University of Nevada, Reno

Using Pre- and Post-Process Labeling Techniques for Cluster Analysis

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering

by

Damien Ennis

Dr. Frederick C. Harris, Jr./Dissertation Advisor

December, 2014

THE GRADUATE SCHOOL

We recommend that the dissertation
prepared under our supervision by

DAMIEN ENNIS

Entitled

Using Pre- and Post-Process Labeling Techniques for Cluster Analysis

be accepted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

Frederick C. Harris, Jr., Advisor

Sergiu Dascalu, Committee Member

Theodore Lambert, Committee Member

Dwight Egbert, Committee Member

Anna R. Panorska, Graduate School Representative

David W. Zeh, Ph. D., Dean, Graduate School

December, 2014

Abstract

As the amount and variety of data increases through technological and investigative advances, the means to analyze and manage this data becomes more critical. Unsupervised machine learning algorithms can be used to group large datasets into categories, thereby facilitating new insight into similarities in data that would, on the surface, appear to be disparate. This dissertation investigates three unsupervised clustering algorithms: the self-organizing map, the K-means algorithm, and affinity propagation. These three algorithms are applied to large datasets from three different domains—organizational management, bioinformatics, and financial markets—each of which presents its own challenges in terms of data management and knowledge discovery. Specifically, the self-organizing map is used to cluster a variety of academic library data to show how it can be used to aid in operational and strategic decision-making. Both a self-organizing map and the K-means algorithm are used to cluster genomic data to show how they can be used to identify possible organisms that are present in a metagenomic sample. Affinity propagation is used to cluster stock performance data to show how it can be used to aid in making investment decisions. In addition, different semi-supervised labeling techniques are employed in combination with these clustering algorithms to assist with knowledge discovery in these three areas. The applicability of these different labeling techniques for various types of problems is discussed, and the success of these combinations in facilitating several types of data analysis is explored, providing researchers with guidance about the applicability of these strategies.

Acknowledgments

I'd like to thank my advisor Frederick Harris, who has been both a friend and a mentor to me for many years. His guidance throughout my studies has been invaluable. I'd also like to thank the members of my committee, as well as the faculty of the Computer Science and Engineering Department, for the support and intellectual challenges that they have provided me. Finally, I'd like to thank my friends and family for their constant encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Because human investigation of large datasets can be both costly and time-consuming, new solutions are needed for data analysis. And the problem is not merely one of quantity: In a survey of recent advances in clustering, Anil K. Jain observed that, "The increase in both the volume and the variety of data requires advances in methodology to automatically understand, process, and summarize the data" [34]. Fortunately, techniques of artificial intelligence are allowing computers to solve otherwise intractable problems with efficiency.

One subfield of artificial intelligence is machine learning, which combines aspects of probability and statistics with computer science. Machine learning uses evolving algorithms that improve their performance on the basis of feedback. The techniques used in machine learning focus on making computers adapt over time to be more accurate in performing an action, whether this action is the classification of a piece of data, the prediction of an outcome, or the making of a decision for some autonomous agent. Consider a computer-controlled enemy in a video game that is adapting to the behavior it encounters. When the player initially confronts this enemy, he can easily

defeat it. However, on subsequent matches, the computer-controlled enemy has refined its technique and become substantially more challenging—if not unbeatable.

Machine learning is generally categorized into two separate approaches: supervised learning and unsupervised learning. With supervised learning, an algorithm is trained against known input-output pairs before being fed unknown inputs. Once trained on known data, the algorithm is applied to unknown inputs to elucidate useful information. Examples of this include a neural network and a support vector machine [50]. Unsupervised machine learning acquires "correct" outputs by adapting the algorithm's parameters solely on the input data. That is, the algorithm refines itself in an iterative manner based only on the raw data. Similar to supervised learning, this is often done in two phases, with a distinct set of inputs for training and a separate set for hypothesis testing. Examples of unsupervised learning algorithms include the self-organizing map (SOM), the K-means algorithm, and affinity propagation [50, 52], techniques which will be explored in this dissertation.

One common usage of unsupervised machine learning is in clustering complex datasets. Unsupervised machine learning can be used with problems in which there is a vast amount of data and in which each data point consists of a vector of many features. In these cases, a clustering technique can facilitate new insight into the similarity of data that would, on the surface, appear to be disparate. Clustering algorithms have been successfully applied within many fields, including biology, marketing, and seismic analysis. A subcategory of clustering algorithms, semi-supervised clustering, involves prior known data observations, which are used to guide the examination of the clustered data [11]. This dissertation describes different techniques for using processes for labeling

data in a semi-supervised fashion in combination with clustering algorithms, including the complete and partial labeling of data occurring at different stages within the clustering process. The applicability of these different labeling techniques for various types of problems is discussed, and the success of these combinations in facilitating data analysis is explored.

This dissertation is structured as follows: Chapter 2 presents a discussion of unsupervised and semi-supervised machine learning clustering techniques, as well as their limiting factors. In Chapter 3, clustering's applications to various domains are discussed, and a survey of the related literature is presented. These domains include organizational decision-making, bioinformatics, and financial markets. In addition, previous work that has led to the use of the original labeling techniques used in this dissertation are discussed as well. Chapter 4 presents the use of an SOM to cluster data for use in operational decision-making of an academic library. In order to look for areas of high performance, analysis of the library data was conducted through the application of a “library performance metric” in conjunction with the SOM. Analysis of the map provides value in elucidating qualities of libraries that are performing well based on the metric. This application of an SOM demonstrates how it can be used to facilitate organizational comparisons and potentially lead to more informed decision-making. Chapter 5 describes the usage of two clustering techniques, the SOM and the K-means algorithm, to cluster data from an environmental sample collected from a hot springs habitat and to provide a visual analysis of that data. A project pipeline is described that uses an unsupervised clustering algorithm to identify which reference genomes should be included for further analysis in determining possible organisms that are present in a

metagenomic sample. The labeling is done post-clustering to represent the data in a concise manner in order to aid biological investigators. Chapter 6 presents the use of the affinity propagation algorithm in the analysis of stock data. Here, the labeling is applied to a small subset of the overall data to recognize stocks that cluster with known successful investments and to provide a method for possible stock selection. Chapter 7 contains a comparison of the efficacy of the clustering techniques in combination with the various labeling strategies and suggests directions for further research.

Chapter 2

Background

2.1 Unsupervised Machine Learning

The goal of unsupervised machine learning is to discover "interesting structure" in the provided data [52]. With unsupervised learning there is no preconceived notion of what the correct output is in response to a given input; instead, the process is an attempt to find previously unrecognized common or interesting features that may be present. This type of learning is similar to what is seen in biological organisms, including humans. In fact, one technique discussed below, the SOM, was developed when its inventor was considering the question of how sensory signals are mapped into the cerebral cortex of the brain [50]. Unsupervised techniques can be more easily applied than supervised learning since they require no previous domain knowledge to perform this analysis.

2.2 Clustering Techniques

Clustering data is a principal application of unsupervised machine learning [52], and several algorithms exist to perform this task. Clustering is the process of grouping “like” pieces of data into one of several bins. Aggarwal and Reddy succinctly describe

the basic problem that clustering addresses: "Given a set of data points, partition them into a set of groups which are as similar as possible" [3]. This is often based on overall similarity (Euclidian distance or some other metric), but other qualities could also serve as the guideline for this binning process [50]. The goal of a clustering algorithm is to group raw (unlabeled) data on the basis of some underlying features that are not readily apparent. Each cluster represents a collection of like items (data) with similar features, whereas some implied dissimilarity exists between items in different clusters. This can be used to directly elucidate new patterns in complex data, but may also serve to reduce the scope of subsequent analysis, as will be illustrated in Chapter 5. Common problems in which clustering is beneficial are as an intermediate step for data mining problems, collaborative filtering, customer segmentation, data summarization, dynamic trend detection, multimedia data analysis, biological data analysis, and social network analysis [3]. In further chapters, some of these problems will be directly addressed, including the use of clustering as an intermediate step for data mining and as a tool for biological data analysis.

In the remainder of this chapter, three techniques will be highlighted that are especially promising for working with large datasets: the SOM, the K-means algorithm, and affinity propagation. These particular three algorithms have been successfully applied in many fields; two of the three represent historically successful algorithms, while the third is a more recently developed technique, which provides an interesting point of comparison. These techniques have different strengths and levels of computational complexity that dictate their usage in regard to specific domains and data

sources. In all cases these algorithms are quite scalable, require limited domain knowledge, and are capable of evaluating datasets of differing types and sizes.

2.2.1 Self-Organizing Map

The SOM was originally developed by Teuvo Kohonen [41], who described it as a tool that is "able to convert complex, nonlinear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display." An SOM is suitable for problems that involve detecting correlations, reducing dimensionality, finding hidden patterns, and classifying data [42]. An SOM synthesizes data that consists of many variables and produces as its output a simplified view of the data that consists of a regular grid or lattice of neurons—usually hexagonal or rectangular in structure. In the grid, each piece of data is assigned to the neuron with the weight vector that most closely matches the data's feature vector. Input data items with similar features are placed in close proximity on the grid. Often, this proximity on the grid illustrates an interesting property of the dataset, so an SOM allows one to see at a glance which data items are related in a way that otherwise would not be readily apparent [42]. Both Chapters 4 and 5 discuss in some detail the application of this algorithm in two distinct domains.

An SOM differs from the more traditional supervised neural network models in that it requires no pre-training on input-output pairs. As an unsupervised, competitive learning algorithm (i.e., in which neurons compete for each input vector and are updated based on the data they acquire), an SOM is not attempting to find previously determined

“correct” outputs. Instead, the algorithm is used to find similarities between different inputs, and these similarities are visualized through the resulting map [42, 50].

An SOM provides a method of mapping data from a higher-dimensional space to a lower-dimensional output space (e.g., mapping data with ten features onto a two-dimensional map). An SOM is most commonly configured as a two-dimensional array of neurons because it provides an easy visualization of information for investigation. The SOM's neurons are connected to their immediate neighbors in a grid pattern. Figure 2.1 shows two input data points which are mapped to a rectangular lattice, with each square representing a neuron on the map [42, 50].

Each neuron’s weight vector (i.e., the collection of numerical data associated with a neuron) is modified by the collected numerical data from the input points. Figure 2.2 demonstrates that the SOM algorithm has relative ordering preservation, which Marsland describes as follows: "Inputs should be preserved by the ordering in the neurons so that neurons that are close together represent inputs that are close together, while neurons that are far apart represent inputs that are far apart" [50].

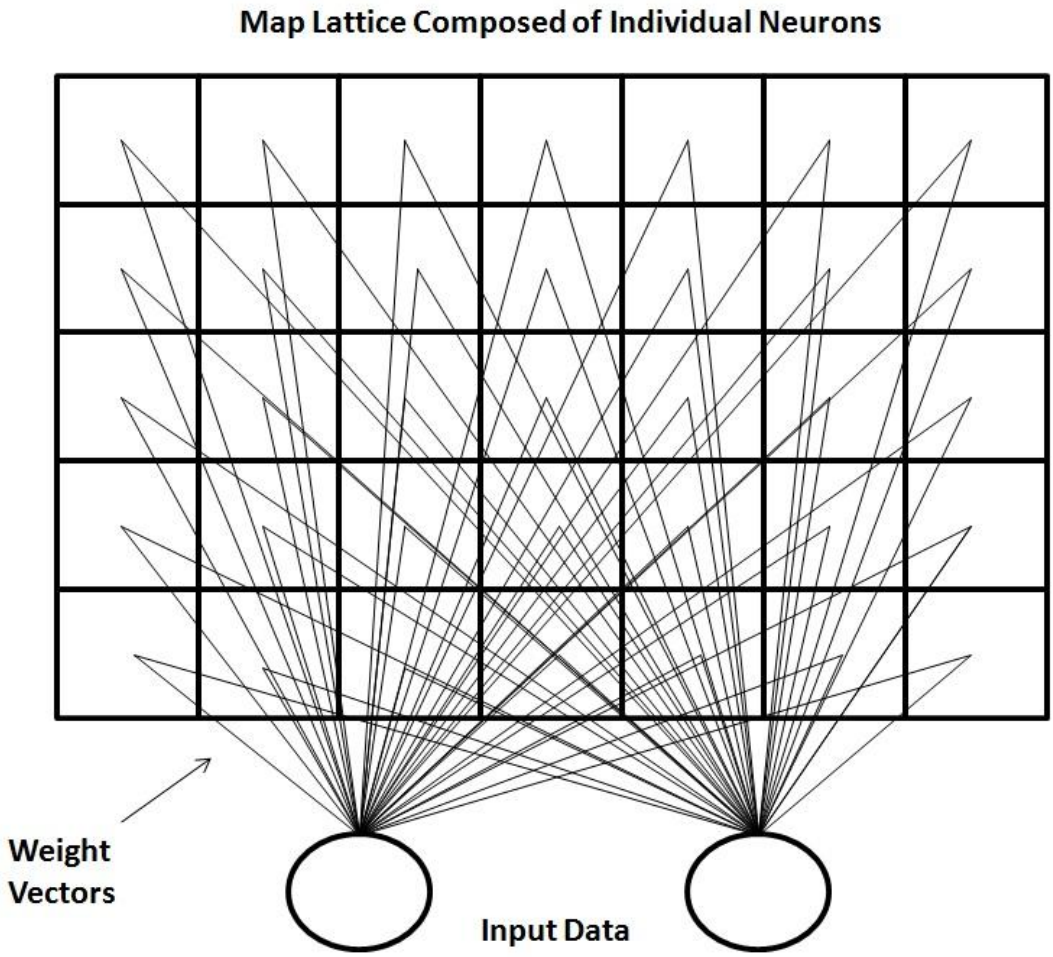


Figure 2.1: An Illustration Showing the Dimensionality of the SOM Data [22]

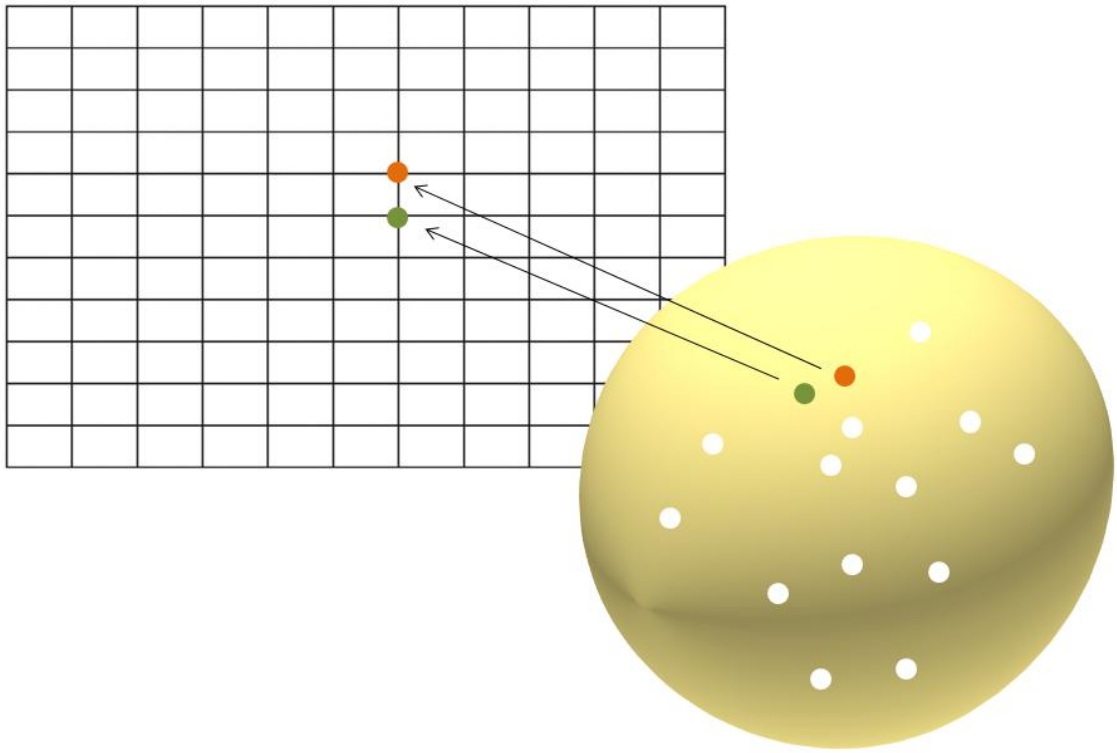


Figure 2.2: A Schematic Showing the Relationship between the Location of Input Data and their Assignment to Proximal Output Neurons in an SOM

Through a weight vector, each neuron in an SOM is connected to the input layer. The dimensionality of the weight vector matches the number of features of the input data. Each neuron's weights are set to an initial random value within a designated range. The *N*-dimensional weight vector is defined as follows:

$$w_i = [w_1, w_2, \ldots, w_n] \quad (1)$$

The SOM is executed in three steps: acquiring and normalizing the data, training the SOM on the supplied data, and then gaining insight regarding the data from the trained SOM. The training process is iterative, with the assignment of input vectors resulting in adjustments to the map. During the training phase an input vector is selected

at random from the training dataset and presented to the map of neurons [41]. The neuron whose weight vectors most closely matches the input data as determined by Euclidian distance is selected as the winning neuron. This is defined as:

$$\text{winning neuron } c = \arg\min \|x - w_i\| \quad (2)$$

Once the winning neuron has been selected, the weight vectors of the winning neuron and the neurons that reside in its neighborhood are adjusted to more closely match the current input. The smaller the Euclidian distance is between the input vector and winning neuron, the larger the change to its current weight values:

$$w_i(t+1) = w_i(t) + \alpha(t)h_{ci}(t)[x(t) - w_i(t)] \quad (3)$$

where:
$x(t)$ is the input vector randomly drawn from the input set at time t
$\alpha(t)$ is the learning rate function
$h_{ci}(t)$ is the neighborhood function centered on the winning neuron at t

Murtagh describes this process in the following manner: "The SOM method makes the surface of neurons resonate . . . in accordance with the outside world as represented by the input vectors" [53]. Both the learning rate represented by the size of the change to the weight vectors and the neighborhood radius decrease over the number of iterations, which is represented through a Gaussian function:

$$h_{ci}(t) = \exp\left[\frac{-\|r_c - r_i\|^2}{2\sigma(t)^2}\right] \quad (4)$$

where:
$\sigma(t)$ is the width of the Gaussian kernel
$-\|r_c - r_i\|^2$ is the distance between the winning neuron c and the neuron i with r_c and r_i representing the two-dimensional positions of neurons c and i on the SOM grid

Vectors continue to be chosen randomly from the training data, and the steps above are repeated until changes to the weight values in a given iteration are below a specific threshold [42, 50].

2.2.2 K-means

The K-means algorithm represents the most popular and simplest partitional algorithm [34]. It has been in existence for over 50 years and was discovered by several different researchers working independently in different fields [34]. It is one of the most widely used clustering algorithms due to its "ease of implementation, simplicity, efficiency, and empirical success" [34]. It has been widely applied in many fields, and represents a type of exclusive clustering that places the data objects into a pre-set number of classes. Hartigan and Wong describe the goal of K-means as "to divide *M* points in *N* dimensions into *K* clusters so that the within-cluster sum of squares is minimized" [29].

Like an SOM, application of the K-means algorithm also proceeds in three main phases. The first phase is initialization, which involves choosing the number of clusters *k*. Some extensions to the algorithm include dynamic selection of the number of clusters. When employing this technique, Hartigan suggests running the algorithm with differing values for *k* and evaluating the variance in each case to determine the most advantageous number of clusters [28]. Once the number of clusters has been determined, *k* random vectors are chosen with the same dimensionality and value ranges as the input data. These values are then assigned to serve as the cluster centers u_j. The second phase represents the unsupervised learning, which involves an iterative process similar to that used with an SOM, whereby each input data item x_i is compared against the cluster

centers and then assigned to the one that it is closest to in terms of Euclidean distance (although certain variants of the algorithm use different metrics). As a result of using a Euclidian metric, the K-means algorithm finds ball-shaped clusters in data [34]. This is represented by:

$$d_i = \min_j d(x_i, \mu_j) \tag{5}$$

As a new data item is assigned to a particular cluster, that cluster center point is recalculated as the mean of all currently assigned data items, according to the following equation:

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i \tag{6}$$

Figure 2.3 provides a graphical representation of the updating of the cluster centers.

The process continues until such time that the center values do not update. In the final phase of the process, the clustered data that has minimized the sum of the squared error over all *k* clusters can be used for further analysis after the cluster center points have stabilized [28, 50].

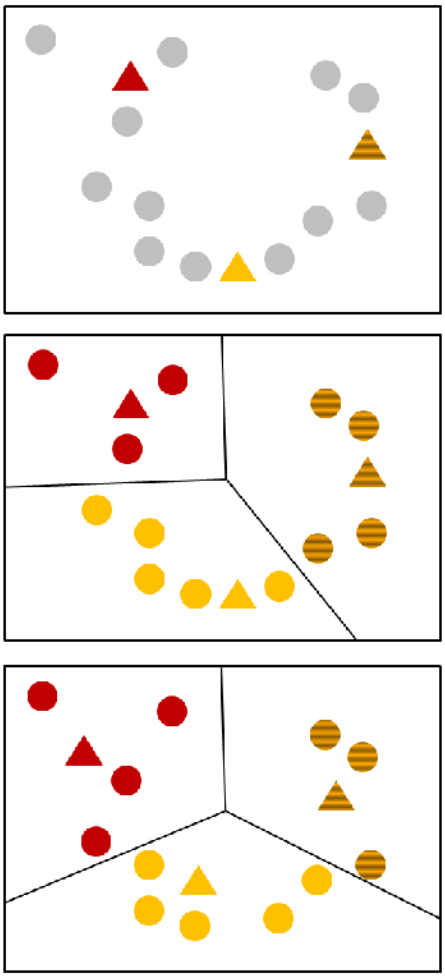


Figure 2.3: A Graphical Representation of the K-Means Algorithm Showing the Updating of Cluster Centers and Subsequent Clusters

2.2.3 Affinity Propagation

Affinity propagation is a clustering technique that was developed by Brendan Frey and Delbert Deuck in 2007 and is based on passing information between data points [23]. Unlike the K-means and SOM algorithms (where the dimensions of the grid of neurons is chosen in advance), the number of clusters does not need to be decided prior to application. Similar to the two previous algorithms discussed, this technique relies upon randomly selecting initial exemplars and then refining them iteratively to be

representative of the clustered data. The initial exemplars are actual data points rather than random weights, as in the example of the SOM [23]. Input is the *N x N* collection of similarity values between each pair of data points. This input collection is created by running an all-versus-all comparison as a preprocessing step.

One of the limitations of both K-means and the SOM is their sensitivity to the initial selection of both the number of clusters and the starting midpoint/exemplar values. Affinity propagation takes the approach that simultaneously considers all data points as potential exemplars [23]. Each data point is viewed as a network node. Messages are recursively transmitted along the network edges until an optimal set of clusters is achieved.

The input to affinity propagation takes the form *s(i,k)*, which indicates how well data point *k* is as an exemplar for data point *i*. As stated previously, the number of clusters is not pre-specified; instead, preferred data points are selected based on the *s*(*k*,*k*) value.

The algorithm proceeds by exchanging messages between nodes. Two types of messages are sent: responsibility and availability. Responsibility, represented by *r*(*i*,*k*), informs candidate exemplar *k* as to how well-suited it is to serve as the exemplar for data point *i*. Availability, represented by *a*(*i*,*k*) and sent from the exemplar *k* to data point *i*, states how appropriate it would be for data point *i* to choose *k* as its exemplar. Combining these messages results in the decisions of which points should be exemplars and, for all other data points, which exemplar they should be assigned to. Essentially, each data point chooses a data point to serve as its exemplar/centroid. Data points may choose themselves, which increases the number of clusters [52].

Affinity propagation attempts to maximize Formula 7. Let $c_i \in \{1,...,N\}$ represent the centroid for datapoint *i*:

$$S(c) = \sum_{i=1}^{N} s(i, c_i) + \sum_{k=1}^{N} \delta_k(c) \qquad [52] \qquad (7)$$

The first term is a measurement of similarity of each piece of data to the centroid, whereas the second term is a choice penalty formally described as:

$$\delta_k(c) = \begin{cases} -\infty c_k \neq k \text{ but } \exists i : c_i = k \\ 0 \quad otherwise \end{cases} \qquad (8)$$

Figures 2.4 and 2.5 show how the two types of messages are passed along the edges of the network.

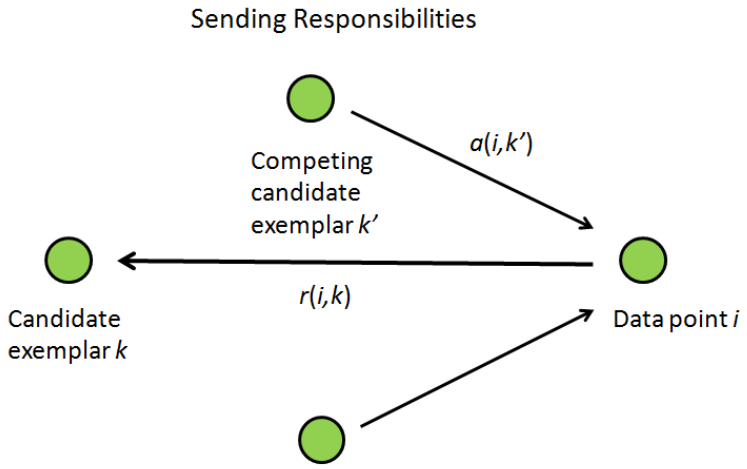


Figure 2.4: Sending Responsibilities in Affinity Propagation (Adapted from [23])

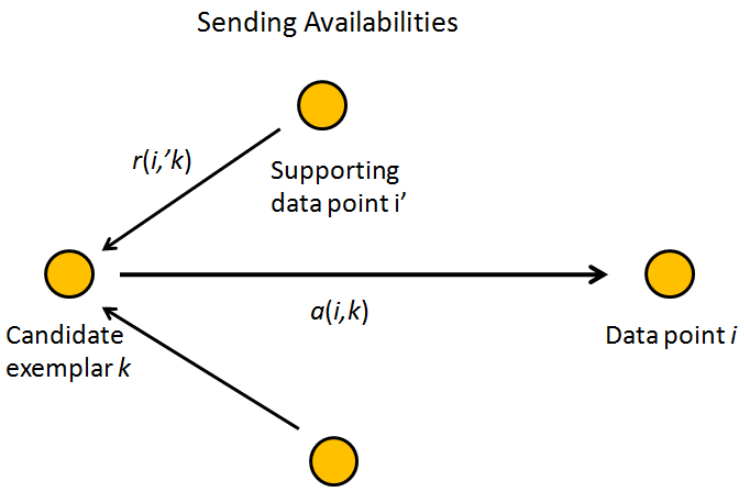


Figure 2.5: Sending Availabilities in Affinity Propagation (Adapted from [23])

Every data point is represented by a node and through a sequence of message passing, the exemplars and corresponding clusters are determined.

2.2.4 Other Clustering Techniques

In addition to the techniques highlighted above, several other clustering techniques exist with varying strengths and weaknesses. These other clustering algorithms include but are not limited to DBSCAN, BIRCH, CURE, and the Fuzzy *c*-means algorithm [77]. These techniques can be very broadly categorized into two different types: hierarchical clustering and partitional clustering [35, 77]. Jain *et al.* describe these different strategies for approaching the problem of clustering: "Hierarchical clustering algorithms produce a nested series of partitions based on a criterion for merging or splitting clusters based on similarity. Partitional clustering algorithms identify the partition that optimizes (usually locally) a clustering criterion" [35]. Other classifications of clustering algorithms exist, such as "soft" (or overlapping),

in which data items are partially assigned to multiple clusters as opposed to being placed within one ("hard"). The Fuzzy *c*-means algorithm is an example of a "soft" clustering technique.

This dissertation will further evaluate the SOM, the K-means algorithm, and affinity propagation within different contexts. In terms of their relation to other types of clustering, the affinity propagation algorithm demonstrates some hierarchical qualities and is often adapted to fall within the hierarchical clustering category, whereas both the SOM and K-means are considered to be partitional algorithms.

2.3 Semi-Supervised Approaches

Traditionally, unsupervised clustering methods proceed with no outcome measure and no previous knowledge about relationships within the dataset [11]. Jain points out that the data-driven nature of clustering makes it very difficult to design clustering algorithms and that incorporating side information regarding the feature vectors can be extremely useful in finding good partitions [34]. Thus, when information is available about the clusters, a combined approach can be utilized. Using this additional side information represents a semi-supervised type of clustering technique. An example of this type of side information are cluster labels of some of the data known ahead of the clustering process; Chapter 6 will provide further exploration of this technique. In some cases, according to Bair, "one may wish to identify clusters associated with a particular outcome variable" [11]. A semi-supervised approach associated with specific outcome variables will be explored in Chapters 4 and 6. For all of the techniques that will be

discussed in this dissertation, the clustering itself proceeds in an unsupervised fashion, but the subsequent knowledge discovery is enhanced by the addition of labeling.

Chapter 3

Domain Applications

Clustering techniques have been applied to problems in several different domains. Clustering has been used for vastly diverse purposes, from helping to identify and eliminate fraud within health insurance claims [58] to aiding in a graduate student's selection of which business school to attend [37]. This dissertation will focus on three domains where clustering techniques can be particularly beneficial: organizational decision-making, bioinformatics, and finance. In additional, the use of labeling, with both pre- and post-process methods, is employed to enhance the knowledge discovery step of the clustering process.

3.1 Organizational Decision-Making

Organizations are complex entities with many operational aspects that can be optimized through better decision making. Unsupervised machine learning can assist in the analysis of several areas of an organization to help managers minimize human bias and make more strategic organizational choices. Studies in this area have used unsupervised learning to focus on several aspects of organizational analysis, including

customer segmentation [71], business models [38], and product positioning [25]. Machine learning can even be used to classify organizational characteristics that are not easily quantified. For example, Cobo *et al.* used fuzzy *c*-means clustering to analyze metaphors used in the description of an organization (e.g., organism, political system, transformation, etc.) in order to extract knowledge about different aspects of an organization [16].

One important aspect of organizational management involves attempts to better understand customer behavior. For organizations engaged in e-commerce, customer behavior in web interfaces can be critical. In this field, the clustering ability of SOMs has been used to make web searching more relevant to users' needs. For example, SOMs have been used to facilitate greater usability of web directories through subject term clustering [82–83], to create more personalized web searches through user profile modeling [17], and to create multilingual web directories through webpage hierarchies [79].

These techniques are beneficial in that they can simultaneously evaluate a large number of organizational characteristics that, on the surface, do not appear to be immediately linked. For example, Rutherford *et al.* used an SOM to highlight differences within small businesses in order to identify subcategories of operations that would have greater utility for organizational planning [63]. In their analysis they selected features such as owner education, owner experience, firm age, type of pension plan, capital structure, and source of funding as the profile for the SOM. Their results found that firm size as measured by the number of employees is critical in predicting success because it is intricately connected to other features that can be isolated and evaluated [63].

In Chapter 4 of this dissertation, an SOM will be applied to characteristic data of academic libraries to assist in resource allocation and future planning, a project that is similar to Rutherford *et al.'s* analysis of small businesses [63]. Previous work using unsupervised learning in libraries has focused largely on collections of materials. Specifically, Linton *et al.* used an SOM to analyze the abstracts of articles in management journals [48] and An *et al.* used an SOM to analyze the subject terms of articles in library and information science journals to determine which journals had the most similar content for the purpose of aiding in literature selection [7]. However, no previous work has used unsupervised clustering for the purpose of facilitating resource management and strategic planning in libraries.

3.2 Bioinformatics

Bioinformatics, an interdisciplinary science that uses computational methods to investigate problems in molecular biology, is another area in which machine learning techniques play a prominent role. Rong summarizes the province of bioinformatics as "exploring underlying mechanisms of biological complexes, verifying biological hypotheses, and providing evidence through *in silico* simulation for further theoretical development" [62]. It has its origins in the 1960s when the first protein sequence databases were constructed [76]. Since then it has continued to grow in utility as DNA sequencing technology and the subsequent mapping of organisms' genomes have created such vast collections of data that it has moved the information available past the scope of a lab investigator.

The "raw material" of bioinformatics consists of both nucleic acids, which are the DNA and RNA molecules that contain the genetic instructions for living organisms, and proteins, which are the products of nucleic acids that consist of chains of amino acids [85]. The sequence data of DNA is comprised of nucleotides, whose bases are represented by a single letter (A, adenine; G, guanine; T, thymine; C, cytosine). Often, these sequences are millions of characters long and can represent several organismal genes. Protein data is the product of expressed genes within an organism. This data is also represented as a sequence of characters which are the amino acids present in the protein, with each of 20 amino acids being represented by a single or three-letter code, as shown in Table 3.1.

Amino Acid	**3 Letter Code**	**1 Letter Code**	**Amino Acid**	**3 Letter Code**	**1 Letter Code**
alanine	ala	A	isoleucine	ile	I
arginine	arg	R	leucine	leu	L
asparagine	asn	N	lysine	lys	K
aspartic acid	asp	D	methionine	met	M
asparagine	asx	B	phenylalanine	phe	F
cysteine	cys	C	proline	pro	P
glutamic acid	glu	E	serine	ser	S
glutamine	gln	Q	threonine	thr	T
glutamine	glx	Z	tryptophan	trp	W
glycine	gly	G	tyrosine	tyr	Y
histidine	his	H	valine	val	V

Table 3.1: Amino Acids with Three and Single Letter Codes

A variety of clustering techniques are commonly applied to analyze not only DNA and protein sequence data but other molecular biological datasets as well, such as imagery data. SOMs have been used for a number of different functions, such as

classifying DNA sequences [54], identifying similarities and differences among protein homologs [26], and comparing newly discovered protein sequences with known sequences [4]. The K-means algorithm has been employed in the analysis of protein sequence motifs [20, 84] and in the classification of ion mass spectrometry images [43]. Affinity propagation has been evaluated in the partitioning of protein interaction graphs [67].

Sequence classification is a recurring theme in clustering's application to bioinformatics. This involves identifying similarities in sequences for the purpose of determining sequence identity and biological function. Yang *et al.* describe how clustering techniques are essential for protein classification since experimental characterization cannot keep up with newly sequenced protein data: "One approach is to classify each family into distinct clusters consisted [sic] of functionally related proteins. When a new protein is assigned to a cluster, the biological function of this cluster can be attributed to this protein with high confidence" [78]. In order to accomplish this goal, Yang *et al.* employed the affinity propagation algorithm to cluster and classify protein sequences from three different datasets [78]. With a similar purpose, Naenna *et al.* used an SOM to classify DNA on the basis of a region of the genomic sequences—specifically, the splice junctions which are the boundaries between exons and introns, which are those sections of the DNA that produce a phenotype and those that do not [54].

One subfield of bioinformatics involves the study of metagenomic data, which is the collection of genomic sequences present in a given environmental sample. Chapter 5 will discuss the application of clustering techniques to metagenomic data in the manner of the works described above. However, unlike these studies, the work described in

Chapter 5 clusters a collection of sequences from a single environment and uses these as the basis for the analysis.

3.3 Financial Markets

Unsupervised clustering has been applied to many aspects of financial market analysis, including bankruptcy prediction [47], credit scoring [47], financial reporting [32], accounting [14], and stock performance analysis [39]. A central goal in this area is the prediction of future performance, which is critical for the successful management of any organization. In a study that used both supervised and unsupervised methods for financial forecasting, Powell *et al.* stated that "Stock forecasting is a major component of any finance institution because predictions of future prices, indices, volumes and many more values are often incorporated into the economic decision-making process" [60]. Traditional stock analysis generally proceeds in two directions: technical analysis, which uses past market data in combination with behavioral economics, and fundamental analysis, which includes financial information such as earnings, dividends, and cash flow as well as external economic factors [5]. Machine learning techniques can remove the need for domain expertise in these areas and efficiently facilitate the prediction of stock performance on a large scale.

Unsupervised clustering techniques have been used for various aspects of market prediction in a variety of settings. Pavlidis *et al.* used unsupervised clustering to perform financial forecasting in foreign exchange markets [57]. They used three clustering algorithms (Growing Neural Gas, DBSCAN, and *k*-windows) followed by a feed-forward neural network that was trained on each cluster to act as a local predictor of performance

(a supervised approach) [57]. Wang used the K-means algorithm to cluster stock data from the China Shanghai 180 exchange [68]. Wang focused on several features of the equities including profitability, capital expansion, asset management, growth, and solvency, finding a positive correlation with investment success [68]. Tsai *et al.* used an SOM with a decision tree to analyze trading preferences among various types of investors in the Taiwan stock market [66]. They focused on two factors which affect the portfolio choices of investors: stock characteristics (e.g., earnings per share, dividend yield, market-to-book) and investor features (e.g., gender, wealth levels) [66].

Like the works cited above, Chapter 6 of this dissertation has a similar focus on the prediction of the future performance of equities. In that chapter, the affinity propagation algorithm is used on a pre-selected subset of stocks traded on the NASDAQ in conjunction with historical high-performing stock data.

3.4 Applications in These Domains

The remainder of this dissertation focuses on unsupervised clustering's application to specific examples of the three domains described above. All algorithms were implemented leveraging the .NET framework. For the work in Chapters 4 and 5, SOM and K-means clustering tools were constructed in C#, and for the work in Chapter 6, an affinity propagation tool was constructed in the functional programming language F#. In each chapter different methods of analysis will be explored, and the usage of various data labeling strategies will be evaluated in the improvement of knowledge discovery of the completed clusters. The work in Chapters 4 and 6 applies a metric-based label to the raw data before the application of an unsupervised clustering algorithm. In

Chapter 4, all data is labeled with a metric, whereas in Chapter 6, a small subset of the data is labeled to track "good" data. These techniques fall within the bounds of semi-supervised machine learning, but represent an original melding of a distinct pre-process metric being applied before cluster analysis is performed. The technique described in Chapter 5 uses labeling in a post-clustering process to help an investigator perform a quick visual analysis of the genomic data present in the sample and help direct future analysis.

Chapter 4

Clustering with a Metric-Based Label for Library Management

*Note: The basis for this chapter was previously published as D. Ennis, A. Medaille, T. Lambert, R. Kelley, and F. C. Harris, “A comparison of academic libraries: an analysis using a self-organizing map,” *Perform. Meas. Metrics*, vol. 14, no. 2, pp. 118–131, 2013.

4. 1 Introduction

Data comparisons among libraries can provide valuable information for making choices regarding resource allocations and service provisions. In its "Standards for Libraries in Higher Education," the Association of College and Research Libraries recommends that academic libraries use external comparisons with their peers for benchmarking purposes in order to identify strengths and weaknesses and “to develop a more informed picture of institutional standing within the higher education marketplace” [8]. In “Determining Quality in Academic Libraries,” Pritchard writes that “The ability to make unambiguous and meaningful comparisons is an important issue in assessment” [61], and in a discussion of library criteria, Knightly includes comparisons with other organizations as one of seven types of measurement, noting that comparisons can reveal

both strengths and areas in need of improvement [40]. While library comparison data needs to be understood within the context of individual library environments, it can inform decision making when used in combination with other data.

Cluster analysis is one method that can be used to compare library data. Lorr defines clustering as "the grouping of entities into subsets on the basis of their similarity across of set of attributes," and cluster analysis can be especially useful for revealing patterns and relationships within complex datasets [49]. In the field of library and information science, cluster analysis has been used to study term indexing, web searching, journal citations, and user behavior. Cluster analysis has been less frequently used to study libraries as a whole; however, in a study of the multiple dimensions that comprise academic library effectiveness, McDonald and Micikas used cluster analysis to distinguish among five different library groups. Of these groups, the authors identified a cluster of "highly effective" libraries that were located at institutions that had good financial support, limited enrollments, specialized curricula, and a large ratio of books per student [51].

One clustering technique that has not yet been used to evaluate academic library data is an SOM. For this study, an SOM was used to identify data points that could be correlated with high resource and service usage in academic libraries. To choose the metrics for analysis, a number of different studies were consulted that describe the value of measures such as circulation, attendance, weekly public service hours, building and e-resource usage, reference transactions, and attendance at instruction sessions, among others [18, 31, 56, 59, 70, 73]. When correlated with other data, these metrics have provided valuable insights. For example, in analyzing the connection between traditional

and newer measures of academic libraries, Weiner found that a significant relationship existed among service metrics (numbers of reference transactions and instructional presentations, and attendance at instructional presentations) and more traditional library measures such as budget, staff, and clientele [70]. Whitmire found that at certain types of institutions a positive relationship existed between library resources and students' gains in critical thinking, but a negative relationship existed between library services and undergraduates' library use [73]. Emmons and Wilkinson looked for correlations among academic library measures of staff, collection, circulation, and services (number of reference questions and percent of students receiving instruction), and institutional measures of retention and graduation [21]. They found that a significant relationship existed between library staff and both retention and graduation rates. Because of the prevalence of service and usage factors in recent library correlation studies, several of these oft-cited metrics were selected for analysis using an SOM.

The current study sought to answer the following question: Can an SOM cluster analysis of complex academic library data be used to reveal meaningful relationships among resource and service measures and other library factors—relationships that might otherwise be overlooked? To answer this question, this chapter looked at three commonly reported measures of resource and service usage (circulation, attendance at instruction sessions, and reference transactions) and used an SOM cluster analysis to determine whether correlations could be found with features related to library expenditures, personnel, materials, and service offerings—data that is consistently tracked by most academic libraries. First, an SOM was used to cluster library data. Then the output was analyzed for the purpose of: (1) seeing which libraries clustered together

on the basis of their combined features, (2) determining which clusters could be identified as “high-performing,” and (3) identifying the distinguishing characteristics of the high-performing library clusters. It is important to note that while an SOM can be used to cluster data and facilitate the discovery of correlations among the data, it cannot provide an explanation as to why those correlations exist. However, an SOM mapping and cluster analysis can provide a useful starting place for more detailed evaluations.

4. 2 Methodology

4.2.1 Data Used

Library data were collected from U.S. and Canadian academic libraries that are members of the ACRL by using the ACRL Metrics data portal [8]. Fifteen library features for the fiscal year 2010 were selected for analysis in an SOM and are listed with their definitions in Table 4.1. These features consist of a combination of resources such as collections, staffing, and expenditures, and activities such as giving presentations to groups and staffing service desks. These particular features were selected because they were consistently reported among the majority of libraries in the data portal and they covered a broad range of library features.

Feature	Definition
Volumes in Library	Total number of physical units that have been cataloged, classified, and made ready for use
Total Serials	Total number of unique serial titles
Monographs	Annual monograph expenditures
Current Serials	Annual serial expenditures
Other Library Materials	Annual expenditures on items other than monographs and serials such as backfiles of serials, charts and maps, audiovisual materials, and manuscripts
Miscellaneous Expenditures	Annual expenditures on items other than library materials such as expenditures for bibliographic utilities, literature searching, and security devices
Salaries and Wages of Professional Staff	All salaries and wages of professional staff, excluding fringe benefits
Salaries and Wages of Support Staff	All salaries and wages of support staff, excluding fringe benefits
Salaries and Wages of Student Assistants	All student wages, regardless of budgetary sources of funds
Professional Staff	Number of FTE staff that the library considers professional, such as librarians, computer experts, systems analysts, and/or budget officers
Support Staff	Number of FTE staff that are not included in the count of professional staff, excluding maintenance and custodial staff
Student Assistants	Number of FTE student assistants employed by the library
Staffed Service Points	Number of staffed public service points in main and branch libraries
Weekly Public Service Hours	Total hours that the library is open per typical full-service week
Presentations to Groups	Total number of presentations made as part of bibliographic instruction programs and through other planned class presentations, orientation sessions, and tours

Table 4.1: Library Features Used in the SOM

4.2.2 SOM Process

An SOM was constructed according to the specifications described in the previous section. Prior to running the data through the SOM, the data were examined for

outliers that suggested mistakes or inaccuracies. These included values well outside expected category ranges—for example, smaller institutions with reported reference transactions that were orders of magnitude larger than major research universities. These outliers were removed, leaving data from a total of 1,395 libraries. The data were then normalized according to the number of full-time equivalent (FTE) students enrolled at the institution. Finally, the data were run through the SOM and fed to a 44 x 44 neuron map, a size which was selected because it was large enough to accommodate the data and resulted in a graphic that facilitated the analysis.

The resulting SOM visualization was then analyzed for common features among the clusters. The analysis was conducted by computing a library performance metric (LPM) that was based on three features that represent usage of the library: total number of reference transactions, total participants in group presentations, and total circulation transactions (Table 4.2). The three features that comprise this metric were selected because they have consistently been employed to measure library usage in a number of studies [21, 70, 73]. While other usage metrics—such as the number of database logins, website visits, or full-text article downloads—would have provided both interesting and valuable information as well, these data were either tracked too inconsistently or were not provided by a significant number of libraries. The three LPM values were combined so that they had equal weight in the final score and were normalized according to the number of students (FTE) at each institution.

Metric	Definition
Reference Transactions	Total number of reference transactions, both in person and through virtual means
Participants in Group Presentations	Total number of participants in presentations made as part of bibliographic instruction programs and through other planned class presentations, orientation sessions, and tours
Circulation Transactions	Total number of items lent, including renewals

Table 4.2: Items Used to Compute the LPM

4.3 Results and Analysis

The library SOM is displayed in Figure 4.1. Data were clustered according to the features listed in Table 4.1, and each library is represented numerically by its LPM. Figure 4.1 shows how the libraries clustered according to the fifteen features, with libraries with similar characteristics grouping together at different locations on the map. Once the libraries had clustered, the high-LPM libraries were identified through the application of the LPM label, and the features of the high-LPM libraries were then analyzed to determine what led to their locations on the map. Thus, the LPM was used only as a label on the visualized map and had no bearing on the actual cluster position within the map.

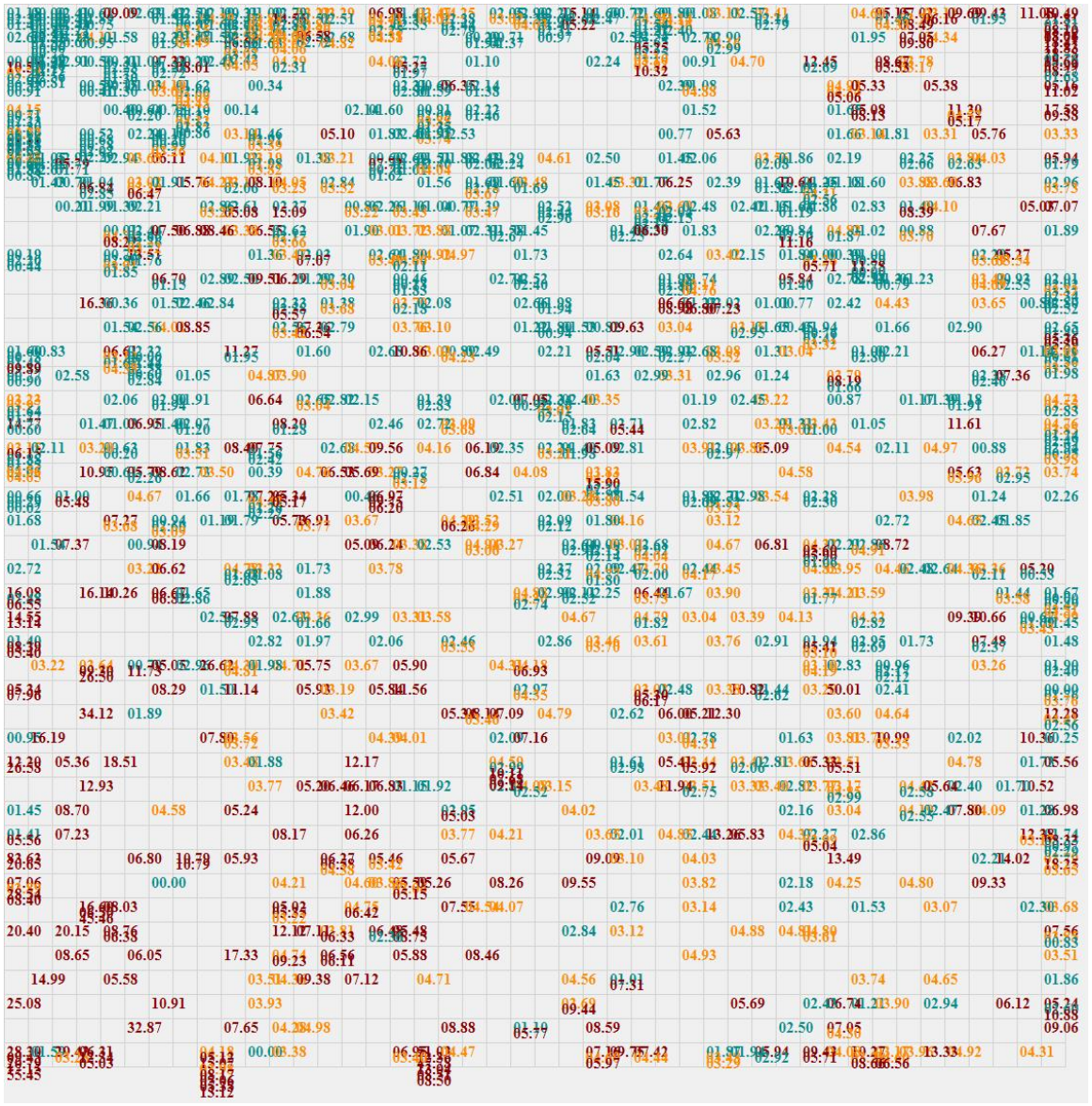


Figure 4.1: SOM Output with Libraries Represented by Their Colored LPM Labels.
Colors indicate the following: red = high, orange = middle, and green = low.

Analysis of the SOM revealed a number of different characteristics. One characteristic of the map is that many resource variables (i.e., greater expenditures, greater numbers of materials, and higher numbers of staff; calculated as a ratio to student FTE) increase when descending on the map. Thus, those libraries with greater material

and staff expenditures and greater numbers of staff tended to collect along the bottom of the map, while those libraries with lower resources tended to collect along the top part of the map. Moving horizontally on the map revealed variations in terms of other features, including number of presentations to groups and number of public service hours. Those libraries that offered greater numbers of presentations and public service hours per student FTE generally collected on the right side of the map, while those with lower numbers collected on the left.

In Figure 4.1 the LPM is represented on the SOM by color. Those libraries with a high LPM are colored in red, a medium LPM in orange, and a low LPM in green. Figure 4.2 shows the same SOM with the neurons colored according to the average LPM of the libraries assigned to that neuron, with the lowest-scoring libraries in dark green and the highest-scoring libraries in dark red. In showing the LPM average of all libraries that were placed at a particular neuron on the map, Figure 4.2 provides a clearer visualization of the different clusters. In looking at the color locations on the map, it appears that most of the low LPM libraries (green) appear in the upper portion of the map, while a greater number of the middle and high LPM libraries (orange and red, respectively) appear in the lower portion.

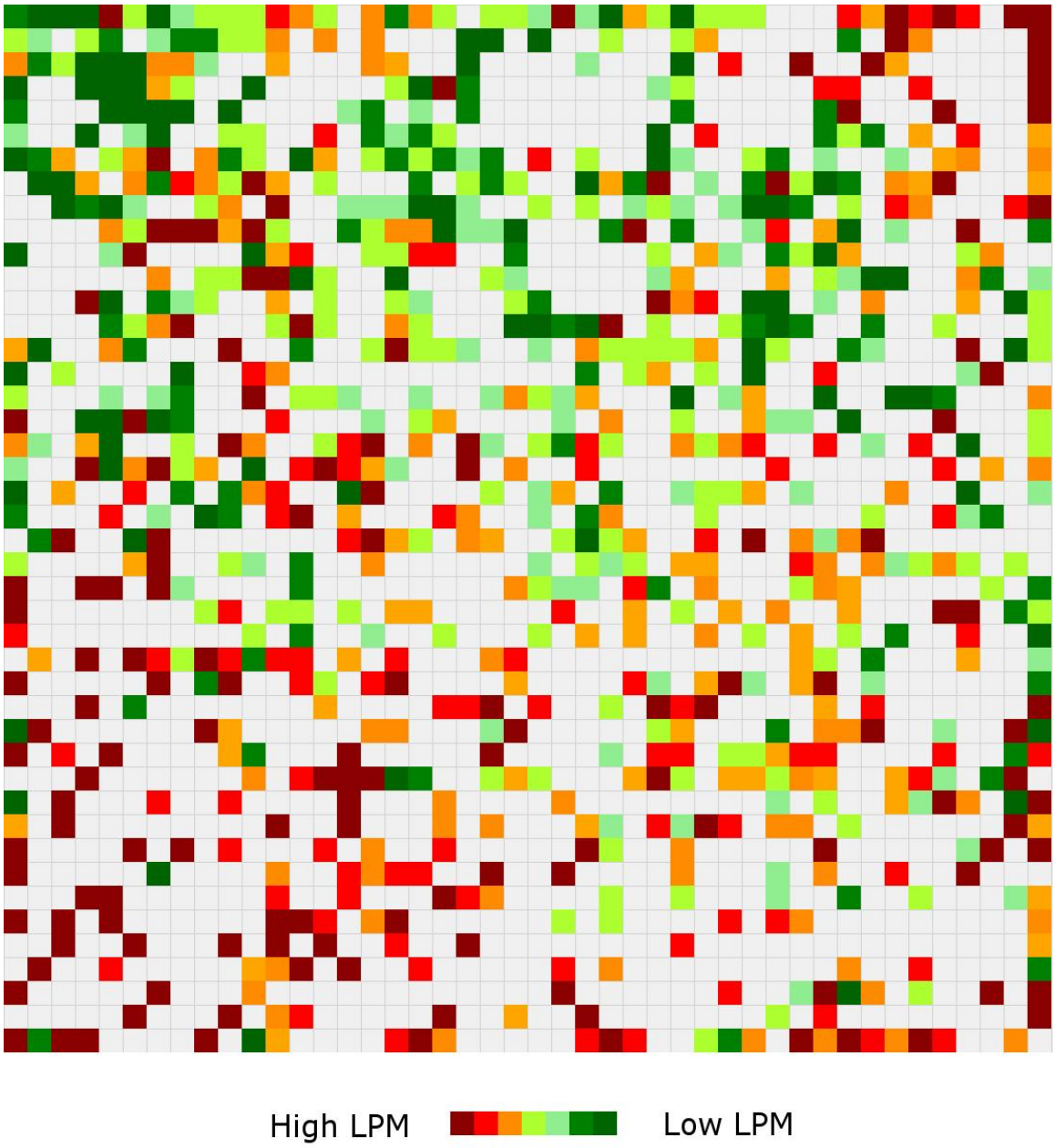


Figure 4.2: SOM Output with the Color Dispersion as Determined by the Average LPM per Neuron

Although the diverse features of the libraries resulted in a variety of placements on the map, three areas emerged of particular interest, and these areas are outlined and labeled in Figure 4.3. First, an area of low-performing libraries appears in the upper left corner of the map. An analysis of their common features reveals that these libraries are

generally low in resources such as expenditures and staff; thus, this area has been labeled as Lower Resource, Low Performing in Figure 4.3.

Two high-performing library areas (outlined in Figure 4.3) emerged in the lower-left and upper-right portions of the map. An analysis of the common features of libraries in the lower-left area reveals that these high performers have greater numbers of resources (i.e., larger budgets, more materials, and higher numbers of staff), and this area has been labeled as Higher Resource, High Performing in Figure 4.3. This area includes libraries at research universities such as the University of North Carolina, the University of Southern California, and Johns Hopkins University, among others. Thus, for this group of libraries, greater numbers of resources per student FTE can be correlated with better library performance, as measured by the LPM.

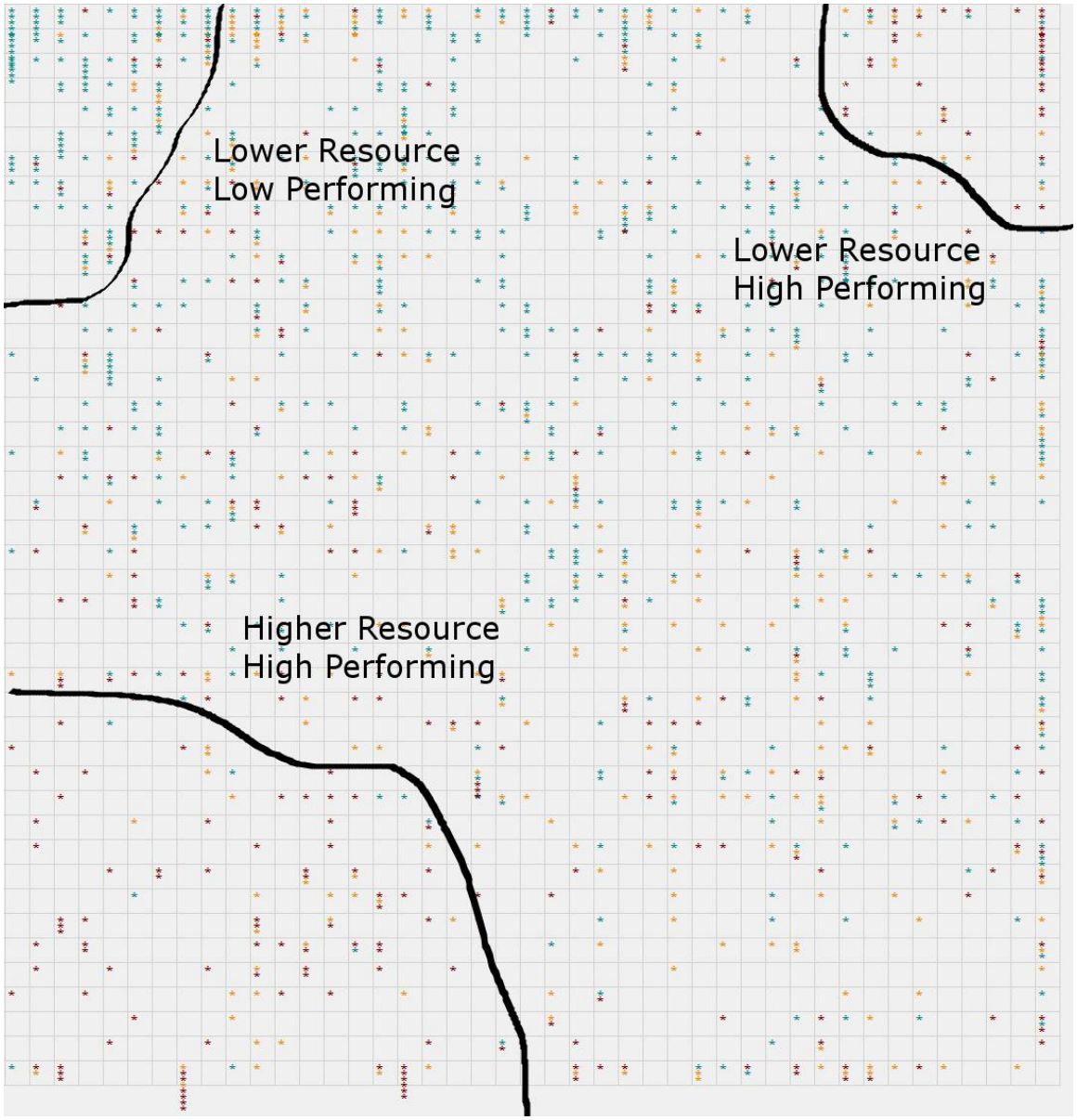


Figure 4.3: SOM Output Showing the Three Library Areas

The third area of interest in the upper-right area of the map also contains several high-performing libraries (outlined in Figure 4.3). Unlike the high-resource area in the lower-left, an analysis of the common features of this high-performance area in the upper-right reveals that these libraries are generally low in resources (i.e., smaller budgets, fewer materials, and lower numbers of staff). This area includes libraries at

institutions such as Florida Keys Community College and Tougaloo College and has been labeled as Lower Resource, High Performing in Figure 4.3. An analysis of the features of the libraries in this area reveals that these libraries give greater numbers of presentations to groups, offer a greater number of public service hours, and have greater numbers of staffed service points per student FTE. Thus, it would appear that their high LPM scores can be correlated to these higher service features rather than to greater numbers of resources.

It is significant, although perhaps unsurprising, that libraries with greater numbers of resources (in the Higher Resource, High Performing area) achieve higher levels of performance, as measured by the LPM. However, the implications of those libraries in the Lower Resource, High Performing area are also significant. In offering greater numbers of these services, the Lower Resource, High Performing libraries may be achieving higher levels of performance by a means other than through greater numbers of resources. It may be, for example, that by offering greater numbers of group presentations, these libraries are educating their users about library resources, which in turn leads to greater numbers of circulation transactions, or they are educating their users about the assistance available in the library, which in turn leads to greater numbers of reference transactions. In addition, by being open for longer hours, libraries are making their collections more readily available for checkout.

4.4 Conclusion and Suggestions for Further Study

An SOM can provide considerable value in its ability to cluster large library datasets and to facilitate comparisons among libraries. In this study, an SOM has been used to cluster library features such as resources, staff, and expenditures, but it could easily be used to facilitate the analysis of other types of library data as well. While a cluster analysis technique such as an SOM cannot provide a holistic picture of the state of a single library, it can be used to help one to quickly ascertain data similarities among several features at large numbers of libraries. In this chapter, the SOM used in combination with the LPM labeling has elucidated relationships among certain library features with regard to performance, but it cannot explain why those correlations exist. This is just the starting point for further evaluation. Future research could delve into the demographic characteristics and other features of the high-performing libraries in order to arrive at a more detailed picture of how these libraries achieve high usage rates.

Certainly, the metrics chosen for the LPM used here offer only a partial picture of how libraries are being used. For example, total circulation transactions represent only some use of library collections and do not represent factors such as in-library usage of material, database usage, or article downloads. In addition, the fifteen library features used in this SOM were selected for visualization because of their recurrence as typical points of library comparison, but other library features would provide for a fruitful comparison as well. In addition, a different library performance metric could be constructed from different outputs, such as building usage or e-resource usage.

For this analysis, data were normalized on the basis of student FTE, but this choice does not account for all of the factors that could influence the results, such as institutional type, budget, sources of funding, consortial agreements, etc. In addition, it was felt that the greatest insights would come from using the largest set of library data possible in order to find relationships that were not easily discernible by obvious or pre-existing classifications. However, future work might involve the identification of library groups that cluster within certain parameters, such as size (e.g., small, medium, and large libraries), budget, funding support, institutional mission, etc. In fact, this approach to cluster analysis could aid in the identification of peer groups. Academic libraries typically turn to their institutional administrative offices for an official list of peer institutions [9], but libraries on these official lists can differ greatly. Pritchard writes that "It is possible that the peer institutions used by the administration for strategic planning will not each have a library that functions comparably" [61]. An SOM can be used to cluster similar libraries and identify library peer groups; libraries could then look to the high performers within the same cluster to see what kinds of choices these libraries are making.

Although the SOM analysis provided here cannot answer all of the questions prompted by the clustering of the high performers, the analysis does have implications for libraries seeking to improve their performance. In times of decreasing budgets, libraries that want to improve their outputs may be unable to achieve this result through increasing their resource expenditures or hiring more staff. This cluster analysis suggests that libraries may be able to improve their performance by putting a greater emphasis on the services they offer to their users, whether it be through providing greater numbers of

staffed service points and public service hours or providing greater numbers of instruction sessions and outreach opportunities. Future research could analyze the common features among these libraries that are not apparent in the data points examined in this dissertation.

Chapter 5

Facilitating Metagenomic Analysis Using Clustering

5.1 Introduction

Metagenomics, a relatively new field within biology, involves the classification of DNA sequence fragments from environmental samples consisting of diverse microbial populations. Research in metagenomics focuses on both the classification of organism types and the functions of sequences that are present in environmental samples. The latter method essentially treats the metagenome as a single genomic item which can provide information about metabolic processes within the habitat [27]. Because metagenomics research involves vast amounts of data, its analysis calls for the use of tools that can facilitate the process with efficiency. This type of large-scale analysis of genomic data falls within the interdisciplinary field of bioinformatics, which is a merging of molecular biology, probability, and computer science. Clustering techniques within the field of machine learning provide excellent methods for analyzing metagenomic data.

A number of approaches have been used to analyze DNA sequences in metagenomic data, including sequence similarity searches, sequence composition methods, and phylogenetic methods [12]. A Basic Local Alignment Search Tool

(BLAST) search provides one type of similarity comparison, wherein a query sequence is compared against a library of known sequences in order to find similar sequences that fall within a designated threshold limit [6]. An example of a sequence composition method is a Markov model, which is a probabilistic-based model of the composition of the likely DNA sequence. A Markov chain model assigns the current nucleotide base a probability based on the composition of the preceding nucleotides [36]. Phylogenetic techniques use an evolutionary relationship model to locate where the query sequence most likely fits. Applications may even employ a combination of these techniques to perform sequence classification.

Several studies have investigated methods for better facilitating the analysis of metagenomic data. Ghosh *et al.* developed a method for rapidly determining sequences in a metagenomics habitat. Their technique, called HabiSign, focuses on finding habitat-specific oligonucleotide usage patterns to differentiate between metagenomes which have variations at phenotypic, species, and biome levels [24]. Abubucker *et al.* developed a method which uses short DNA reads to analyze both the presence and abundance of microbial communities within the host environment [1]. Their system, HMP Unified Metabolic Analysis Network (HUMAnN), profiles the metabolic potential of microbial communities by using a series of steps that involve gene and pathway-level quantification, noise reduction, and smoothing of data [1].

Clustering methods are also used to facilitate metagenomic analysis [12, 45]. These methods include CD-HIT, which orders genomic sequences by length and then assigns the longest ones as cluster seeds. It then compares the reads against the existing clusters by using a greedy incremental algorithm, a type of algorithm that makes the local

optimal choice at each decision point [46]. CD-HIT is less computationally intensive than several other methods that compare every sequence against all the others present in the sample [46]. Uclust, another greedy algorithm-based method, differs from CD-HIT in that it uses heuristics to facilitate faster sequence comparisons [19].

The field of machine learning offers clustering techniques that have not yet been widely applied in metagenomics research. Machine learning clustering techniques use computer algorithms that adapt to the data presented, and they can be categorized into two varieties: supervised and unsupervised. A supervised algorithm relies upon previously evaluated data to create a classification system, whereas an unsupervised algorithm can be used to group unknown data into collections of similar items, also known as binning [50]. Thus, machine learning techniques can effectively cluster extremely large quantities of data with some or no training, a quality which makes them well-suited to metagenomics analysis.

Two types of unsupervised algorithms that offer great potential for metagenomic analysis are the SOM and the K-means algorithms. Both techniques have been successfully applied in other fields but have had only limited application in metagenomics. Of note, however, is a study by Weber *et al.* which employed a machine learning algorithm in the analysis of five metagenomes of medium complexity. The authors created an application, called TaxSOM, that uses a self-organizing map to plot unknown genomic signatures against known types in order to provide taxonomical classification [69].

The work described in this chapter applies the SOM and K-means algorithms to the analysis of metagenomic data collected from a sample from the Great Boiling Springs

Habitat in Nevada. The typical approach to metagenomic analysis centers on taxonomical classification, generally proceeding in one of two directions: either by comparing the complete sequences or by focusing on the 16sRNA genes present in the sample. The 16sRNA genes comprise the DNA sequence that specifies the small ribosomal subunit and is often used for species identification [24]. The approach taken here uses a variation of the former and attempts to identify possible candidates to serve as reference genomes for use in further analysis. A reference genome is a documented collection of nucleotide sequences and protein products from a specific species, often documented from multiple samples. These sequences, which are often annotated by domain experts, can serve as points of comparison to unknown or novel sequences.

This chapter attempts to answer the question: Can the use of machine learning clustering techniques provide a guideline as to which reference genomes should be included for further analysis in determining possible organisms that are present in a metagenomic sample? This chapter does not attempt to directly taxonomically classify organisms present in the sample, but rather "pre-bins" the data so as to provide a method to streamline future analysis. The unsupervised techniques discussed here can serve as a launching point for elucidating protein sequences that could serve as possible reference comparisons to a specific metagenomic sample and lead to further study, possibly by using supervised learning or even domain-expert analysis.

5.2 Methodology

This study was conducted in three phases. First, an algorithm was used to cluster metagenomic reads. The clusters are composed of similar sequences, most likely from related genes. These clusters, in turn, can be used to ascertain what functionalities are present within the given sample habitat by using the consensus sequences within the clusters to perform a BLASTX similarity search (BLASTX compares a nucleotide query sequence translated in all reading frames against a protein sequence database [13]. This is similar to the goal of the HUMAnN system [1]; however, the pipeline described here does not focus on functionality. In the second phase the estimated midpoints of the clusters were used as the basis for BLASTN searches, which allowed for the quick identification of likely genomes present (BLASTN compares a nucleotide query sequence against a nucleotide sequence database) [13]. In the third phase, a visual representation of the clustered data + BLAST results is produced, which demonstrates the frequency of each projected organism. These phases together could also serve as the start of a pipeline to quickly identify good choices to use as reference genomes for further analysis. Figure 5.1 provides a flow chart representing the process.

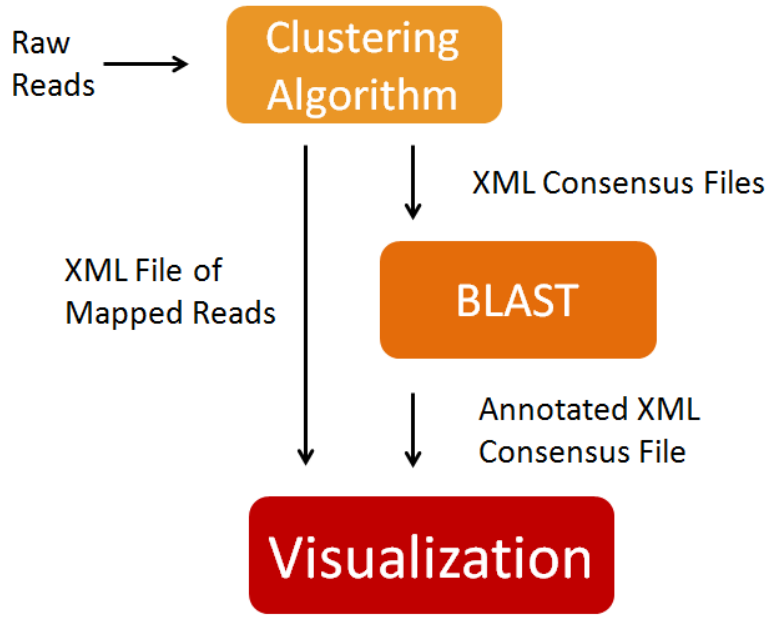


Figure 5.1: Metagenomic Data Processing Pipeline

5.2.1 Data Used

Any collection of mostly non-ambiguous FASTA sequences (ATCG only; N is supported as well) could be used in the first phase of this process. FASTA is a text format for both nucleotide and protein data which includes a descriptive line and the single-letter code representation of either nucleotide or protein sequences. The data used in this study consisted of the Great Basin Boiling Springs, Nevada habitat sample, collected on December 1, 2008. The sample data were obtained from the Integrated Microbial Genomes with Microbiome Samples website [33]. The sample is an aquatic thermal springs ecosystem with temperature ranges as high as 90°C. The sequencing technology used on the project was Roche 454 Titanium. The estimated size of the data is 6,283,876 kilobases.

5.2.2 Clustering Pipeline

C# software applications were developed which are capable of selecting a FASTA formatted reads file and clustering the reads either within an SOM or via the K-means algorithm. The SOM program shows a small grid representation of the clustering (which serves as the first visualization of the clustered sequences) and outputs an XML file listing each read and its assigned location on the map. After the initial clustering run, the results are carried forward in a simple XML format, which is illustrated in the Appendix. A second XML file is generated which displays the neuron weight sequences, as well as the consensus sequence of the neurons with the highest number of mapped sequences. The XML output files were used for the last two parts of this project. A screenshot of the application in action is shown in Figure 5.2.

A selectable number of encoded partial sequences were used as input vectors to both the map and K-means clusters. The neuron with the closest weight vector to the input is assigned that particular input vector and has its weights updated to more closely match the assigned input vector. Other nodes within the winning neuron's neighborhood have their weights updated as well to a lesser amount. This continues to happen in an iterative process until the weight updates fall below an assigned threshold, at which point the map is essentially complete with similar sequences being assigned in close proximity to each other.

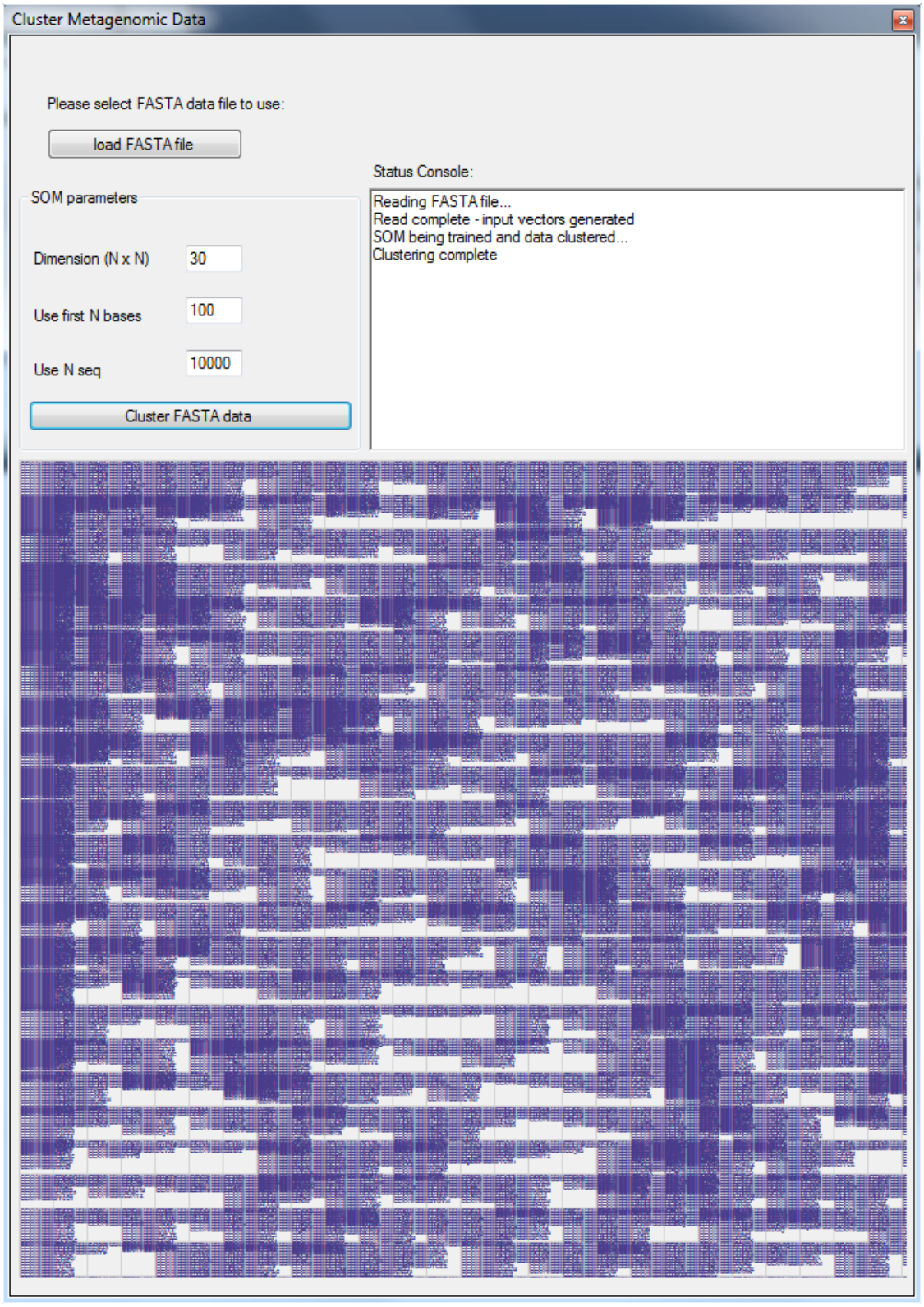


Figure 5.2: 100-Base Read SOM with a 30 x 30 Grid of Neurons

It was necessary to employ an encoding scheme that avoided prejudicing the clustering algorithms. A naive encoding scheme could result in the middle two nucleotide values appearing closer to each other. This issue was avoided by encoding each base as four values (e.g., A = 1,0,0,0), which essentially represents each nucleotide at an equal distance from each other in three-dimensional space. However, this did increase the encoded input vector by a factor of four from the single-character representation to the four-character vector. Using this encoding scheme, the clustering results were promising based on the close similarity of the decoded neuron weight sequences versus the consensus sequences built from all of the sequences assigned to that particular neuron. See Table 5.1 for a comparison of the calculated consensus sequence against the cluster center weight vector.

Weight Sequence	**Consensus Sequence**
AAAGAGAAAAGATTAAAAAGAAAA TAAAAAAAAAAGAAATGAATTAAG AGAAAATGAGAATGGTGAAGAATA AGAATATTAAAAGATATAAAAAAA AATA	AAAGAGATAAGAGAAATAAATAAA GCTAAAAAGAAGGAAATGGTTATAA GAAAATAAGAATGAAGAAGAGAAA GTTTATAAAAAGCTATAGAAAAAAA TA

The table demonstrates a comparison of a sample neuron weight sequence vs. the consensus sequence generated from the collection of reads assigned to that neuron. The sequences' close similarity (based on a character-by-character analysis) serves to validate the process.

Table 5.1: Comparison of a Sample Neuron Weight Sequence Vs. the Consensus Sequence

The next step involved processing the XML results file that was generated from the clustering algorithms to find likely represented genomes (partial). The consensus XML file with the neuron weight sequences was used as the basis for this phase. Manual BLASTN queries were performed using the microbial genome database at the National

Center for Biotechnology Information [13]. The standard settings were used with the exception of reducing the word size down to seven based on the short length of the sequences. Promising results from the top several “center” sequences were manually added to the consensus XML file, while weight sequences that produced no acceptable matches were removed, as shown in the Appendix.

The updated XML file, including the labeled BLASTN results, was fed to a Python application, using the Matplotlib library. The results represented the frequency of the likely genomes, as well as the overall clustered map of sequences. A visualization resembling a heat plot was used to show the frequency of sequences at each position on the map, with the labeled sequences provided by the BLAST results serving as guide locations.

The Python script expects two XML files matching the project's proprietary formats as input. The first portion of the script reads in and parses the two XML files, generating a count of assigned sequences at each neuron location (x,y). In addition, it also parses the updating consensus XML file for the names of the likely reference sequences which will serve as labels within the frequency map.

5.3 Results and Analysis

The clustering phase of the pipeline was executed several times with various sequence lengths, numbers of reads, and map sizes for the SOM, as well as several different cluster counts for the K-means algorithm. The BLASTN comparisons were made using varying sequence lengths, but in the end the visualization phase was initiated

with only the 100-base results, which can be seen in Table 5.2. The table shows selected neuron weight vectors which were used as the BLAST query sequence generated from the trained SOM, the most similar result as returned by BLAST, the associated E-value (the expected chance of randomly seeing a match to the query sequence), and the percent of the query sequence that matched the top result. Table 5.2 shows the possible organisms present in the metagenomic sample which could be used as reference sequences and warrant further investigation. In particular, the *Thermoanaerobacter siderophilus* result is especially promising, in that it is a thermophilic, anaerobic, spore-forming bacterium that has been found in hydrothermal vents. This genome is a likely candidate for this environment and therefore can be used as a reference genome.

Neuron Weight Sequence	Top BLAST Result	E-Value	% Identity
TTTTTGAGAAAAAGAAAAAG AAGATAAAAAAAAAAAAAA AGAATGAAGATAGAAGAAG AATTGAGAGGAGTAAGGGAA AGGGTTAAGAAAGTTAATAA AA	NC_018664.1 Clostridium acidurici 9a chromosome, complete genome	9e-07	88%
TTTCTTTTTTGAACTTTACCTT TTGCTACTTTGGAACTTTTTA AAACACCTAGAGCATTAACA ACAAACCCTTCTACTGTTTCT TTTAAATCTTCTACCT	BAFA01000036.1 Staphylococcus aureus PM1 DNA, contig: PM1contig00036	0.019	89%
TTTGGAAGAGATTTGAAAGA AGAAAGATAGAAAAAACTTA GGAACTAAAGAAATAAGAGC GCATAAGAGAAACAAGAAA AAAAAAAGAGATTTTTTGGA A	AKXG02000035.1 Leptospira interrogans serovar Grippotyphosa	0.068	89%
TTTTGAGGAAAATAGAAAGG TAAAAGGAAAAATGTAGTAA TTAGTTAAAGAAAAAAGAAA GGGTGTGAGAAAAAAAATAA AAAAAATAGAAAATATGGAA	ADEJ01000237.1 Clostridium difficile 6534 contig_237, whole genome shotgun sequence	5e-04	83%
GTAAAGAATGTTTTTAGATAT ACGAAGAAAAATTAGTAAAA AAAAAAAAAAAAAAAAAAA ATTATAAAAAAAAAAAAAAA AAAGAAATTAAAAAAGCAAA	AJUD01000013.1 Thermoanaerobacter siderophilus SR4	6e-15	88%

Table 5.2: 100-Base Sequences BLASTN Results

A demonstration of the efficacy of either of these clustering techniques can be seen in the comparison of the consensus sequence from the largest cluster of the K-means algorithm with the results of the SOM. The largest cluster of sequences in the 100-base K-means run produced the same top result as the consensus weight sequence from the SOM application. Table 5.3 shows that the top BLAST result for the K-means algorithm

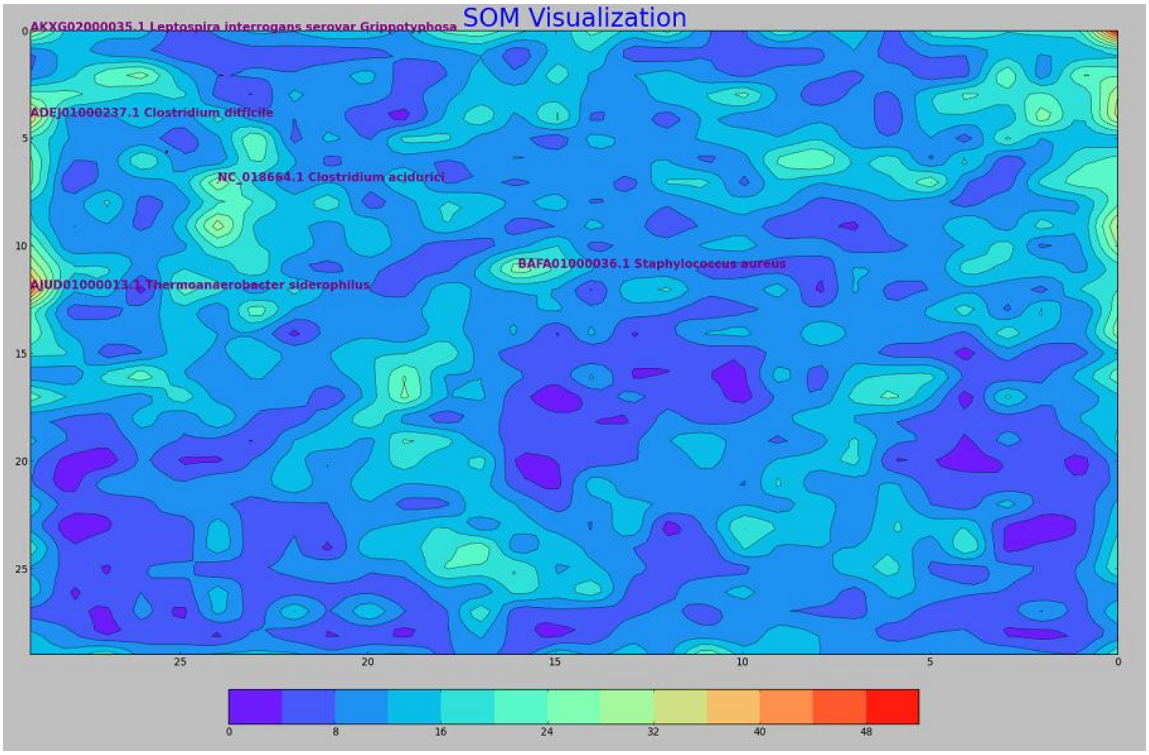
was also *Thermoanaerobacter siderophilus* SR4. However, the identity percentage and E-value are less significant than the SOM results.

Neuron Weight Sequence	Top BLAST Result	E-Value	% Identity
TTATAAAAGATATGAAGATA GGTTAGGAGATTTTTGAGAA GGCTAAAACAGTAGAATTTT CTTAAGGATTATCAGAAAAA TAAAAAAAAAAGGAATTTAA	AJUD01000013.1 Thermoanaerobacter siderophilus SR4	0.004	78%

Table 5.3: Top K-Means 100-Base Sequence Result

Figure 5.3 shows the results of the final phase of the processing pipeline and provides an illustration of the clustering of the sample sequences. Color positions on the map represent the abundance of sequences similar to that particular cluster center. Several cluster centers are labeled with the corresponding top BLAST results to give an investigator a visual overview of the abundance and possible types of sequences present in the sample. This representation demonstrates the applicability of the system and shows the benefit of refining and proceeding with this clustering technique.

The previously discussed CD-HIT application [19] was employed to attempt to validate the techniques used here. The metagenomic data used in the development of the described pipeline was subjected to the CD-HIT-454 application for clustering. These results were fed to the consensus building program, CD-HIT-CONSENSUS, which creates a set of consensus sequences from the clusters provided by CD-HIT-454. However, this provided a limited point of comparison as almost no reduction in data was found when evaluating mapped reads against reference genomes.



The key shows that color positions on the map represent the number of sequences assigned to that location. Clusters are shown in the light blue, yellow, and red areas.

Figure 5.3: SOM Data Visualization with BLASTN Results Labeled

The pipeline described here could be adapted to work in a fashion similar to the HUMAnN system described in Abubucker *et al.*, which seeks to determine the functionalities present in the metagenomic sample [1]. This direction would involve finding similarities in the consensus (or weight) sequences within the map versus known pathways to elucidate possible functionalities.

The complete pipeline was executed using only the first 100 (a configurable parameter) bases of each read as the input vectors to the SOM and K-means algorithms. A future project could involve the construction of the map that uses the entire read sequences as input.

5.4 Conclusion

This work attempts to solve a problem similar to that addressed by the TaxSOM application, a tool that trains an SOM on the basis of known genomic signatures and then maps metagenomic reads to the pre-calculated map [69]. The pipeline presented in this chapter uses the metagenomic reads themselves to train the clustering technique as a first step. The metagenomic data are then clustered using the trained algorithm—either the SOM or the K-means algorithm. This technique is both applicable and advantageous because it involves no initial preparation, attempting to reduce the data from the raw reads without reference to an outside source for comparison. The clustered sequences can then be used to reduce the amount of data to carry forward for downstream analysis.

The techniques employed in this chapter form the starting point in a metagenomic analysis pipeline. The clustering algorithms used in conjunction with the visualization tool allow a researcher to rapidly move to a more directed investigation of a metagenomic sample. The overarching goal of this pipeline is to provide a quick visual guide to an investigator as to what may be present in a habitat's sample and to identify which reference genomes may be selected for further analysis. Figure 5.3 illustrates how an investigator can use the visualization to see the amount of sequences that share similarities with the cluster labels. The clustering performed here was limited to a small subset of a much larger metagenomic data file in order to validate the functionality of the method. Future work could use the pipeline to cluster larger metagenomic datasets. In addition, in this study the connectivity between the clustering, BLAST, and visualization was user driven, but in the future these steps could all be rolled into a single tool. Finally,

the system described here could be enhanced to go in several different directions, including the classification of taxonomies present in a sample, the clustering of metagenomic reads for a reduction in computational requirements, or even as an evaluator of functional pathways that are likely to exist in a sample.

Chapter 6

Labeled Affinity Propagation to Predict Stock Performance

6.1 Introduction

Fundamental analysis is a method of equity prediction based on evaluating core company parameters such as revenue, expenses, assets, and liabilities, as well as certain ratios such as a company's net profit margin (net income/sales). According to Chavan and Patil, "Fundamental analysis mainly depends on statistical data of a company. It may include audit reports, financial status of the company, the quarterly balance sheets, the dividends and policies of the companies whose stock are to be observed. It also includes analysis of sales data, strength and investment of company, plant capacity, the competition, import/export volume, production indexes, price statistics, and the daily news or rumors about company" [15]. Augmenting the fundamental analysis of stocks with the use of machine learning techniques has shown promise because it allows researchers to quickly process large collections of stock data.

Several different machine learning approaches have been employed in the area of stock performance prediction, including linear regression, artificial neural networks, genetic algorithms, support vector machines, case-based reasoning, and others [81].

Using an artificial neural network to perform a fundamental analysis of stock returns, Yildiz and Yezegel found an average increase of 22.32% on returns when accounting for market risk and other effects [80]. In addition, they noted that strong evidence exists that the relationship between fundamental ratios and stock returns is non-linear and requires more complex techniques than linear estimation [80]. Wong *et al.* used a fuzzy neural system to address some of the shortcomings of a standard neural system; they used a conglomeration of the different techniques of expert systems and fuzzy reasoning in conjunction with a neural network [74]. Xia *et al.* used a support vector machine to build a regression model of historical time series data in an attempt to predict future trends of stock prices [75]. Although a number of different machine learning techniques have been used for stock prediction, the use of clustering algorithms has not been commonly applied for stock prediction. Specifically, affinity propagation does not appear to have been employed in this domain.

Parameters selected for these algorithms often follow the common financial ratios associated with fundamental analysis. For example, Kryzanowski *et al.* used an artificial neural network that was focused on a set of financial ratios, such as debt-to-equity and return on total assets [44]. In a study that used more traditional methods, a similar set of ratios was used for forecasting [64].

In this chapter the affinity propagation clustering algorithm is used in conjunction with pre-process performance information to attempt to predict future equity trends. The overall goal of this chapter is similar to the literature cited above, but the methodology is distinct. The question addressed here is whether the affinity propagation algorithm can be used to cluster historical stock data, including fundamental characteristics, in such a way

that future price performance can be predicted. The inclusion of a known high-performing stock is used to augment the predictive efficacy of the technique.

6.2 Methodology

For this chapter, fundamentals for a random set of companies were collected from January through December 2011 and corresponding company stock data were collected from January 3, 2012. In addition, specific data was collected regarding a known successful stock to serve as the pre-clustering process label. Netflix was selected because of its substantial rise over this period. For each stock (company) selected, an associated vector was created from the designated parameters. These vectors were then compared for similarity against each of the others (ALL versus ALL) using Euclidian distance. From these similarity comparisons, a matrix was constructed which served as the data to be fed forward to the affinity propagation application.

6.2.1 Data Used

Data was obtained from the Wharton Research Data Services database, which is available from the Wharton School at the University of Pennsylvania and provides both financial and market data for more than 13,000 international companies [72]. The database has access to a number of different datasets, but the Compustat dataset from Standard & Poor's was used to collect the necessary company and stock information. Compustat provides both current and historical information on publicly held companies in the United States and Canada, and contains annual and quarterly data items from

company income statements, balance sheets, statements of cash flow, and securities [72]. Within Compustat, data was drawn from the Fundamentals Annual module and the Security Daily module.

Data was collected that represented each company's financial profile for January, 2012, a date which was selected to correspond to Netflix's strong positive trend. In order to obtain company financials for this date within the Fundamentals Annual module in Compustat, annual data was selected using the dates of January 2011 through December 2011, and companies were searched by ticker symbol. Parameters were selected based on previous literature [44], which included important balance sheet and income statement items. However, unlike the studies cited above, these parameters were used directly without being combined into ratios. Within the Security Daily module, closing stock prices were obtained for the data of January 3, 2012, which was the first business day of the year. Table 6.1 shows a listing of parameters used for each equity's data vector.

Abbreviation	**Parameter**
AT	Assets - Total
DLTT	Long-Term Debt - Total
LCT	Current Liabilities - Total
SEQ	Stockholder's Equity - Total
ACT	Current Assets - Total
LT	Liabilities - Total
BKVLPS	Book Value Per Share
SALE	Sales/Turnover (Net)
NT	Net Income (Loss)
PRCCD	Price - Close - Daily

Table 6.1 Parameters Used for Each Equity's Data Vector

Data was collected for Netflix and other companies that are listed on the NASDAQ. A complete listing of 2,905 companies that are currently in the NASDAQ was obtained from its website [55]. A pseudo-random number generator was used to select 50 companies; however, some company parameters were not reported for the chosen time period. Eight companies with missing parameters were removed, resulting in a total of 42 companies for the analysis. Table 6.2 shows the final listing of the companies used in this study.

6.2.2 Functional Programming F#

An affinity propagation application was developed using the functional programming language F#. The paradigm switch from imperative languages to a functional language involves the realization that variable values should not be changed and are, in fact, immutable [30]. This feature especially enables high levels of concurrency with no specific need for the locking of variables [30].

Functional programming is frequently employed in the financial industry [10]. Its clarity in terms of representing mathematical models in a concise fashion makes it a good choice for computationally intensive algorithms. The concept of higher-order functions (i.e., functions that take other functions as parameters) is extremely beneficial in implementing complex models while maintaining modularity and maintainability. A specific advantage of F# is the inclusion of type providers, which allows for the quick integration of information sources (in this case, stock data) into a strongly typed data structure [65]. This work was developed in F# in order to make it accessible and extensible by those in the financial industry.

Ticker Symbol	Company	Ticker Symbol	Company
AAR	American Airlines Group Inc	MEOH	Methanex Corp
ADBE	Adobe Systems Inc	MKSI	Mks Instruments Inc
ANAC	Anacor Pharmaceuticals Inc	MPAA	Motorcar Parts Of Amer Inc
BLKB	Blackbaud Inc	NFLX	Netflix Inc
EPAY	Bottomline Technologies Inc	NUVA	Nuvasive Inc
CVCO	Cavco Industries Inc	PEBK	Peoples Bancorp Nc Inc
PLCE	Childrens Place Inc	PHIIK	Phi Inc
CTAS	Cintas Corp	PCH	Potlatch Corp
CRUS	Cirrus Logic Inc	PLPC	Preformed Line Products Co
CLCT	Collectors Universe Inc	QGEN	Qiagen Nv
CRVL	Corvel Corp	ROLL	Rbc Bearings Inc
DNKN	Dunkin' Brands Group Inc	REMY	Remy International Inc
FNGN	Financial Engines Inc	RGEN	Repligen Corp
FFIN	First Finl Bankshares Inc	SCHL	Scholastic Corp
GRMN	Garmin Ltd	SMTC	Semtech Corp
HSKA	Heska Corp	SWKS	Skyworks Solutions Inc
AWAY	Homeaway Inc	SLRC	Solar Capital Ltd
IIIN	Insteel Industries	TTWO	Take-Two Interactive Sftwr
ITRN	Ituran Location & Control	TREE	Tree.Com Inc
LORL	Loral Space & Communications	UBSI	United Bankshares Inc/Wv
MATW	Matthews Intl Corp	WFM	Whole Foods Market Inc

Table 6.2 Company Names and Ticker Symbols for the Final Data Clustering

6.2.3 Affinity Propagation Process

As described previously, affinity propagation is a modern clustering algorithm that has been shown to outperform many other algorithms in the field. The F# application expects two .CSV files—one representing the ALL versus ALL similarity matrix and the second is the preferences for each data item in terms of initial exemplar selection. The matrix was built in Microsoft Excel. The Euclidian distance for each stock vector was calculated. The entirety of this data was formatted as a collection of three columns

representing the similarity matrix, with the first two parts being the row/column label and the final being the similarity score between those particular stock vectors. A subset of the 42x42 similarity matrix is shown in Table 6.3.

	1	2	3	4	5	6	7
1	0	46535.41187	47924.2079	42895.628	47882.3918	47129.89579	47144.63765
2	46535.41187	0	2623.16441	4225.3858	2593.22523	1858.602999	2249.753049
3	47924.20791	2623.164407	0	6531.7843	180.119814	1012.482554	852.0179886
4	42895.62838	4225.38584	6531.78435	0	6484.72924	5573.578734	5851.429608
5	47882.39177	2593.225228	180.119814	6484.7292	0	1009.358568	840.6018729
6	47129.89579	1858.602999	1012.48255	5573.5787	1009.35857	0	445.7413542
7	47144.63765	2249.753049	852.017989	5851.4296	840.601873	445.7413542	0

Table 6.3: Subset of the 42 x 42 Similarity Matrix

Validation of the F# affinity propagation tool was accomplished by using Frey and Deuck's online clustering tool [2]. Results from an initial subset of nine companies were indistinguishable, which was expected since the source code for the C implementation of the algorithm was used as the basis for the F# development.

6.3 Results and Analysis

Multiple iterations of the process were undertaken. The test run of the application used a single label (Netflix: known price increase) and the first nine randomly selected stocks. This run served only as a validation of the correctness of the application.

The first run using the initial data collection used Netflix as the known outperforming stock. For this primary run 41 additional stocks were included. Eleven clusters were generated from this collection of 42 stock feature vectors. Analysis of these

clusters showed that the initial random draw of stocks resulted in limited predictive value. Since the stocks were selected from the NASDAQ in a random fashion, several had features that led directly to their clustering together. For example, several small companies with stock prices under $5 clustered together. The initial price data of the companies in each cluster were compared to current stock prices drawn from August 22, 2014 to see how they performed. Netflix was clustered singly during this first full run of the application and had the highest percent price increase of 563%.

A second approach in regard to data selection seemed advisable. For this approach only companies with somewhat similar qualities were used. These stocks were still randomly pulled from the NASDAQ index, but were only added to the collection of stocks presented to the application if they fell within a certain threshold of the label companies' parameters. Specifically, these companies needed to have a stock price of $15 or more and have meaningful asset levels.

The clustered results of the second run are illustrated in Figure 6.1. Coincidentally, 11 clusters were again produced, with seven stocks clustering singularly. Of the remaining 35 stocks, 17 clustered with the exemplar Cavco, seven clustered with the exemplar Phi Inc., seven clustered with the exemplar Take-Two Interactive, and finally, the remaining four clustered with the exemplar Methanex. This time, Netflix did not cluster singly and instead it was included in the Methanex cluster.

To determine a performance metric, the current price information was compared against the historical price information. Closing stock prices were again obtained for August 22, 2014 and compared against the data from January 3, 2012. The mean cluster price change is indicated below each cluster in Figure 6.1. In addition, in the Netflix

cluster the mean of the three other stocks was calculated excluding Netflix from the calculation. The three stocks in this cluster—Methanex, Cintas, and Qiagen—showed an average increase of 114%. Since the NASDAQ Composite index rose only 71% over the same period of time, these three stocks outperformed the NASDAQ Composite. This indicates that if this technique had been used in stock selection over this time period, it would have outperformed the NASDAQ Composite, resulting in a greater rate of return on the buyer's investment. However, it should be noted that this data consisted of a small sample set, and therefore additional iterations of the process would be necessary to refine and validate the technique.

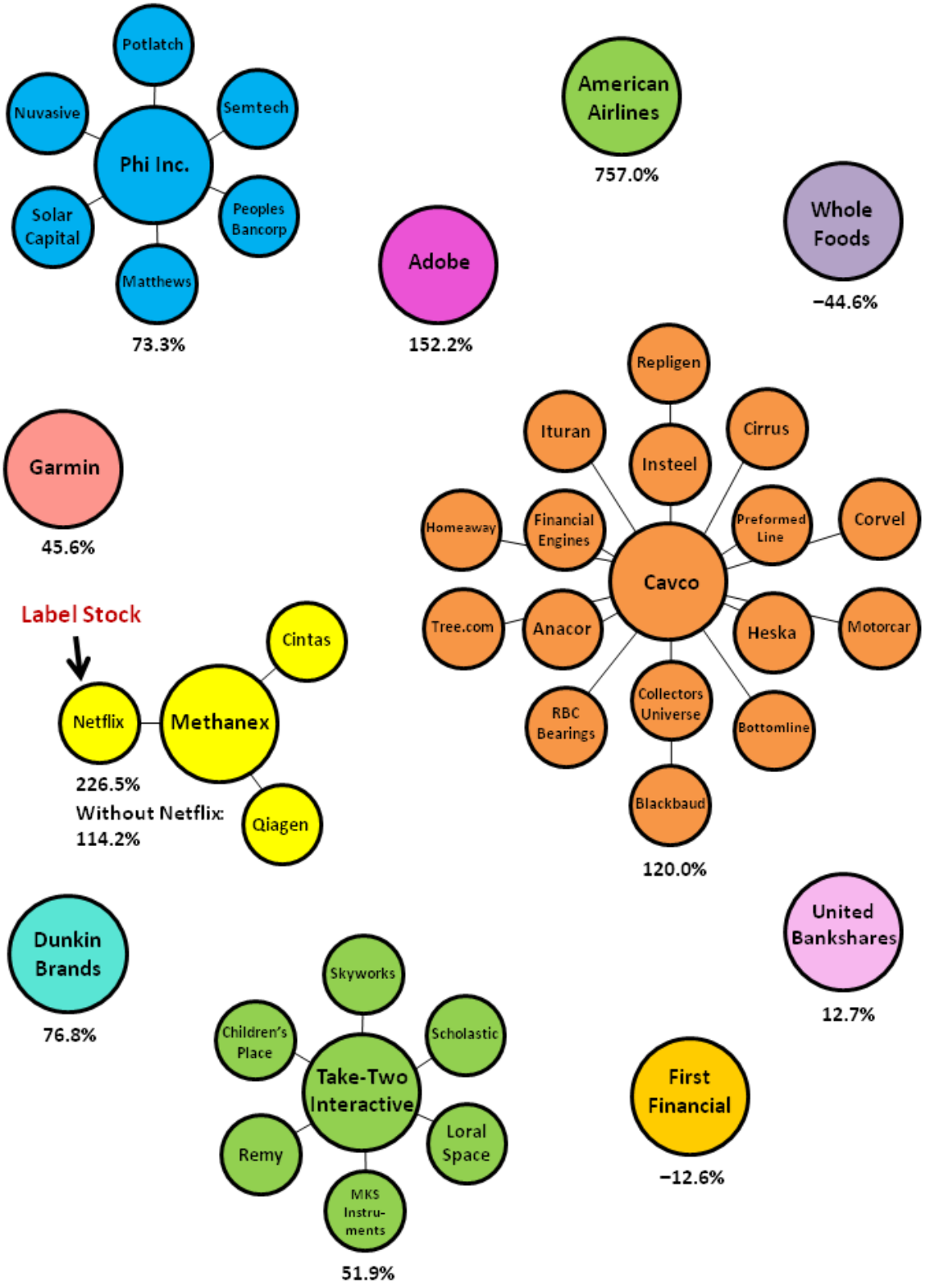


Figure 6.1: Clusters from the Second Data Run with the Mean Cluster Price Change Indicated

6.4 Conclusion

The naive approach attempted in the first run of this study showed a limited correlation to the prediction of the success of these stocks (in terms of rising price). However, with refinement of the initial vector selection to involve some additional analysis, a more predictive outcome could be achieved. The second approach in which stocks were chosen based on similarity to the label stock resulted in more intrinsic predictive value. The cluster including the label stock Netflix is considered the "winning" cluster for this analysis. In this case, winning is defined as the prerequisite inclusion of the label stock and each member in the designated cluster out-performing the market. The inclusion of Netflix with the three other stocks points to possible fundamental factors that may lead to market out-performance. It is worth nothing that one of the singleton clusters (American Airlines) performed better than the labeled cluster, and a second one (Adobe) also out-performed the labeled cluster when the calculation excluded Netflix. Nonetheless, these singleton clusters do not detract from the successful clustering of the label stock with other out-performers.

This tool, like similar methods, would be best used in conjunction with traditional analysis. The technique described herein was a limited investigation into its efficacy; possible extensions to the technique could involve a pre-process analysis to better select included features. Based on the initial results, another approach would be to use the initial clusters of the random collection as the basis for an additional run of the algorithm. For example, if in an initial run produced a cluster of low asset companies, then the companies in this cluster could be used as the basis for a more targeted clustering run.

This, in effect, combines the two approaches described above without requiring the need for filtering the selection of stocks.

Ultimately, the goal of the technique described here is to help elucidate a pattern of company fundamentals that may lead to better stock performance. When assessing the strength of a company, a potential investor could factor hundreds of different data points into an analysis—data that come from company balance sheets, income statements, audit reports, dividends, news stories, and other sources. Which fundamental data points are most critical? And which combination of these factors may correlate to success? A clustering tool such as the one described here could be refined to help investors make these determinations.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

As the amounts and complexity of data continue to surge, analysts are challenged to evaluate data effectively and to make decisions in a timely manner. It often becomes necessary to greatly automate the process of analysis by pairing down the collection of data before individual investigation can proceed. This work shows how the application of clustering algorithms can greatly enhance this process in a variety of fields. In addition, this work demonstrates that pre-process information can be used in conjunction with a clustering algorithm to enhance and direct the research process.

The applications described in this work show how the use of three different clustering tools can be used to augment a decision-making process, and each algorithm provides its own strengths in relation to the problem to be solved. The SOM inherently serves as an excellent decision-making tool with its natural representation of a visual map. It is flexible with a wide variety of datasets, such as those ranging from organizational parameters to metagenomic reads, as has been demonstrated here. K-means, as the seminal clustering algorithm, is a viable choice in many situations as well, although it does not produce the same relational output that is the staple of the SOM.

Both the SOM and the K-means are flexible in that they can accommodate extremely large datasets. The affinity propagation algorithm does have some constraining limitations on dataset sizes due to its use of an N x N matrix, which results in a $O(n^2)$ memory requirement. However, its ability to cluster effectively is well documented and shown here as well. With very large datasets, accommodations to the affinity propagation process must be made to allow a researcher to proceed with the algorithm [23]. In general, the three algorithms presented herein have different computational complexities as well as usage requirements.

This work has demonstrated the feasibility of using these techniques in various domains. The use of clustering as a decision-making tool has been rarely applied in library management specifically and in organizational management generally. Using machine-learning-driven quantitative analysis in conjunction with traditional operations planning (i.e., staff planning, asset allocation, budgeting) has enormous potential. As shown in Chapter 4, a cluster of high-scoring libraries (LPM) was found to have attributes other than those usually thought to have an impact on successful performance.

As has been demonstrated in previous research, clustering is well-suited to the burgeoning field of bioinformatics but its applicability to the subfield of metagenomics has had only limited exploration. Because data is so vast and diverse in the field of metagenomics, a researcher has no choice but to rely on the sorts of tools described in Chapter 5. Because metagenomics researchers are evaluating both the species present in samples and the biological mechanisms functioning within samples, they can greatly benefit from a robust clustering tool which directs them to a targeted subset of data for further investigation. The *Thermoanaerobacter siderophilus bacterium* found through a

BLAST similarity comparison with the consensus sequence of one of the cluster centers illustrates the effectiveness of the technique in quickly leading a researcher to focus his analysis.

Although various machine learning techniques have been used to facilitate stock performance analysis, no previous study has applied affinity propagation in this manner. In this study, the use of the label stock led to the rapid identification of other stocks that outperformed the NASDAQ Index. This not only represents a different approach but also shows that the technique can be used effectively in combination with company fundamentals. Similar to the strategy employed with the library study, the stock analysis employed in Chapter 6 shows the effectiveness of using a clustering tool in conjunction with traditional, domain-specific analysis.

This work not only demonstrates the applicability of various clustering techniques to different domains, but it also shows that the addition of various targeted labeling strategies, applied in a method that is a variation of semi-supervised learning, can augment the process of data analysis. Previous studies in the fields investigated here have not combined a calculated metric label before clustering. These results have shown this combination method to be successful in the rapid identification of critical values (e.g., high performers, representative organisms). With the library analysis, a label was applied to every data point and was used to determine clustered areas of successful libraries on the SOM. The library performance metric calculated for each institution provided a novel way of locating and interpreting the SOM results. Comparable performance metrics could be developed for several domains to enhance the knowledge discovery step of a clustering algorithm. With the metagenomics analysis, labeling was applied after the

clustering process to provide a researcher with a quick overview of possible reference genomes. With the financial analysis, a single known high-performer was included in the data collection to track the winning cluster (with "winning" defined as the prerequisite inclusion of the label stock and each member in the designated cluster out-performing the market). The diverse labeling strategies presented in this work can be used as a guide for future research by illustrating different ways in which labeling can be integrated into the clustering process to enhance knowledge discovery, a process which is dependent upon the goals of the project and the features of the data.

The applications described here do present some limitations. Each of these tools focuses on a subset of a larger collection of data as well as a limited number of available features. To truly embrace the full potential of these techniques, the applications would need to support all available data in an unfiltered manner. In addition, some of the correlations have not been analyzed in the full context of their domains. The data points for each application were selected because they frequently recurred in the literature as common points of analysis, but if trained to use these types of tools, domain experts could potentially incorporate more and varied data points. Finally, additional iterations employing these tools would be required to refine them and to increase their efficacy.

7.2 Future Work

Each of the applications described herein provide future opportunities to advance research in the domain. The work on organizational decision-making described in Chapter 4 could be expanded by conducting further analysis on the features of the high-

performing clusters or by identifying organizations that cluster within certain pre-established parameters. Both of these techniques might help leaders to arrive at a more detailed understanding of their organizational effectiveness and to identify similarly categorized organizations that can be used for benchmarking purposes. Chapter 5 offers a complete pipeline for metagenomic analysis, but future work could incorporate the pipeline into a single tool to be used by an investigator. This metagenomic tool was used to help an investigator quickly determine possible candidates to be used as reference genomes, but it could also be developed to focus the analysis in a number of different directions, including classification of sequences and the discovery of functional pathways. The affinity propagation-based technique described in Chapter 6 could be extended by focusing on company fundamentals that contribute the greatest to determining stock over-performance. The tool itself could be used in a multi-iterative process to group peer stocks and then subsequently to filter out high performers.

The major contribution of this work is that it provides researchers with guidance about both the applicability of clustering algorithms to problems in different fields and the use of labeling strategies to enhance knowledge discovery in these domains. Essentially, this work can be conceptualized as a matrix of clustering algorithms and problem domains, with the type of clustering algorithm occupying one axis and the domains occupying the other. A future expansion of this matrix could be greatly beneficial in determining both the efficacy of clustering in additional domains and the particular algorithms most applicable to specific fields.

Bibliography

[1] S. Abubucker, N. Segata, J. Goll, A. M. Schubert, J. Izard, B. L. Cantarel, B. Rodriguez-Mueller, J. Zucker, M. Thiagarajan, B. Henrissat, O. White, S. T. Kelley, B. Methé, P. D. Schloss, D. Gevers, M. Mitreva, and C. Huttenhower, "Metabolic reconstruction for metagenomic data and its application to the human microbiome," *PLoS Comput. Biol.*, vol. 8, no. 6, p. e1002358, Jan. 2012.

[2] Affinity Propagation Web Application, Probabilistic and Statistical Inference Group, University of Toronto. Available: http://www.psi.toronto.edu/affinitypropagation/webapp [Accessed: 6 Oct. 2014].

[3] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*. Hoboken: Chapman and Hall/CRC, 2013.

[4] N. Ahmad, D. Alahakoon, and R. Chau, "Classification of protein sequences using the growing self-organizing map," in *4th International Conference on Information and Automation for Sustainability*, pp. 167–172, 2008.

[5] H. Allen and M. P. Taylor, "The use of technical analysis in the foreign exchange market," *J. Int. Money Financ.*, vol. 11, no. 3, pp. 304–314, 1992.

[6] S. F. Altschup, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, pp. 403–410, 1990.

[7] L. An, J. Zhang, and C. Yu, "The visual subject analysis of library and information science journals with self-organizing map," *Knowl. Organ.*, vol. 38, no. 4, pp. 299–319, 2011.

[8] Association of College and Research Libraries Metrics. Available: http://www.acrlmetrics.com [Accessed: 6 Oct. 2014].

[9] Association of College and Research Libraries, "Standards for libraries in higher education," 2011. Available: http://www.ala.org/acrl/sites/ala.org.acrl/files/content/standards/slhe.pdf [Accessed: 6 Oct. 2014].

[10] J. Astborg, *F# for Quantitative Finance*. Birmingham: Packt Publishing, 2013.

[11] E. Bair, “Semi-supervised clustering methods,” *arXiv.org*, 2013. Available: http://arxiv.org/pdf/1307.0252.pdf [Accessed: 6 Oct. 2014].

[12] A. L. Bazinet and M. P. Cummings, “A comparative evaluation of sequence classification programs,” *BMC Bioinformatics*, vol. 13, p. 92, 2012.

[13] BLAST (Basic Local Alignment Search Tool), National Center for Biotechnology Information. Available: http://blast.ncbi.nlm.nih.gov/Blast.cgi [Accessed: 6 Oct. 2014].

[14] M. Van den Bogaerd and W. Aerts, “Applying machine learning in accounting research,” *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13414–13424, Sep. 2011.

[15] P. S. Chavan, “Parameters for stock market prediction,” *Int. J. Computer Technology & Applications*, vol. 4, no. 2, pp. 337–340, 2013.

[16] A. Cobo and G. Schneider, “Fuzzy clustering: application on organizational metaphors in Brazilian companies,” *J. Inf. Syst. Technol. Manag.*, vol. 9, no. 2, pp. 197–212, 2012.

[17] C. Ding and J. C. Patra, “User modeling for personalized web search with self-organizing map,” *J. Am. Soc. Inf. Sci. Technol.*, vol. 58, no. 4, pp. 494–507, 2007.

[18] R. E. Dugan, P. Hernon, and D. A. Nitecki, *Viewing Library Metrics from Different Perspectives: Inputs, Outputs and Outcomes*. Santa Barbara: Libraries Unlimited, 2009.

[19] R. C. Edgar, “Search and clustering orders of magnitude faster than BLAST,” *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, Oct. 2010.

[20] E. Elayaraja, K. Thangavel, M. Chitralegha, and T. Chandrasekhar, “Extraction of motif patterns from protein sequences using SVD with rough K-means algorithm,” *Int. J. Comput. Sci. Issues*, vol. 9, no. 6, pp. 350–356, 2012.

[21] M. Emmons and F. C. Wilkinson, “The academic library impact on student persistence,” *Coll. Res. Libr.*, vol. 72, no. 2, pp. 128–149, 2011.

[22] D. Ennis, A. Medaille, T. Lambert, R. Kelley, and F. C. Harris, “A comparison of academic libraries: an analysis using a self-organizing map,” *Perform. Meas. Metrics*, vol. 14, no. 2, pp. 118–131, 2013.

[23] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[24] T. S. Ghosh, M. H. Mohammed, H. Rajasingh, S. Chadaram, and S. S. Mande, "HabiSign: a novel approach for comparison of metagenomes and rapid identification of habitat-specific sequences," *BMC Bioinformatics*, vol. 12 (Suppl. 13), p. S9, Nov. 2011.

[25] G. C. Hadjinicola, C. Charalambous, and E. Muller, "Product positioning using a self-organizing map and the rings of influence," *Decis. Sci.*, vol. 44, no. 3, pp. 431–461, Jun. 2013.

[26] L. Hamel and G. Sun, "Toward protein structure analysis with self-organizing maps," in *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pp. 1–8, 2005.

[27] J. Handelsman, M. R. Rondon, S. F. Brady, J. Clardy, and R. M. Goodman, "Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products," *Chem. Biol.*, vol. 5, no. 10, pp. R245–R249, Oct. 1998.

[28] J. A. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.

[29] J. A. Hartigan and M. A. Wong, "Algorithm AS136: a K-means clustering algorithm," *J. R. Stat. Soc.*, vol. 28, no. 1, pp. 100–108, 1979.

[30] K. Hinsen, "The promises of functional programming," *Comput. Sci. Eng.*, vol. 11, no. 4, pp. 86–90, 2009.

[31] N. A. Van House, B. T. Weil, and C. R. McClure, *Measuring Academic Library Performance: A Practical Approach*. Chicago: American Library Association, 1990.

[32] S.-Y. Huang, R.-H. Tsaih, and W.-Y. Lin, "Unsupervised neural networks approach for understanding fraudulent financial reporting," *Ind. Manag. Data Syst.*, vol. 112, no. 2, pp. 224–244, 2012.

[33] Integrated Microbial Genomes with Microbiome Samples, Joint Genome Institute. Available: https://img.jgi.doe.gov/cgi-bin/m/main.cgi [Accessed: 6 Oct. 2014].

[34] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[35] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 2000.

[36] D. R. Kelley and S. L. Salzberg, “Clustering metagenomic sequences with interpolated Markov models,” *BMC Bioinformatics*, vol. 11, no. 1, p. 544, Jan. 2010.

[37] M. Y. Kiang and D. M. Fisher, “Selecting the right MBA schools—an application of self-organizing map networks,” *Expert Syst. Appl.*, vol. 35, no. 3, pp. 946–955, Oct. 2008.

[38] C. Kim, H. Lee, and Y. Park, “A taxonomical classification of business models on mobile business: patent analysis and SOM mapping,” in *IEEE International Conference on Management of Innovation and Technology*, 2006, vol. 1, pp. 478–482.

[39] K. Kim and H. Ahn, “Simultaneous optimization of artificial neural networks for financial forecasting,” *Appl. Intell.*, vol. 36, no. 4, pp. 887–898, Jun. 2011.

[40] J. J. Knightly, “Overcoming the criterion problem in the evaluation of library performance.,” *Spec. Libr.*, vol. 70, no. 4, pp. 173–177, 1979.

[41] T. Kohonen, “Self-organizing maps,” *Neurocomputing*, vol. 21, pp. 1–6, 1998.

[42] T. Kohonen, *The Self-Organizing Map*, 3rd ed. Berlin: Springer, 2001.

[43] A. R. Konicek, J. Lefman, and C. Szakal, “Automated correlation and classification of secondary ion mass spectrometry images using a k-means cluster method,” *Analyst*, vol. 137, no. 15, pp. 3479–87, Aug. 2012.

[44] L. Kryzanowski, M. Galler, D. W. Wright, and M. Galier, “Using artificial neural networks to pick stocks,” *Financ. Anal. J.*, vol. 49, no. 4, pp. 21–27, 2014.

[45] W. Li, L. Fu, B. Niu, S. Wu, and J. Wooley, “Ultrafast clustering algorithms for metagenomic sequence analysis,” *Brief. Bioinform.*, vol. 13, no. 6, pp. 656–668, Nov. 2012.

[46] W. Li and A. Godzik, “CD-HIT: a fast program for clustering and comparing large sets of protein or nucleotide sequences,” *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, Jul. 2006.

[47] W. Lin, Y. Hu, and C. Tsai, “Machine learning in financial crisis prediction: a survey,” *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.*, vol. 42, no. 4, pp. 421–436, Jul. 2012.

[48] J. D. Linton, M. Himel, and M. J. Embrechts, “Mapping the structure of research: business and management as an exemplar,” *Ser. Rev.*, vol. 35, no. 4, pp. 218–227, Dec. 2009.

[49] M. Lorr, *Cluster Analysis for Social Scientists*. San Francisco: Jossey Bass Publishers, 1983.

[50] S. Marsland, *Machine Learning: An Algorithmic Perspective*. Boca Raton: CRC Press, 2011.

[51] J. A. McDonald and L. B. Micikas, *Academic Libraries: The Dimensions of Their Effectiveness*. Westport: Greenwood Press, 1994.

[52] K. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge: MIT Press, 2012.

[53] F. Murtagh and M. Hernández-Pajares, "The Kohonen self-organizing map method: an assessment," *J. Classif.*, vol. 12, no. 2, pp. 165–190, Sep. 1995.

[54] T. Naenna, R. A. Bress, and M. J. Embrechts, "DNA classifications with self-organizing maps (SOMs)," in *IEEE International Workshop on Soft Computing in Industrial Applications*, 2003, pp. 151–154.

[55] NASDAQ.com, Company List (NASDAQ, NYSE, & AMEX). Available: http://www.nasdaq.com/screening/company-list.aspx [Accessed: 6 Oct. 2014].

[56] M. J. Oakleaf, *The Value of Academic Libraries: A Comprehensive Research Review and Report*, American Library Association, Chicago, 2010.

[57] N. G. Pavlidis, V. P. Plagianakos, D. K. Tasoulis, and M. N. Vrahatis, "Financial forecasting through unsupervised clustering and neural networks," *Oper. Res.*, vol. 6, no. 2, pp. 103–127, May 2006.

[58] Y. Peng, G. Kou, A. Sabatka, Z. Chen, D. Khazanchil, and Y. Shi, "Application of clustering methods to health insurance fraud detection," in *International Conference on Service Systems and Service Management*, 2006, pp. 116–120.

[59] R. Poll, "Measuring impact and outcome of libraries," *Perform. Meas. Metrics*, vol. 4, no. 1, pp. 5–12, 2003.

[60] N. Powell, S. Y. Foo, and M. Weatherspoon, "Supervised and unsupervised methods for stock trend forecasting," in *40th Southeastern Symposium on System Theory*, 2008, pp. 203–205.

[61] S. M. Pritchard, "Determining quality in academic libraries," *Libr. Trends*, vol. 44, no. 3, pp. 572–594, 1996.

[62] Z. Rong, *Machine Learning Approaches to Bioinformatics*. Singapore: World Scientific Publishing Company, 2010.

[63] M. W. Rutherford, P. McMullen, and S. Oswald, “Examining the issue of size and the small business: a self-organizing map approach,” *J. Bus. Econ. Stud.*, vol. 7, no. 2, pp. 64–79, 2001.

[64] M. Sharma and Preeti, “Prediction of stock returns for growth firms—a fundamental analysis,” *J. Bus. Perspect.*, vol. 13, no. 3, pp. 31–40, 2009.

[65] D. Syme, K. Battocchi, K. Takeda, D. Malayeri, and T. Petricek, “Themes in information-rich functional programming for internet-scale data sources,” in *Proceedings of the 2013 Workshop on Data Driven Functional Programming*, 2013, p. 1.

[66] C.-F. Tsai, Y.-C. Lin, and Y.-T. Wang, “Discovering stock trading preferences by self-organizing maps and decision trees,” *Int. J. Artif. Intell. Tools*, vol. 18, no. 4, pp. 603–611, Aug. 2009.

[67] J. Vlasblom and S. J. Wodak, “Markov clustering versus affinity propagation for the partitioning of protein interaction graphs,” *BMC Bioinformatics*, vol. 10, p. 99, Jan. 2009.

[68] R. Wang, “Stock selection based on data clustering method,” in *Seventh International Conference on Computational Intelligence and Security*, 2011, no. 5, pp. 1542–1545.

[69] M. Weber, H. Teeling, S. Huang, J. Waldmann, M. Kassabgy, B. M. Fuchs, A. Klindworth, C. Klockow, A. Wichels, G. Gerdts, R. Amann, and F. O. Glöckner, “Practical application of self-organizing maps to interrelate biodiversity and functional data in NGS-based metagenomics,” *ISME J.*, vol. 5, pp. 918–928, May 2011.

[70] S. A. Weiner, “Library quality and impact: is there a relationship between new measures and traditional measures?” *J. Acad. Librarianship*, vol. 31, no. 5, pp. 432–437, Sep. 2005.

[71] G. Weiyi, “SOM clustering analysis for telecommunication customer segmentation,” in *International Conference on Management and Service Science*, 2009, pp. 1–4.

[72] Wharton Reserach Data Services, The Wharton School at the University of Pennsylvania. Available: https://wrds-web.wharton.upenn.edu/wrds [Accessed: 6 Oct. 2014].

[73] E. Whitmire, “Academic library performance measures and undergraduates’ library use and educational outcomes,” *Libr. Inf. Sci. Res.*, vol. 24, no. 2, pp. 107–128, Jan. 2002.

[74] F. S. Wong, P. Z. Wang, and T. H. Goh, “Fuzzy neural systems for stock selection,” vol. 48, no. 1, pp. 47–52, 2014.

[75] Y. Xia, Y. Liu, and Z. Chen, “Support vector regression for prediction of stock trend,” in *6th International Conference on Information Management, Innovation Management and Industrial Engineering*, 2013, pp. 123–126.

[76] J. Xiong, *Essential Bioinformatics*. Cambridge: Cambridge University Press, 2006.

[77] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[78] F. Yang, Q. Zhu, D. Tang, and M. Zhao, “Evolutionary bioinformatics clustering protein sequences using affinity propagation based on an improved similarity measure,” *Evol. Bioinform.*, vol. 5, pp. 137–146, 2009.

[79] H.-C. Yang, H.-W. Hsiao, and C.-H. Lee, “Multilingual document mining and navigation using self-organizing maps,” *Inf. Process. Manag.*, vol. 47, no. 5, pp. 647–666, Sep. 2011.

[80] B. Yildiz and A. Yezegel, “Fundamental analysis with artificial neural network,” *Int. J. Bus. Financ. Res.*, vol. 4, no. 1, pp. 149–159, 2010.

[81] P. D. Yoo, M. H. Kim, and T. Jan, “Machine learning techniques and use of event information for stock market prediction: a survey and evaluation,” in *Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation*, 2007, pp. 1–7.

[82] J. Zhang and L. An, “Visual component plane analysis for the medical subjects based on a transaction log,” *Can. J. Inf. Libr. Sci.*, vol. 34, no. 1, pp. 83–111, 2010.

[83] J. Zhang, L. An, T. Tang, and Y. Hong, “Visual health subject directory analysis based on users’ traversal activities,” *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 10, pp. 1977–1994, 2009.

[84] W. Zhong, G. Altun, R. Harrison, P. C. Tai, and Y. Pan, “Improved K-means clustering algorithm for representing common structural property,” *IEEE Trans. Nanobioscience*, vol. 4, no. 3, pp. 255–265, 2005.

[85] M. Zvelebil and J. O. Baum, *Understanding Bioinformatics*. New York: Garland Science, 2008.

Appendix. XML File Formats

MappedSequences.xml

<?xml version="1.0" encoding="UTF-8"?>

-<MappedData>

<Data
Squence="CATTCTTATATCTCTTGTTCCGCCTTCATAATCTTACACCTGGCCTTC
CTCCTATTACGAACCTTCCACCGCGTAATTTACCTCTTACAGAACTCAAACA">
GCXNWHH02GP572,18,22,TTTTGCTTTATATTTTTCTTTATCTTCAAATTTAAGT
TTTTCATTTTTTTCCACTAAATCTATTACCACTTCTAAGTAAATAACTTTAAGA
AGTTCGACT</Data>

<Data
Squence="GAACGTCATTAGGATACCGACTACGCAGAAGAGATGACAACTTCA
CCAACTACAATAAGATGCAGTCCACTATAGCTATGTTGGCGCAGGCGAACGC
ATT">GCXNWHH02HND40,25,9,AAAGAGAAAAGATTAAAAAGAAAATAGAAA
AAAAAAAAAACGATGAAGATAAAATGAAAATGGTAAACAATAAAAATATTG
AAAAAGATAAAAAAAAAAA</Data>

<Data
Squence="TTTACAGATATGCGCTCCTGCTCTATTCCCAAAGCAAGGAAGATTC
TGTATTATCTTGTTTTGATAGGTGCAATCTTTGGCTCGTCTGCAATTAAAGTGC
">GCXNWHH02HFAV5,27,29,TTTTAGAACCTTACTTTTTTCTTTGTTTTTACTAA
TGTAAAACATTTTTGAGCTATTTTTTTTCATCCTTTTATTAATTCTTGTCTTATC
CATACTATCA</Data>

<Data
Squence="CTCATTATTCAGACTTCTGGATCCTTGATATATGGAGCTATGTGTCC
ATTCTCTCTTATAGGTGTTGCAATCCAGTATTTCCGTTAGTTGGCACCGAGGC"
>GCXNWHH02FVWJK,7,11,TTTTTTATAAAGTCTTAAAAATTGTCTATGGTTTT

CTTGGATTTTCACTATATTACCAAGAAAAGGAGTTACTAAAGTTTATCTTTGT
AAGAATTAATGA</Data>

...

Labeled ConsensusSequences.xml

<?xml version="1.0" encoding="UTF-8"?>

-<ConsensusSequences>

<Sequence> <Location>0,10</Location> <Count>28</Count>
<Weights>TTTTACTTCATTTTCCAATTACTTAAAGCTTTATTATTCTTTTTTCTC
TATTCTTTCCATATTATTTTTTATATTCTCTACTTTTTTTTTTGTATTTTTTT</Wei
ghts>

<Consesus>TTTTGAATTCTTTTCCATATTGACTAAGAGCTCATTGTGTTTCGTCT
TCATAGTTCCCATTATCTAATCTATAACCTTTACTTTTTTTTCCTATTTTTTT</C
onsensus> </Sequence>

<Sequence> <Location>24,7</Location> <Count>29</Count>
<Weights>TTTTTGAGAAAAAGAAAAAGAAGATAAAAAAAAAAAAAAAGAAT
GAAGATAGAAGAAGAATTGAGAGGAGTAAGGGAAAGGGTTAAGAAAGTTAA
TAAAA</Weights>
<Consesus>GATTTTGGAAAAAGATTGCGAAGAAAAGAAAGAAAAAGAAGAGT
AAATGTATATGATAAAAAGGTAAAGGTAGAGGAAAGGGTTAATAATGTACCT
ATTA</Consensus> <TopBlastResult>NC_018664.1 Clostridium
acidurici</TopBlastResult> </Sequence>

<Sequence> <Location>16,11</Location> <Count>29</Count>
<Weights>TTTCTTTTTTGAACTTTACCTTTTGCTACTTTGGAACTTTTTAAAACA
CCTAGAGCATTAACAACAAACCCTTCTACTGTTTCTTTTAAATCTTCTACCT</
Weights>
<Consesus>TTTCAGTTTTGACCATCATCCTTTACTATTTTCGTACTTTTCTCAAC
TCCAACAGCAATGAAAGCAAACATTTAAGCAGTTTCTTTTAAATCACCTTCTT
</Consensus> <TopBlastResult>BAFA01000036.1 Staphylococcus
aureus</TopBlastResult>

</Sequence> -<Sequence> <Location>29,0</Location> <Count>30</Count>
<Weights>TTTGGAAGAGATTTGAAAGAAGAAAGATAGAAAAAACTTAGGAAC
TAAAGAAATAAGAGCGCATAAGAGAAACAAGAAAAAAAAAAAGAGATTTTTT
GGAA</Weights>
<Consesus>TTTGGAAGGTATGTTAAAAAAGAAAAAAAGAAAATACGTAGGAA
TAACTATAATAAGACCTCATAGTATAAACCAGGGAAAAATAGGAATTTTTAT
GGAT</Consensus> <TopBlastResult>AKXG02000035.1 Leptospira interrogans
serovar Grippotyphosa</TopBlastResult> </Sequence>

...