Methods in Clustering

Katelyn Gao[*] Heather Hardeman[†] Edward Lim[‡]
Cristian Potter[§] Carl Meyer[¶] Ralph Abbey[¶]

August 5, 2011

Abstract

Cluster Analytics helps to analyze the massive amounts of data which have accrued in this technological age. It employs the idea of clustering, or grouping, objects with similar traits within the data. The benefit of clustering is that the methods do not require any prior knowledge of the data. Hence, through cluster analysis, interpreting large data sets becomes, in most cases, much easier. However one of the major challenges in cluster analytics is determining the exact number of clusters, k, within the data. For methods such as k-means and nonnegative matrix factorization, choosing the appropriate k is important. Other methods such as Reverse Simon-Ando are not as dependent on beginning with the correct k. In this paper, we discuss these methods and apply them to several well-known data sets. We then explore techniques of deriving the number of clusters from the data set and lastly several points of theoretical interest.

1 Introduction

Cluster Analysis is a type of data analysis that partitions observations into groups so that observations in the same group have similar characteristics.

[*] MIT, Cambridge, MA 02139
[†] University of Montevallo, Montevallo, AL 35115
[‡] Johns Hopkins University, Baltimore, MD 21218
[§] East Carolina University, Greenville, NC 27858
[¶] North Carolina State University, Raleigh, NC 27695

On numerical data, numbers that are close to one another would be clustered together. Nonnumerical data, such as text data, can be converted to numerical data and be clustered according to how often an observation is associated with another observation.

Human brains are naturally good at identifying groups and patterns. However, there are limits to being able to manually sort through and categorize data without the aid of computers. Data is often overwhelming in volume. Numerical data of large dimensions are impossible to visualize. Computers also perform computation much quicker than humans do. These traits make computers ideal for treatment of large sets of numerical data for clustering.

There are numerous algorithms that cluster n data points into k clusters when k is specified. However, in many situations k is unknown it is often necessary to guess k before running an algorithm. Our ultimate task is to explore a method of deriving k from the data set by analysis of eigenvalues. In Section 2, we outline several clustering methods that require the user to input the number of clusters. In Section 3, we describe a relatively new clustering method based on the Simon-Ando Theory. In Sections 4, 5, 6, and 7, we show the results of applying those methods on some well-known data sets. In Section 8, we discuss the method of determining k, and Section 9 explores the uncoupling measure of a matrix, about which we have made several interesting observations.

2 Background

2.1 Clustering Methods

2.1.1 k-means

k-means is a two-step iterative algorithm that assigns n observations into a user defined k number of clusters. The algorithm can be seen as an optimization problem that seeks to find the clustering M that minimizes the following objective function:

$$\sum_{i=1}^{k} \sum_{x_j \in M_i} ||x_j - \mu_i||^2$$

where μ_i is the centroid of the points in cluster i.

The algorithm initially randomly selects the clusters and their centroids. In the assignment step, each observation is clustered with the mean closest to it, as measured by a distance metric. The most common distance metric used is the 2-norm, or the sum of square differences. In the update step, new centroids are calculated amongst the points that were grouped together. The algorithm repeats the assignment and update steps until the assignments no longer change or until it has reached the maximum number of iterations a user has specified.

k-means is useful because it is a simple algorithm and in most practical settings converges quickly. k-means works best on globular shaped data and data that has distinct clusters that spaced apart.

2.1.2 Nonnegative Matrix Factorization (NMF)

In some cases, data is nonnegative. Even if it is not, it is possible to adjust it to make it nonnegative without destroying the structure the data. Nonnegative Matrix Factorization is an algorithm developed by Lee and Seung [?] that factors a $m \times n$ nonnegative matrix V into two nonnegative matrices W, which is $m \times k$ and H, which is $k \times n$ such that the divergence, or the norm of $V - WH$ is minimized. The k is specified by user input, and may be the number of clusters.

There are several ways in which W and H may be found, but we used Lee and Seungs multiplicative update method that minimizes the Frobenius norm of $V - WH$. The algorithm is outlined below:

1. Initialize W and H randomly.
2. $H_{ij} \leftarrow H_{ij} \frac{(W^T V)_{ij}}{(W^T W H)_{ij}}$ $W_{ij} \leftarrow W_{ij} \frac{(V H^T)_{ij}}{(W H H^T)_{ij}}$
3. Stop when the objective function is below a certain tolerance or the number of iterations exceeds the maximum.

The advantage of NMF lies in the fact that the non-negativity constraint allows the data to be represented by purely additive linear combinations of "basis vectors". In other factorizations, such as the Singular Value Decomposition (SVD), a negative component may exist in the coefficients, rendering that component un-interpretable. If the NMF factorization happens to yield a sparse representation, meaning there are many zeros in H, interpretation becomes even easier.

The results of NMF are not unique and depend on initialization, which is random. To use NMF as a clustering algorithm, we take the maximum element in each column of H and place the corresponding observation in the cluster corresponding to the row of the maximum element.

2.2 Other tools

2.2.1 Consensus Matrix

The adjacency matrix is a tool that indicates how often observations cluster with one another. After k-means, NMF, or any other clustering algorithm is run, the adjacency matrix will place a 1 in its ij-th entry if observations i and j have been clustered together and 0 if they have not. We create an adjacency matrix each time we run an algorithm and later calculate an consensus matrix by taking the average of the adjacency matrices.

2.2.2 Sinkhorn-Knopp Algorithm

The Sinkhorn-Knopp algorithm is another tool we used. The algorithm converts a nonnegative square matrix into a doubly stochastic matrix by alternately scaling the row sums and the column sums.

The algorithm places all the eigenvalues of a matrix into the interval $[0, 1]$ while preserving its structure. In addition, if the original matrix is symmetric, the new, doubly stochastic matrix is as well.

3 Simon-Ando Theory

This theory utilizes the clustering methods and tools introduced above and proposes a method for determining k.

3.1 Description

The Simon-Ando theory aims to predict the long term equilibrium distribution of macroeconomic systems by observing the distribution of the current state and only a few future states. The idea was that each micro-economy could be studied as a separate entity and later be conglomerated to model the macroeconomy. In the terminology of Markov chains, we can view the macroeconomy as a transition matrix that is nearly uncoupled.

An uncoupled matrix is a block matrix with zeros on the off diagonal blocks; a nearly uncoupled matrix would have very small numbers on the off diagonal blocks. In the early stages of the evolution of the nearly uncoupled matrix, the relationships within individual block matrices will dominate the relationships between the blocks. Hence, the distribution will start evolving as if the blocks were independent from each other, as if the transition matrix is uncoupled. The distribution will settle to a temporary equilibrium. This is called the short run stabilization period.

As we move away from the early stage of the evolution, we enter a mid-run relative equilibrium. The distribution will move away from the temporary equilibrium and move to another temporary equilibrium as the near zeros in the off diagonal entries start exerting their influence. Although the distribution changes, the ratios between elements in each block matrix will remain approximately equal.

The final stage of the evolution is called the global equilibrium. The distribution will settle down to a limit that is no longer changing. The ratios between elements in each block matrix still remain approximately equal.

3.2 Reverse Simon-Ando

In our research we reverse the process of the Simon-Ando theory. The idea is that if we know the limiting distribution, we can trace back the origin and determine information about the micro-economies, which we view as clusters. We assume that if clusters are distinct, the data will yield the structure of a nearly uncoupled matrix. Since data often comes unordered, the actual matrix may be a permutation of a nearly coupled matrix.

The Sinkhorn-Knopp algorithm turns a consensus matrix into a doubly stochastic matrix. A doubly stochastic matrix has a limiting distribution that is uniform. Because the ratios of the distribution remain approximately constant throughout the evolution, by the Simon-Ando theory, data points in the same cluster have approximately equal probabilities in the short run stabilization period.

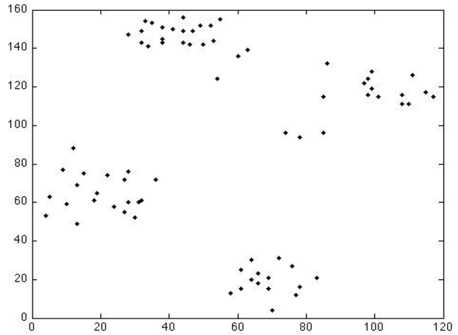
We execute the Sinkhorn Knopp algorithm on consensus matrices compiled from runs from k-means and NMF to get doubly stochastic matrices P. Since eigenvalues are continuous functions of the entries of the matrix, small changes in the entries of the matrix will result in small changes in eigenvalues. Hence, the eigenvalues of a nearly uncoupled matrix will be close to that of a corresponding uncoupled matrix. Suppose the data set is a completely uncoupled $n \times n$ matrix with three blocks. Then, the matrix P would give n eigenvalues, three of them one. By the continuity of eigenvalues, if the data is nearly uncoupled we would have n eigenvalues with one equal to one and two being close to one. Therefore, the number of clusters would be approximately equal to the number of eigenvalues close to one.

Our implementation of the method is as follows. We initialized the probability distribution randomly. Then, we looked at the sorted list of eigenvalues of the consensus matrix, from largest to smallest, after the Sinkhorn-Knopp algorithm has been applied to it. The number of eigenvalues prior to the biggest gap between consecutive numbers in that list is the number of clusters k. To identify the short-run equilibrium, we stopped the algorithm when consecutive iterations of the probability distribution were close enough. To cluster, we sorted the probability distribution and then, based on the $k - 1$ largest gaps between consecutive elements, divided the data into k groups.

4 Ruspini

4.1 *Description*

The Ruspini data set, first given in [**?**], is a collection of 75 points, arranged in four groups, in the Euclidean plane.

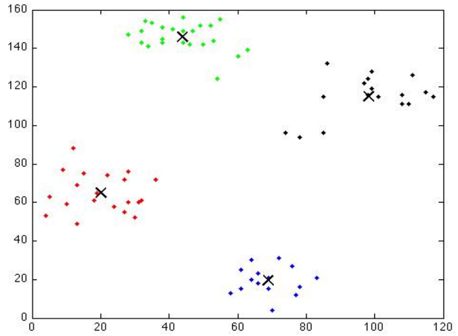


It is widely used to illustrate the effectiveness of clustering methods. In the following three sections, we will show the results of applying each of the three clustering methods discussed in Section 2 to this data set.

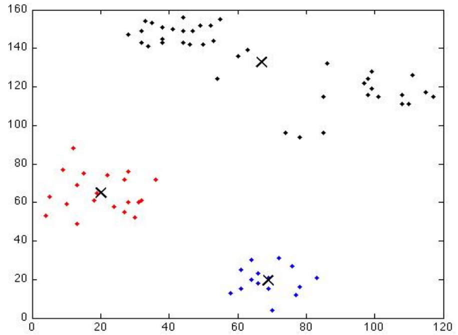
4.2 k-means

We implemented our own version of the k-means algorithm discussed in Section 2.1, with $k = 4$ as we know there are four clusters. In the initialization step, we first randomly assigned clusters to each point and then calculated the centroids. The algorithm stops when the cluster assignments do not change after an iteration of the assignment and update steps.

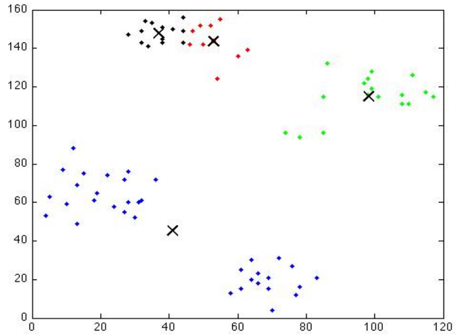
Because of the element of randomness in the algorithm, the clustering results are different every time. In the following figures, the crosses indicate the cluster centroids. Sometimes the clusters are correct:



At times one or more of the clusters are empty:



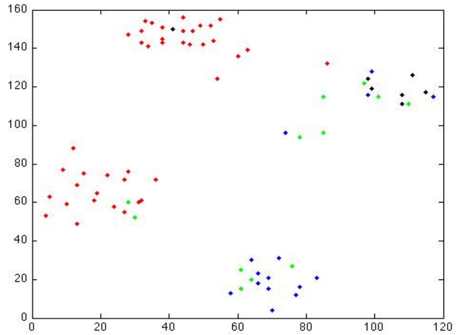
Lastly, sometimes one cluster is split into two, and two other clusters are combined into one:



4.3 NMF

We also implemented our own version of the NMF algorithm discussed in Section 2.2. Since the Ruspini data matrix is 2×75 and there are four clusters, we chose W to be 2×4 and H to be 4×75.

The following figure depicts the results of NMF clustering on the Ruspini data set.

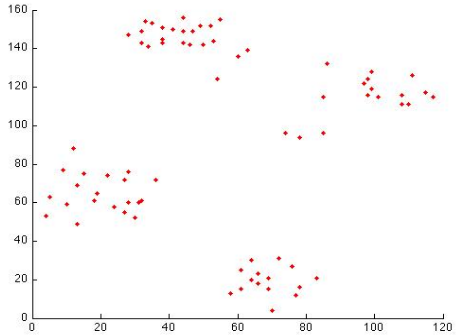


It is clear that NMF does not work well on the Ruspini data set. We suspect that this is due to an inherent weakness of the process by which clusters are assigned.

Suppose that the first column of H is as follows: $[0.45, 0.46, 0.05, 0.04]$. Our algorithm definitively places the first data point in cluster 2, but in actuality it belongs neither in cluster 1 completely nor cluster 2 completely. Therefore, situations like this could lead to poor clustering.

4.4 Reverse Simon-Ando

As with the k-means and NMF methods, we implemented our own reverse Simon-Ando algorithm. The following figure depicts the result of the reverse Simon-Ando method on the Ruspini data set. To create the consensus matrix, we ran the k-means algorithm with $k = 2$, $k = 3$, $k = 4$, $k = 5$, and $k = 6$, ten times each.



Clearly, there is only one cluster; the largest gap in the sorted list of eigenvalues occurred between the first and second elements. This suggests that the method of finding the number of clusters described above is unreliable and needs to be improved.

5 Leukemia

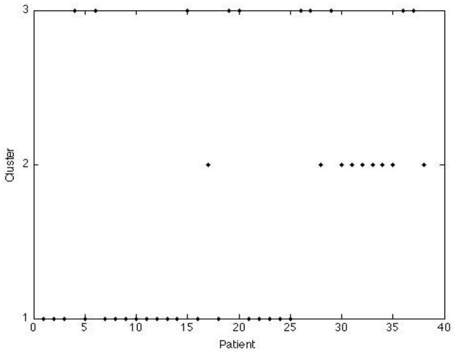
5.1 Description

We received data set from the Broad Institute of MIT and Harvard on gene expression levels of leukemia patients [**?**]. The data set is a 5000 × 38 matrix from microarray that records the expression level of 5000 genes from each of the 38 patients. The data has been arranged in order so that the diagnosis of Leukemia patients are as follows:

Patients 1-19: Acute Lymphoblastic Leukemia B-cell (ALL-B) Patients 20 28: Acute Lymphoblastic Leukemia T-cell (ALL-T) Patients 29 38: Acute Myeloid Leukemia (AML)

5.2 *k*-means

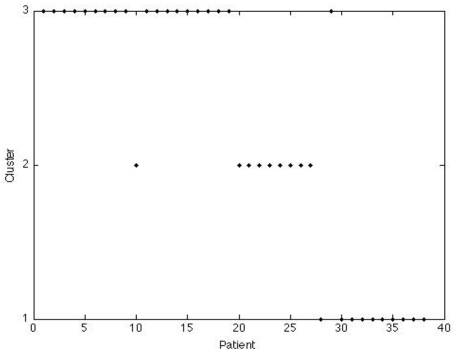
Since there are three types of leukemia, we used $k = 3$ when we ran our k-means algorithm on the data. The following figure gives the clustering result:



The clustering was not good at all - more than a third of the patients were misclassified.

5.3 NMF

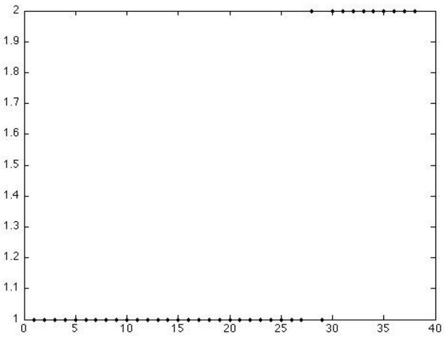
The following figure gives the clustering results from running our NMF algorithm on the data with $k = 3$.



This clustering is a vast improvement on the k-means result. Only two patients, 10 and 29, were misclassified. However, information from the Broad Institute indicates that at least patient 29 was most likely misdiagnosed.

5.4 Reverse Simon-Ando

We constructed a consensus matrix using NMF, with 10 runs each of $k = 2$, $k = 3$, $k = 4$, and $k = 5$. The following is the clustering result from reverse Simon-Ando using that matrix.



This clustering is good - only patient 29 is misclassified, and as discussed above, that was most likely a misdiagnosis. One weakness, though, is that only two clusters were recognized, not three. This suggests once again that purely looking at the largest gap in the list of eigenvalues may not be the best method.

6 Iris

6.1 Description

Another data set used this summer was the Iris data set found in Matlab [**?**]. This data set is a 150 × 4 matrix. It is represented by 150 observations of four different measured variables: the sepal length, sepal width, petal length, and petal width. Three species are represented in the data set: setosa, versicolor and virginica.

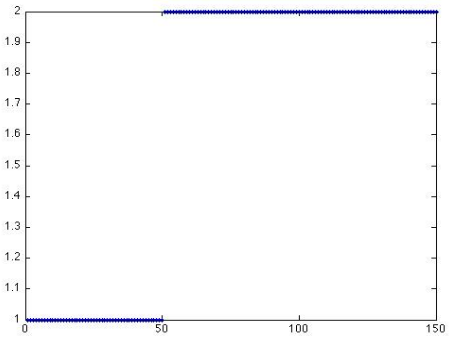
6.2 Analysis

Since the clustering of the Iris data set was known, it was used to test the clustering capabilities of the Reverse Simon-Ando method.

First, we built consensus matrices S with 5, 10, and 50 iterations of the NMF and k-means methods, and applied the Sinkhorn-Knopp algorithm to get doubly stochastic matrices P. Then, the eigenvalues of P were studied to determine the number of clusters. For matrices constructed using NMF, when $k = 2$, $k = 2, 3, 4$, and $k = 2, 3, 4, 5$, the largest gap in the sorted list of eigenvalues of P suggested that there were 2 clusters, whereas for matrices using k-means, it sometimes suggested only 1 cluster . This also indicates that the location of the largest gap in the list of eigenvalues may not be the best way to determine the number of clusters.

It is interesting to note that when the k-means algorithm with $k \leq 5$ was used to build the consensus matrices, the gap in the eigenvalues of P increased. This suggests that k-means may not be the best method to use on the Iris data set.

The following is a clustering result of the reverse Simon-Ando method using a consensus matrix constructed with 10 iterations of NMF, $k = 2, 3, 4, 5$. As discussed above, there are two clusters shown.



7 Grimm

7.1 Description

The final data set used this summer was the Grimm data set [**?**]. This data set was a 705 × 63 term document matrix which represented 705 terms in 63 of the Brothers Grimm fairy tales. After removing the words that only appeared in exactly one of the fairy tales, the data set became a 668 × 63 matrix. This matrix was used to cluster the fairy tales.

7.2 Analysis

Before normalizing the matrix, we attempted to cluster with $k \in [2, 10]$. There appeared to be no consistency with any of the fairy tales clustering together. Then, we used k-means and NMF with $k = 2$. Out of three runs, only two groups of fairy tales (one group of 14 and the other of 8) clustered together consistently.

After these results, the matrix was normalized and more clustering methods were used on the data. Consensus matrices were built with $k = 2$, $k = 2, 3$, and $k = 2, 3, 4$ for 5, 10 and 50 iterations of k-means and NMF. Reverse Simon-Ando was then used on these consensus matrices. In all but four cases, RSA gave only 1 cluster for the Grimm Data set. Of the four cases that gave 2 clusters, only two returned similar clusters and only one produced a second cluster that contained more than 3 elements.

In addition, a few more cases were ran with $k = 5, 6, 7, 8, 9, 10$ and $k =$

$10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20$. Running RSA on these consensus matrices did not provide any new information. We concluded that the Brothers Grimm fairy tales are either not similar and only cluster with themselves, or those that do cluster together are so weakly clustered that they do not appear when applying clustering methods on the data set as a whole.

8 Finding the number of clusters

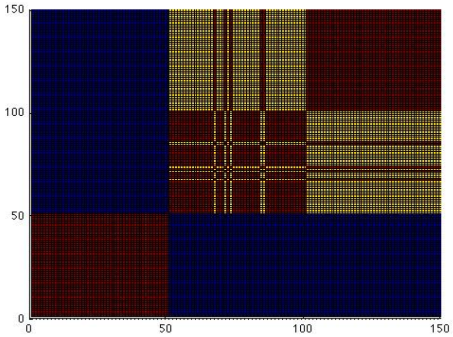
Finding the correct number of clusters k is an important question in clustering. None of the clustering methods mentioned in Section 2 are ever completely accurate every time. By removing some of the randomness, the consensus matrix helped to resolve some of this inaccuracy.

Besides the consensus matrix, we used another method - studying the eigenvalues and eigenvectors of the matrix resulting from using Sinkhorn-Knopp algorithm on the consensus matrix. It was conjectured in the reverse Simon-Ando method that the number of eigenvalues prior to the largest gap in the sorted list eigenvalues would be able to tell how many clusters were in the data set and that the eigenvectors could describe the content of those clusters. While this hypothesis has a mathematical basis, we tested it on the Iris data set and the Leukemia data set in Matlab.

8.1 Iris

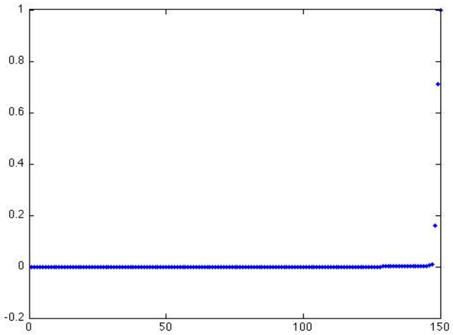
Consensus matrices were built from 50 iterations of NMF and k-means for the Iris data set with $k = 3$, $k = 2, 3, 4$ and $k = 2, 3, 4, 5$.

The following is a heat map of one such consensus matrix C for the Iris data set. We can clearly see three clusters in red with some mis-clustering.

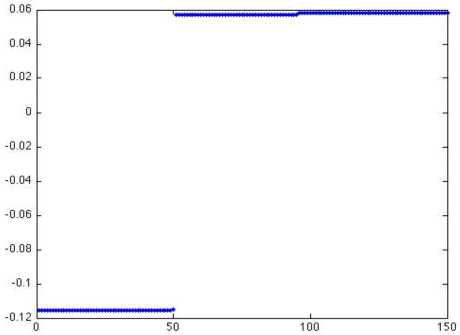


After building the consensus matrices, the Sinkhorn-Knopp algorithm was run on them. Then, the eigenvalues and eigenvectors of these new, doubly stochastic matrices were examined.

The following figure shows the eigenvalues of the doubly stochastic matrix corresponding to C. The location of the largest gap indicates that there are three clusters, as the heat map suggested.

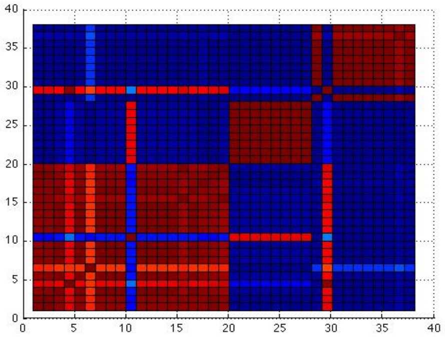


As shown by the following figure, studying the eigenvectors also seems to suggest three clusters. Thus, it appears for the Iris data set that the hypothesis works well in practice.

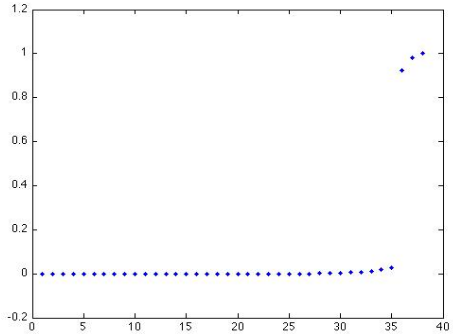


8.2 Leukemia

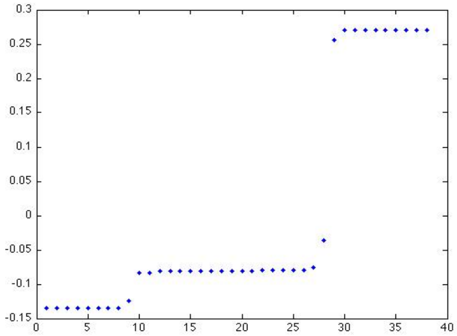
For the Leukemia data set, k-means and NMF were also used to build consensus matrices from 50 iterations with $k = 3$, $k = 2, 3, 4$ and $k = 2, 3, 4, 5$. In the heat map below of one such consensus matrix, we can recognize three clusters with the exception of the mis-clusterings of patients 10 and 29.



Then we ran the Sinkhorn-Knopp algorithm on this matrix; the largest gap in the eigenvalues of the resulting matrix also shows three clusters.

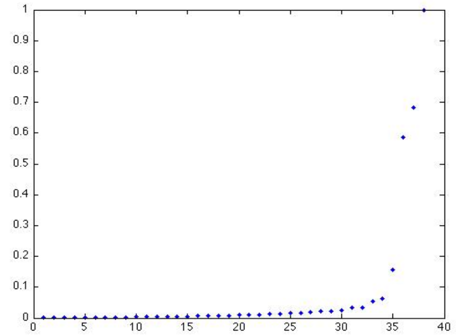


Along with the eigenvalues, the eigenvectors also suggest three clusters.

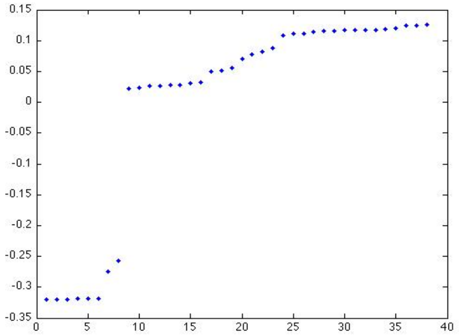


Therefore, our experiments appear to corroborate the hypothesis.

However, the behavior of the eigenvalues consistently agree with the hypothesis, whereas that of the eigenvectors may be more unclear depending on the data. For example, the following eigenvalues for one consensus matrix from the Leukemia data set suggest three clusters.



However when examining the eigenvectors for this same matrix, we see that the clusters are somewhat unclear.



Thus, in most instances, the eigenvalues accurately define the number of clusters. But, the eigenvectors provide a less clear definition of the clusters.

9 Uncoupling Measure

We will now discuss the uncoupling measure of a consensus matrix. In this paper, we use a form of the measure given in [**?**].

Definition 1. *Let S be an $n \times n$ consensus matrix and n_1 and n_2 be positive integers such that $n_1 + n_2 = n$. Suppose that S is in the form*

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

where S_{11} is $n_1 \times n_1$ and S_{22} is $n_2 \times n_2$. The ***uncoupling measure of*** *S*

with respect to n_1 *is the function* $\sigma(S, n_1) = \dfrac{e^T S_{12} e + e^T S_{21} e}{e^T S e} = \dfrac{2e^T S_{12} e}{e^T S e}$. *In other words, the* ***uncoupling measure*** *is the ratio of the sum of the entries in the off-diagonal blocks to the sum of all entries in the matrix.*

This definition can easily be extended to a $k \times k$ block diagonal form of S. Also note that there is a n_1 where $\sigma(S, n_1)$ is minimized.

The uncoupling measure of a consensus matrix has some possible uses and appears to have several interesting properties.

9.1 Applications

Suppose that given a data set, a consensus matrix S is permuted so that the rows and columns corresponding to data points in the same cluster are adjacent. Then, S would be in block-diagonal form, with each diagonal block corresponding to a cluster. In this case, the uncoupling measure can be used to evaluate the effectiveness of the clustering. If the clustering is correct and the clusters are distinct, the uncoupling measure should be close to 0. If instead the measure is closer to 1, then the clustering is not effective; this may also indicate that there are too many clusters.

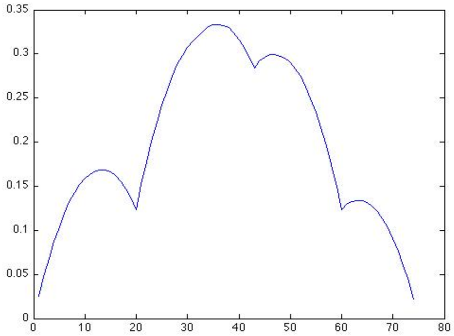
As noted previously, there is a n_1 where $\sigma(S, n_1)$ is minimized - let us call it $\hat{n_1}$. Suppose that we calculate $\hat{n_1}$ for every possible permutation of S. Then, the permutation with the smallest $\hat{n_1}$ gives the best possible partition of the data into two groups; the data corresponding to rows 1 to n_1 is one group and the data corresponding to rows $n_1 + 1$ to n is the other. However, this method is not practical, as the number of permutations of S increases exponentially with n.

9.2 Properties

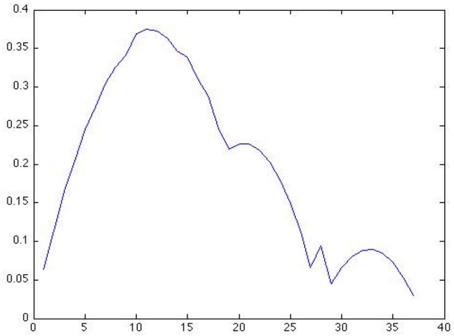
Observation 1. *Suppose that given a data set with k clusters where the cluster assignments are known, we permute a consensus matrix S such that the rows and columns corresponding to data points in the same cluster are adjacent. Then, the graph of $\sigma(S, n_1)$, where n_1 is an integer in $[1, n-1]$, has k areas of concavity, or equivalently, $k-1$ local minima.*

To illustrate the point, the following are graphs of $\sigma(S, n_1)$ for the Ruspini, leukemia, and Iris data sets.

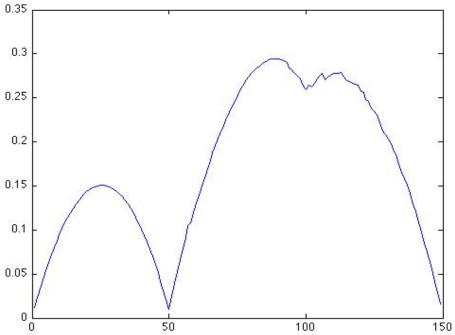
The consensus matrix for the Ruspini data set was constructed from 10 runs each of our k-means algorithm with $k = 2$, $k = 3$, $k = 4$, $k = 5$, and $k = 6$.



The consensus matrix for the leukemia data set was constructed from 10 runs each of our NMF algorithm with $k = 2$, $k = 3$, $k = 4$, and $k = 5$.



The consensus matrix for the Iris data set was constructed from 10 runs each of our k-means algorithm with $k = 2$, $k = 3$, $k = 4$, and $k = 5$.



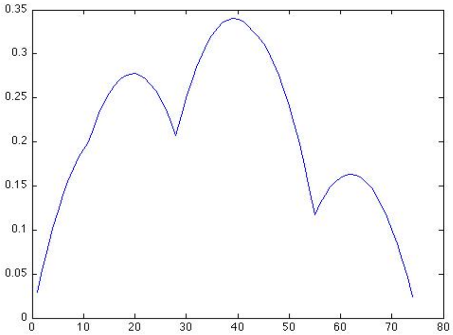
However, this nice structure degrades if the rows and columns of S are not in the correct order.

Observation 2. *Consider the situation described in the previous observation. If we apply the Sinkhorn-Knopp algorithm to S to get P, then the graphs of $\sigma(S, n_1)$ and $\sigma(P, n_1)$ have the same shape. That is, the graph of $\sigma(P, n_1)$ has $k-1$ local minima as well. Only the numbers may change.*

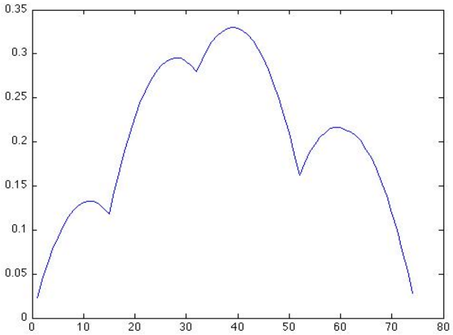
This suggests that the block-diagonal structure of a consensus matrix is not affected by the Sinkhorn-Knopp algorithm.

Observation 3. *Suppose that given a data set and a clustering with k clusters, we permute a consensus matrix S such that the rows and columns corresponding to data points in the same cluster are adjacent. If k is greater than the actual number of clusters in the data set, then the graph of $\sigma(S, n_1)$ has less than k areas of concavity, or less than $k-1$ local minima.*

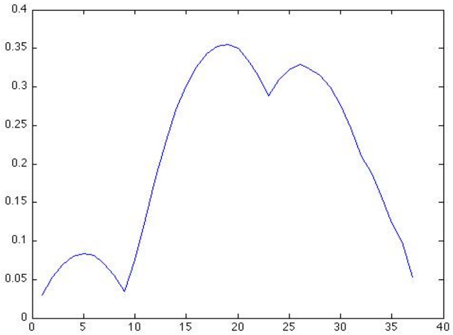
To illustrate this point, we provide two figures. The first is for the Ruspini data set, which has four clusters. The consensus matrix S was constructed from 10 runs each of our k-means algorithm with $k = 2$, $k = 3$, $k = 4$, $k = 5$, and $k = 6$. It was then permuted using a clustering from one run of our k-means algorithm with $k = 5$. Notice that there are three areas of concavity, instead of five.



But, if we permute using a clustering from one run of our k-means algorithm with $k = 4$, there are instead four ares of concavity, which is exactly what we expect.



The second is for the leukemia data set, which has three clusters. The consensus matrix S was constructed from 10 runs each of our NMF algorithm with $k = 2$, $k = 3$, $k = 4$, and $k = 5$. It was then permuted using a clustering from one run of our NMF algorithm with $k = 4$. Note that there are three areas of concavity, instead of four.



This suggests that by examining the graph of $\sigma(S, n_1)$, when S has been permuted based on a clustering, we may discover whether that clustering used too many clusters. Thus, this has the potential to be a tool for clustering, but more research is needed to determine whether this observation is true in general.

10 Conclusion

Because data is so overwhelming in volume and data in higher dimensions is hard to visualize, it is necessary to rely on computers for clustering. However, computers have their shortcomings because they follow a specific rigid algorithm - they lack the human flexibility and intuition. This means peculiarities in data sets may cause an algorithm to work well on one data set and not on another. For example, nonnegative matrix factorization classifies leukemia patients well on the Broad Institute data set, but poorly clusters points in the Ruspini and Iris data sets. On the other hand, K-means works well on the Ruspini data set but not on the leukemia data set. Therefore, it is best to use a variety of methods.

Another common problem in many of the clustering algorithms is that of initialization. Both k-means and NMF use random initializations and both need the number of clusters k to be specified when it is unknown. We have thus aimed to devise a new clustering method that is not heavily dependent on initialization and does not require the user to specify k. We used the ideas of the Simon-Ando theory and reversed the process to determine k. By forcing the consensus matrix to be doubly stochastic, we fixed the future and enabled the study of the past. We also used the properties of eigenvalues to let the computer tell us k, rather than guessing ourselves. While the algorithm is not perfect, experiments on several data sets yield promising results.

11 Acknowledgements

We would like to thank our advisor, Dr. Carl Meyer and our graduate student mentor Ralph Abbey for guiding us through the research process. We would also like to thank North Carolina State University for hosting this research program and the National Science Foundation and National Security Agency for providing funding that made it possible.

NSF grant numbers: DMS-0636568, DMS-0552571, DMS-1063010 NSA grant number: H9823-10-1-0252

References

[1] Daniel D. Lee and H Sebastian Seung, *Algorithms for Non-negative Factorization* **Adv. Neural Info. Proc. Syst. Vol. 13, p. 556-562** 2001

[2] E. H. Ruspini, *Numerical methods for fuzzy clustering.* **Information Science, Vol.2, p. 319-350**, 1970.

[3] The Broad Institute of Harvard and MIT, http://www.broadinstitute.org/mpr/publications/projects/NMF

[4] *Statistics Toolbox* **MATLAB R2011a Documentation** http://www.mathworks.com/help/toolbox/stats/bqzdnrv-1.html

[5] Jacob and Wilhelm Grimm, *Grimm's Fairy Tales* **Project Gutenberg Online** http://www.gutenberg.org/ebooks/2591

[6] Carl D. Meyer and Charles D. Wessell, *Stochastic Data Clustering*, **SIMAX**, under revision, 2010.