3314 Nonferr Metal (no Alum) Prdctn	5331 Lessors Nonfinancial Intang Assets
3321 Forging and Stamping	6112 Junior Colleges
3324 Boiler, Tank, Shipping Container Mfg	6223 Special (no Psych, Sub Abuse) Hosp

The other five place clusters by the error-based criterion

C					
Alamogordo	Belen	Espanola	Grants	Los Lunas	Silver City
Artesia	Deming	Gallup	Las Vegas	Ruidoso	Taos

D				
Aztec	Cedar Crest	Kirtland	Portales	Tijeras
Bernalillo	Clayton	Lordsburg	Raton	Truth Or Cnsqncs
Bloomfield	Corrales	Los Alamos	Santa Rosa	Tucumcari
Bosque Farms	Edgewood	Lovington	Socorro	Tularosa

E					
Arenas Vally	Dexter	Jemez Springs	Mesilla Park	Placitas	Santa Teresa
Bayard	El Prado	Kirtland Afb	Mora	Ranchos De	Sunland Park
Chaparral	Flora Vista	La Luz	Peralta	Sandia Park	Texico

F					
Angel Fire	Des Moines	Grady	Lincoln	Polvadera	Springer
Arrey	Dixon	Grenville	Los Ojos	Quemado	Thoreau
Blanco	Dora	Guadalupita	Maxwell	Ramah	Tinnie
Bosque	Dulce	Hillsboro	Mayhill	Rociada	Tohatchi
Buckhorn	Eagle Nest	Holman	Melrose	Rogers	Ute Park
Cerro	El Rito	Hondo	Monument	Sacramento	Vadito
Chama	Elida	Hope	Mountainair	San Fidel	Vaughn
Chamisal	Embudo	Hurley	Nageezi	San Jose	Veguita
Chimayo	Fence Lake	Isleta	Nara Visa	San Juan Pueb	Wagon Mou
Church Rock	Floyd	Jarales	Navajo Dam	San Patricio	Waterflow
Cleveland	Folsom	La Plata	Paguate	San Rafael	White Sands
Costilla	Fruitland	Laguna	Pena Blanca	San Ysidro	Zuni
Cubero	Garfield	Lake Arthur	Penasco	Santa Cruz	
Datil	Glorieta	Lemitar	Pinon	Shiprock	

G					
Alameda	Casa Blan	Faywood	Loving	Pep	Sedan
Alcalde	Causey	Flying H	Luna	Picacho	Sheep Spr
Algodones	Cerrillos	Fort Sumner	Magdalena	Pie Town	Solano
Alto	Chacon	Fort Wingate	Maljamar	Pinehill	Stanley
Amistad	Cimarron	Gallina	Mc Alister	Pojoaque Val	Sunspot
Animas	Cliff	Gila	Mc Donald	Ponderosa	Tajique
Anthony	Clines Cor	Gladstone	Mc Intosh	Prewitt	Taos Ski Val
Anton Chi	Cloudcroft	Glenwood	Mentmore	Pueblo Of Ac	Tatum
Aragon	Cochiti La	Hagerman	Mescalero	Questa	Tererro
Arroyo Ho	Cochiti Pu	Hanover	Mesilla	Radium Spr	Tesuque
Arroyo Sec	Columbus	Hatch	Mesquite	Red River	Tierra Amaril
Bard	Conchas D	Hernandez	Milan	Regina	Timberon
Bell Ranch	Continental	High Rls Mtn	Milnesand	Rehoboth	Tome
Bent	Cordova	Holloman Afb	Mimbres	Reserve	Tres Piedras
Bingham	Corona	House	Mogollon	Ribera	Vado
Bluewater	Coyote	Jal	Montezuma	Rodeo	Vallecitos
Broadview	Crownpoint	Jamestown	Moriarty	Rowe	Valmora
Buena Vis	Cuba	Jemez Pueblo	Mosquero	Roy	Vanderwagen
Caballo	Cuervo	La Jara	Navajo	Ruidoso Do	Velarde
Cannon A	Dona Ana	La Loma	Newcomb	San Antonio	Villanueva
Canoncito	Elephant Bu	La Mesa	Newkirk	San Jon	Watrous
Canones	Encino	Lakewood	Nogal	San Miguel	Weed
Capitan	Estancia	Lamy	Ocate	Sanostee	Willard
Caprock	Eunice	Lindrith	Ojo Caliente	Santo Dom	Williamsburg
Carrizozo	Fairacres	Loco Hills	Organ	Sapello	Winston
Carson	Fairview	Logan	Pecos	Seboyeta	Yeso

The other five activity clusters by the error-based criterion

C	
Primary economic activities: (2%)	5222 Nondepository Credit Intermediat
1152 Support Activities for Animal Prod	5242 Agencies Brokerages Insurance Activ
	5311 Lessors of Real Estate
Secondary economic activities: (5%)	5312 Offices of Real Estate Agents
2362 Nonresidential Building Construct	5313 Activities Related to Real Estate
2373 Highway, Street, Bridge Construct	5322 Consumer Goods Rental
2381 Foundation, Structure, Bldg Constr	5411 Legal Services
	5412 Acc Tax Prep Bookkeep Payroll Serv
Tertiary economic activities: (93%)	5413 Architectural, Eng, Related Services
3399 Other Miscellaneous Manufacturing	5616 Investigation and Security Services
4238 Machinery, Equip, Supplies Whsle	5617 Services to Buildings and Dwellings
4247 Petroleum, Petroleum Prod Whsle	6111 Elementary and Secondary Schools
4411 Automobile Dealers	6211 Offices of Physicians
4412 Other Motor Vehicle Dealers	6212 Offices of Dentists
4413 Automotive Parts, Acc, Tire Stores	6213 Offices of Other Health Practition
4422 Home Furnishings Stores	6214 Outpatient Care Centers
4431 Electronics and Appliance Stores	6222 Psychiatric, Subs Abuse Hosp
4441 Building Material, Supplies Dealers	6231 Nursing Care Facilities
4442 Lawn Garden Equip Supplies Stores	6241 Individual and Family Services
4452 Specialty Food Stores	6244 Child Day Care Services
4461 Health and Personal Care Stores	7139 Amusement, Recreation Indust
4471 Gasoline Stations	7211 Traveler Accommodation
4481 Clothing Stores	7212 RV Parks and Recreational Camps
4511 Sporting Goods, Hobby, Music Stores	8114 Personal, Household Goods Repair
4512 Book, Periodical, and Music Stores	8123 Drycleaning and Laundry Services
4521 Department Stores	8129 Other Personal Services
4531 Florists	8133 Social Advocacy Organizations
4532 Office Supplies Stationery Gift Stores	8139 Business Profess Labor Politic Org
4533 Used Merchandise Stores	9211 Executive Legisla General Gov Supp
4842 Specialized Freight Trucking	9221 Justice, Public Order, Safety Activ
4884 Support Activities Road Transport	9231 Admin of Human Resource Prog
5191 Other Information Services	9261 Administration of Economic Prog
5221 Depository Credit Intermediation	

D	
Primary economic activities: (2%)	5151 Radio and Television Broadcast
1129 Other Animal Production	5152 Cable Other Subscription Prog
	5171 Wired Telecomm Carriers
Secondary economic activities: (6%)	5172 Wireless Telecomm Carriers
2211 Electric Power Gen, Transm, Distrib	5182 Data Proc, Hosting, Related Serv
2212 Natural Gas Distribution	5223 Activities Related Credit Interm
2213 Water, Sewage and Other Systems	5231 Securities Commodity Contracts
2372 Land Subdivision	5239 Other Financial Investment Activ
	5241 Insurance Carriers
Tertiary economic activities: (92%)	5321 Automotive Equip Rental Leasing
3118 Bakeries and Tortilla Mfg	5323 General Rental Centers
3231 Printing, Related Support Activ	5414 Specialized Design Services
3327 Mach Shop Trnd Screw Nut Bolt Mfg	5415 Computer Systems Design Services
3333 Commerc Serv Industry Machnry Mfg	5416 Management, Scientific, Technical
3339 Other General Purpose Machnry Mfg	5417 Scientific R and D Services
3371 House Inst Furnit Kitchen Cabinet Mfg	5418 Advertising and Related Services
3391 Medical Equipment, Supplies Mfg	5613 Employment Services
4231 Motor Vehicle, Parts, Supplies Whsle	5614 Business Support Services
4233 Lumber, Construct Materials Whsle	5615 Travel Arrange Reserv Services
4234 Prof Commercial Equip Supplies Whsle	5621 Waste Collection
4236 Electrical, Electronic Goods Whsle	6113 Colleges Universities Prof Schools
4237 Hardware Plumb Heating Equip Whsle	6114 Business Schools Computer Mngmt
4239 Misc Durable Goods Merchant Whsle	6116 Other Schools and Instruction
4244 Grocery Related Prod Merchant Whsle	6216 Home Health Care Services
4246 Chem Allied Products Merchant Whsle	6219 Ambulatory Health Care Services
4249 Misc Nondurable Goods Whsle	6221 General Medical, Surgical Hospitals
4251 Whsle Electronic Markets Agents	6233 Community Care Facilities Elderly
4421 Furniture Stores	6243 Vocational Rehabilitation Services
4453 Beer, Wine, and Liquor Stores	7121 Museums, Historical Sites, Similar
4482 Shoe Stores	7223 Special Food Services
4483 Jewelry, Luggage, Leather Goods Stores	7224 Drinking Places (Alcoholic Bev)
4841 General Freight Trucking	8122 Death Care Services
4852 Interurban, Rural Bus Transport	8134 Civic and Social Organizations
4881 Support Activities for Air Transport	9241 Admin Environment Quality Prog
5111 Newspaper, Periodical, Book, Pblshrs	9281 National Security Internat Affairs
5121 Motion Picture and Video Industries	

E	
Primary economic activities: (2%)	3363 Motor Vehicle Parts Manufacturing
1151 Support Activities Crop Production	4232 Furniture, Home Furnishing Whsle
	4235 Metal, Mineral (no Petroleum) Whsle
Secondary economic activities: (7%)	4248 Beer Wine Dist Alcohol Bev Whsle
2111 Oil and Gas Extraction	4529 Other General Merchandise Stores
2131 Support Activities for Mining	4541 Electronic Shopping Mail-Order
2379 Heavy, Civil Engineering Construct	4542 Vending Machine Operators
	4853 Taxi and Limousine Service
Tertiary economic activities: (91%)	4859 Transit Ground Passenger Transport
3116 Animal Slaughtering and Processing	4861 Pipeline Transportation of Crude Oil
3121 Beverage Manufacturing	4882 Support Activities for Rail Transport
3149 Other Textile Product Mills	4931 Warehousing and Storage
3219 Other Wood Product Manufacturing	5611 Office Administrative Services
3241 Petroleum and Coal Products Mfg	5619 Other Support Services
3251 Basic Chemical Manufacturing	5629 Remediation, Waste Mgmt Serv
3323 Architectural, Structural Metals Mfg	6115 Technical and Trade Schools
3328 Coating, Engraving, Heat Treating	6117 Educational Support Services
3329 Other Fabricated Metal Product Mfg	6215 Medical, Diagnostic Laboratories
3331 Agricul, Construct, Mining Mach Mfg	7111 Performing Arts Companies
3334 Vent, Heating, Air, Refrig Equip Mfg	7112 Spectator Sports
3335 Metalworking Machinery Mfg	7113 Promoters Performing Arts, Sports
3344 Semiconductor, Electronic Comp Mfg	8112 Electronic, Precision Equip Repair
3345 Navig, Measrng, Electrmed Instr Mfg	8113 Commerc Indust Mach Equip Repair
3359 Other Electrical Equip, Comp Mfg	

F	
Primary economic activities: (7%)	3325 Hardware Manufacturing
1113 Fruit and Tree Nut Farming	3326 Spring, Wire Product Manufacturing
1114 Greenhouse Nursery Floricult Prod	3336 Eng Turb Power Transm Equip Mfg
1119 Other Crop Farming	3341 Computer and Peripheral Equip Mfg
1121 Cattle Ranching and Farming	3342 Communications Equipment Mfg
1153 Support Activities for Forestry	3351 Electric Lighting Equipment Mfg
	3352 Household Appliance Manufacturing
Secondary economic activities: (1%)	3364 Aerospace Product and Parts Mfg
2123 Nonmetallic Mineral Mining and Qua	3372 Office Furniture (and Fixtures) Mfg
	4241 Paper, Paper Prod Merchant Whsle
Tertiary economic activities: (92%)	4242 Drugs and Druggists' Sundries Whsle
3112 Grain and Oilseed Milling	4243 Apparel, Piece Goods, Notions Whsle
3113 Sugar Confectionery Product Mfg	4543 Direct Selling Establishments
3114 Fruit Veget Preserv Spec Food Mfg	4811 Scheduled Air Transportation
3115 Dairy Product Manufacturing	4821 Rail Transportation
3119 Other Food Manufacturing	4832 Inland Water Transportation
3132 Fabric Mills	4855 Charter Bus Industry
3133 Textile, Finish Fabric Coating Mills	4872 Scenic, Sightseeing Transport, Water
3141 Textile Furnishings Mills	4879 Scenic, Sightseeing Transport, Other
3159 Apparel Accessories, Other Mfg	4885 Freight Transportation Arrangement
3161 Leather Hide Tanning Finishing	4889 Support Activities for Transport
3169 Other Leather, Allied Product Mfg	4921 Couriers
3211 Sawmills and Wood Preservation	5122 Sound Recording Industries
3222 Converted Paper Product Mfg	5173 Telecommunications Resellers
3253 Pesticide Fertilizer Agric Chem Mfg	5179 Other Telecommunications
3254 Pharmaceutical and Medicine Mfg	5251 Insurance, Employee Benefit Funds
3255 Paint, Coating, Adhesive Mfg	5324 Commerc Industri Mach Equip Rent
3256 Soap Cleaning Comp Toilet Prep Mfg	5511 Management of Companies, Enterpr
3259 Other Chemical Product, Prep Mfg	5622 Waste Treatment and Disposal
3261 Plastics Product Manufacturing	6239 Other Residential Care Facilities
3262 Rubber Product Manufacturing	6242 Community Food Housing Relief
3271 Clay Product and Refractory Mfg	7115 Independent Artists, Writers, Perf
3272 Glass and Glass Prod Mfg	7131 Amusement Parks and Arcades
3273 Cement and Concrete Product Mfg	7132 Gambling Industries
3279 Other Nonmetallic Mineral Prod Mfg	7213 Rooming and Boarding Houses
3312 Steel Prod Mfg from Prchsd Steel	7221 Full-Service Restaurants
3315 Foundries	8132 Grantmaking and Giving Services
3321 Forging and Stamping	9251 Admin Housing Prog Urban Plan

G	
Primary economic activities: (10%)	3322 Cutlery, Handtool Mfg
1123 Poultry and Egg Production	3324 Boiler, Tank, Shipping Cont Mfg
1133 Logging	3332 Industrial Machinery Mfg
1141 Fishing	3343 Audio and Video Equipment Mfg
1142 Hunting and Trapping	3346 Mfg Reprod Magnetic Optical Media
	3353 Electrical Equipment Mfg
Secondary economic activities: (2%)	3361 Motor Vehicle Mfg
2122 Metal Ore Mining	3362 Motor Vehicle Body and Trailer Mfg
	3366 Ship and Boat Building
Tertiary economic activities: (88%)	3369 Other Transportation Equipment Mfg
3111 Animal Food Manufacturing	3379 Other Furniture Related Product Mfg
3117 Seafood Product Prep and Pack	4245 Farm Prod Raw Mat Merchant Whsle
3122 Tobacco Manufacturing	4812 Nonscheduled Air Transportation
3131 Fiber, Yarn, and Thread Mills	4851 Urban Transit Systems
3151 Apparel Knitting Mills	4854 School and Employee Bus Transport
3152 Cut and Sew Apparel Mfg	4871 Scenic, Sightseeing Transport, Land
3162 Footwear Manufacturing	4883 Support Activities Water Transport
3212 Veneer, Plywood, Eng Wood Prod	5112 Software Publishers
3221 Pulp, Paper, and Paperboard Mills	5181 Internet Serv Providers Web Search
3252 Resin, Synth Rubber, Fibers Mfg	5259 Other Investment Pools and Funds
3274 Lime and Gypsum Product Mfg	5331 Lessors Nonfinancial Assets
3311 Iron, Steel Mills, Ferroalloy Mfg	5612 Facilities Support Services
3313 Alumina, Aluminum Production	6112 Junior Colleges
3314 Nonferr Metal (no Alum) Production	6223 Special (no Psych Subs Abuse) Hosp

Appendix B

Point Process

A point process is a type of stochastic process (or random process)—an uncertain evolution over time—that models a set of isolated points (random events) in time. For example, a sequence of blood pressure measurements taken for a patient during his/her stay at the hospital may be modeled by a point process.

A point process can be described by its *conditional intensity function*, $\lambda(t, H_t)$ (Daley and Vere-Jones, 2003):

$$\lambda(t, H_t) = \lim_{\Delta t \to 0} \frac{\Pr(N_{(t,t+\Delta t]} = 1 | H_t)}{\Delta t},$$

where $\Pr(N_{(t,t+\Delta t]} = 1 | H_t)$ is the probability of an event occurring in the time interval $(t, t + \Delta t]$ and H_t is the history of events up to time t. The conditional intensity is the conditional instantaneous probability of an event occurring.

Given such a conditional intensity function, we can sample a sequence of events as follows. Starting with current time $t = t_0$, we draw a sample event from the exponential distribution with the rate given by the conditional intensity function for time t and history

H_t, $\lambda(t, H_t)$. This produces a new event at time $t = t_1$ determined by the exponential distribution. The conditional intensity function may change between t_0 and t_1 as time progresses. If the conditional intensity function changes before t_1, the sampled event is not taken and instead a new sample is drawn according to the new rate. If the new sample occurs before the rate changes again, it is retained and the sampling process continues from this time. We repeat this process until a desired end time is reached.

Appendix C

CHLA PICU Data Set Variables

Continuous Variables used to Test Alternative Methods	
Vital Signs	**Interventions**
Abdominal girth (cm)	Amplitude (HFOV) (deltaP)
Best motor response (GCS)	EPAP (cmH2O)
Best verbal response (GCS)	FiO2
Bladder pressure (mmHg)	Frequency (HFOV) (Hz)
Capillary refill rate (sec)	IPAP (cmH2O)
Diastolic blood pressure non-invasive (mmHg)	Inspiratory time (sec)
EtCO2 (mmHg)	MAP (HFOV) (cmH2O)
Eye opening response (GCS)	Mean airway pressure (cmH2O)
Glasgow coma scale total	NIV set rate (bpm)
Heart rate (bpm)	O2 Flow (LPM)
Intracranial pressure (mmHg)	PEEP (cmH2O)
Left pupillary response	Peak Inspiratory Pressure (cmH2O)
Pulse oximetry (%)	Pressure support (cmH2O)
Respiratory rate (bpm)	Tidal volume delivered (ml)
Right pupillary response	Tidal volume expiratory (ml)
Systolic blood pressure non-invasive (mmHg)	Tidal volume inspiratory (ml)
Temperature (C)	Tidal volume set (ml)
	Ventilator rate (bpm)

Demographics
Binary: Sex **Categorical:** **Continuous:**

Encounter Information
Binary: Pre ICU CPR **Categorical:** **Continuous:** Age

Vital Signs	
Binary:	Glasgow coma scale total
	Head circumference (cm)
Categorical:	Heart rate (bpm)
Central Venous Pressure (mmHg)	Height (cm)
	Intracranial pressure (mmHg)
Continuous:	Left pupillary response
Abdominal girth (cm)	Post-ductal pulse oximetry (%)
Best motor response (GCS)	Pulse oximetry (%)
Best verbal response (GCS)	Pupillary response
Bladder pressure (mmHg)	Respiratory rate (bpm)
Capillary refill rate (sec)	Right pupillary response
Cerebral perfusion pressure (mmHg)	EtCO2 (mmHg)
Diastolic blood pressure invasive (mmHg)	Eye opening response (GCS)
Diastolic blood pressure non-invasive (mmHg)	Temperature (C)
Systolic blood pressure invasive (mmHg)	Weight (kg)
Systolic blood pressure non-invasive (mmHg)	Weight for drug calculations (kg)

Laboratory Tests	
Binary:	CBG pH
ABG Base excess (mEq/L)	CSF Bands %
ABG FiO2	CSF Lymphs %
ABG HCO3 (mEq/L)	CSF RBC
ABG O2 sat (%)	CSF Segs %
ABG PCO2 (mmHg)	CSF WBC
ABG PO2 (mmHg)	CSF color
ABG TCO2 (mEq/L)	CSF glucose (mg/dL)
ABG pH	CSF protein (g/dL)
ALT (SGPT) (units/L)	Calcium ionized (mg/dL)
AST (SGOT) (units/L)	Calcium total (mg/dL)
Adenovirus (RVP)	Chloride (mEq/L)
Albumin level (g/dL)	Complement C3 serum (mg/dL)
Alkaline phosphatase (units/L)	Creatinine (mg/dL)
Amylase (units/L)	D-dimer (mg/L FEUA)
B-type Natriuretic Peptide (pg/mL)	ESR
BUN (mg/dL)	Eosinophils %
Bands %	FDP Titer
Basophils%	Ferritin level (ng/mL)
Bicarbonate serum (mEq/L)	Fibrinogen (mg%)
Bilirubin conjugated (mg/dL)	GGT (units/L)
Bilirubin total (mg/dL)	Glucose (mg/dL)
Bilirubin unconjugated (mg/dL)	Haptoglobin (mg/dL)
Blasts	Hematocrit POC (%)
C-reactive protein (mg/dL)	Hematocrit blood (%)
CBG Base excess	Hemoglobin POC (g/dL)
CBG FiO2	Hemoglobin blood (g/dL)
CBG HCO3 (mEq/L)	INR
CBG O2 sat (%)	Influenza A (RVP)
CBG PCO2 (mmHg)	Influenza A H1 (RVP)
CBG PO2 (mmHg)	Influenza A H3 (RVP)
CBG TCO2 (mEq/L)	Influenza B (RVP)

Laboratory Tests (continued)	
Lactate (mg/dL)	Potassium POC (mEq/L)
Lactate Dehydrogenase blood (units/L)	Potassium serum (mEq/L)
Lactic Acid blood (mg/dL)	Protein total (g/dL)
Lipase (units/L)	RBC blood (M/uL)
Lymphocyte %	RDW (%)
MCH (pg)	RSV A (RVP)
MCHC (%)	RSV B (RVP)
MCV (fL)	Reticulocyte count (%)
MVBG Base excess (mEq/L)	Rhinovirus (RVP)
MVBG FiO2	Schistocytes
MVBG HCO3 (mEq/L)	Sodium POC (mEq/L)
MVBG O2 sat (%)	Sodium serum (mEq/L)
MVBG PCO2 (mmHg)	Spherocytes
MVBG PO2 (mmHg)	T4 free (ng/dL)
MVBG pH	TSH (mlU/L)
Macrocytes	Triglycerides (mg/dL)
Magnesium level (mg/dL)	VBG Base excess (mEq/L)
Metamyelocytes %	VBG FiO2
Metapneumovirus (RVP)	VBG HCO3 (mEq/L)
Monocytes %	VBG O2 sat (%)
MVBG TCO2 (mEq/L)	VBG PCO2 (mmHg)
Myelocytes %	VBG PO2 (mmHg)
Neutrophils %	VBG TCO2 (mEq/L)
Oxygentaion index	VBG pH
P/F ratio	White blood cell count (K/uL)
PT	
PTT	**Categorical:**
Parainfluenza 1 (RVP)	
Parainfluenza 2 (RVP)	**Continuous:**
Parainfluenza 3 (RVP)	
Phosphorus level (mg/dL)	
Phosphorus level (mg/dL)	
Platelet count (K/uL)	

Drugs	
Binary:	Cefotaxime
Acetaminophen	Cefoxitin
Acetaminophen/Codeine	Ceftazidime
Acetaminophen/Hydrocodone	Ceftriaxone
Acetazolamide	Cefuroxime
Acetylcysteine	Cephalexin
Acyclovir	Chloral Hydrate
Albumin	Chlorothiazide
Albuterol	Ciprofloxacin HCL
Allopurinol	Cisatracurium
Alteplase	Clarithromycin
Amikacin	Clindamycin
Aminocaproic Acid	Clonazepam
Aminophylline	Clonidine HCl
Amiodarone	Clotrimazole
Amlodipine	Cromolyn Sodium
Amoxicillin	Cyclophosphamide
Amoxicillin/clavulanic acid	Cyclosporine
Amphotericin B	Dantrolene Sodium
Amphotericin B Lipid Complex	Desmopressin
Ampicillin	Dexamethasone
Ampicillin/Sulbactam	Dexmedetomidine
Aspirin	Diazepam
Atenolol	Digoxin
Atropine	Diphenhydramine HCl
Azathioprine	Dobutamine
Azithromycin	Dopamine
Baclofen	Dornase Alfa
Basiliximab	Doxacurium Chloride
Budesonide	Doxycycline Hyclate
Bumetanide	Enalapril
Calcium Chloride	Enoxaparin
Calcium Glubionate	Epinephrine
Calcium Gluconate	Epoetin
Captopril	Erythromycin
Carbamazepine	Esmolol Hydrochloride
Carvedilol	Etomidate
Caspofungin	Factor VII
Cefazolin	Famotidine
Cefepime	Fentanyl

Drugs (continued)	
Ferrous Sulfate	Meropenem
Ferrous Sulfate	Methadone
Filgrastim	Methylprednisolone
Flecainide Acetate	Metoclopramide
Fluconazole	Metolazone
Fluticasone	Metronidazole
Fosphenytoin	Micafungin
Furosemide	Midazolam HCl
Gabapentin	Milrinone
Ganciclovir Sodium	Montelukast Sodium
Gentamicin	Morphine
Glycopyrrolate	Mycophenolate Mofetl
Haloperidol	Naloxone HCL
Heparin	Naproxen
Hydrocortisone	Nesiritide
Hydromorphone	Nifedipine
Ibuprofen	Nitrofurantoin
Imipenem	Nitroglycerine
Immune Globulin	Nitroprusside
Insulin	Norepinephrine
Ipratropium Bromide	Nystatin
Isoniazid	Octreotide Acetate
Isoproterenol	Olanzapine
Isradipine	Ondansetron
Itraconazole	Oseltamivir
Ketamine	Oxacillin
Ketorolac	Oxcarbazepine
Labetalol	Oxycodone
Lactobacillus	Pancuronium
Lansoprazole	Pantoprazole
Levalbuterol	Penicillin G Sodium
Levetiracetam	Pentobarbital
Levocarnitine	Phenobarbital
Levofloxacin	Phenylephrine HCl
Levothyroxine Sodium	Phenytoin
Lidocaine	Piperacillin
Linezolid	Piperacillin/Tazobactam
Lisinopril	Potassium Chloride
Lorazepam	Potassium Phosphate
Magnesium Sulfate	Prednisolone

Drugs (continued)	
Prednisone	Ticarcillin/clavulanic acid
Procainamide	Tobramycin
Propofol	Topiramate
Propranolol HCl	Treprostinil
Prostacyclin	Trimethoprim/Sulfamethoxazole
Racemic Epi	Tromethamine (THAM)
Ranitidine	Ursodiol
Rifampin	Valganciclovir
Risperidone	Valproic Acid
Rocuronium	Vancomycin
Sildenafil	Vasopressin
Sodium Bicarbonate	Vecuronium
Sodium Chloride	Vitamin E
Sodium Phosphate	Vitamin K
Spironolactone	Voriconazole
Sucralfate	
Tacrolimus	**Categorical:**
Terbutaline	
Theophylline	**Continuous:**
Ticarcillin	

Interventions	
Binary:	MAP (HFOV) (cmH2O)
Abdominal X ray	MRI brain
Amplitude (HFOV) (deltaP)	Mean airway pressure (cmH2O)
Arterial line site	Mechnical ventilation mode
Arterial line waveform (duration)	NIV Mode
CT abdomen	NIV set rate (bpm)
CT abdomen/pelvis	Nitric Oxide
CT brain	O2 Flow (LPM)
CT chest	Oxygen mode
CT pelvis	PEEP (cmH2O)
Central venous line site	Peak Inspiratory Pressure (cmH2O)
Central venous line waveform (duration)	Peritoneal dyalisis
Chest X ray	Pressure support (cmH2O)
Chest tube	Tidal volume delivered (ml)
Chest/abd X ray	Tidal volume expiratory (ml)
Continuous EEG	Tidal volume inspiratory (ml)
ECMO hours	Tidal volume set (ml)
ECMO type	Tracheostomy
EPAP (cmH2O)	Ventilator rate (bpm)
FiO2	Ventriculostomy site
Foley catheter	Ventriculostomy waveform (duration)
Frequency (HFOV) (Hz)	
Gastrostomy tube	**Categorical:**
Hemofiltration/CRRT	
IPAP (cmH2O)	**Continuous:**
Inspiratory time (sec)	