11-27-2018

Comparing and Contrasting Clustering Analysis Methods: K-means and Vector in Partition

Lauren Sobral

COMPARING AND CONTRASTING CLUSTERING ANALYSIS METHODS: K-MEANS AND VECTOR IN PARTITION

by

Lauren Sobral

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Mathematical Science

The University of Memphis

December 2018

Abstract

This paper delves into the similarities and differences between two methods of exploratory cluster analysis, K-means and Vector in Partition. Known as the traditional clustering approach, K-means does have some limitations when dealing with clustering complex datasets, specifically datasets with variables of multidimensional vectors. This is the gap the Vector in Partition (VIP) algorithm aims to fill. As a novel approach for clustering multidimensional datasets of both continuous and categorical data, the VIP algorithm has preliminary results that support its ability to correctly cluster simulated datasets of the genetic factors, gene expression, DNA methylation, and single nucleotide polymorphisms. After explaining both the K-means algorithm and the VIP algorithm, an example will be presented of simulated genetic data containing variables with multidimensional vectors that will be analyzed with both algorithms. The results will then be summarized using accuracy, sensitivity, and specificity while highlighting the benefits and limitations of each clustering method.

Table of Contents

List of Figures

1 Introduction

This graduate thesis will discuss two competing exploratory clustering analysis methods. Exploratory clustering analysis aims to find patterns in data without any bias from known pre-conditions. This type of analysis is applicable across several different fields of innovation and research such as marketing to a certain demographic or bio-statistical research of genetics. The two clustering algorithms of focus are K-means and Vector in Partition (VIP). These algorithms construct partitions of data by evaluating each observation with some criteria unique to each method.

Clustering analysis is a statistical classification technique in the Pattern Recognition field. The overarching goal of this field and its techniques is to clarify the recognition of patterns in the decision-making process and to automate these functions using a computer [2]. When clustering observations in a dataset, observations are similar to the observations within the same cluster and dissimilar to the observations in other clusters. For this paper, the data will be of unsupervised classification, meaning there are no predefined clusters. The goal of clustering analysis is to explore and analyze data to use in further research or to gain knowledge of relationships within the dataset. Some clustering analysis methods include hierarchical clustering which forms a set of nested clusters that are organized as a tree, and partitioning clustering which divides a set of data into non-overlapping subsets. K-means and VIP are methods of partitioning clustering. Ideally,

a "good " cluster analysis will have low variance within clusters and high variance between clusters. The following clustering methods are often used in a computer setting in order to analyze large datasets.

Published in 1967, James MacQueen is given credit for the traditional K-means algorithm which is a "process for partitioning an n-dimensional population into k sets on the basis of a sample"[4]. Since then, K-means has been adapted and extended to fit diverse datasets, including options for categorical data with K-modes and mixed data with K-prototypes (K-proto). In 1979, J.A. Hartigan and M. A. Wong published "A K-Means Clustering Algorithm" where they presented a more efficient version of the K-means algorithm. The goal is a K-means algorithm that "divides m points in n dimensions into k clusters so that the within-cluster sum of squares is minimized"[1]. Specifically, Hartigan and Wong allow for a range of k clusters to be analyzed. This version of K-means is the model used in the R package kmeans. These adaptations to K-means have met needs across multiple dataset scenarios; however, K-means is limited to each vector in the n-dimensions to of the same size dimension.

Attempting to fill this gap in clustering complex datasets of both categorical and continuous data, Dr. Meredith Ray created the Vector in Partition algorithm which has the ability to cluster a dataset in which each observation within a subject consists of vectors of varying lengths and data type. The idea of clustering data into clusters by subjects stands, but the VIP algorithm allows for more flexibility in multidimensional datasets with

several moving parts. This algorithm was specifically designed with genetic research in mind. The VIP algorithm is still in testing and research phases which aim to partition datasets with an interest in finding patterns in genetic data related to the medical condition eczema.

2 Statement of Research Objectives

The algorithms for K-means and Vector in Partition will first be defined and explained, delving further into the similarities and differences of the methods. Then, in order to demonstrate how each algorithm works, an analysis of simulated data using both K-means and VIP will be presented. Constructing a formal clustering analysis will show the benefits and limitations presented in each statistical method. The computer program R was used for the simulation of the data and the clustering analysis of K-means and VIP.

3 K-means

The idea behind the K-means clustering method is to partition a dataset of m observations and n dimensions into k clusters where the total within sum of squares is minimized. The cluster centers are the mean of the observations in said cluster. In the K-means algorithm, the cluster centers and number of clusters is not predefined. Figure 1 below demonstrates a simple example explaining the idea behind the algorithm [3].

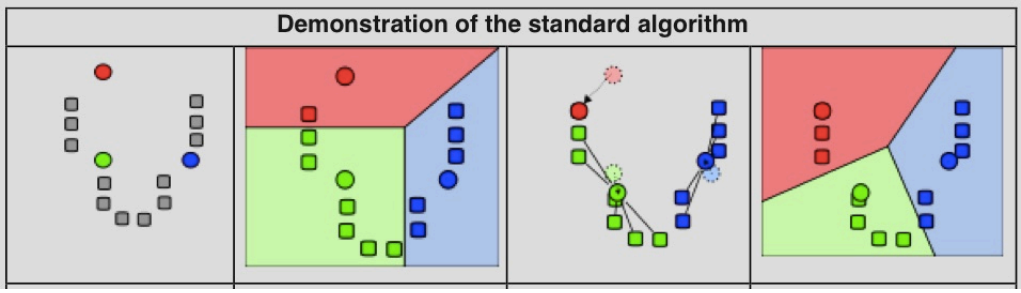

Figure 1: K-means Illustration

In the figure, there are $m = 12$ observations, $n = 1$ dimensions and $k = 3$ clusters. The squares are the observations and the circles are the cluster centers. Each cluster is a partition of the graph. The initial step in K-means is the random selection of cluster centers. Each observation is then assigned to the cluster center closest to that observation based on the Euclidean distance. The cluster center is updated with the current clustering assignment, and then the observations are reassigned to the closest center. This process continues until there is no movement between the clusters.

Algorithm: K-means

The setup for the K-means algorithm is as follows.

Suppose there is a matrix, $\mathbf{X}$, of m observations and n dimensions. For demonstrative purposes, suppose there is only one dimension in our matrix. Start with some number of clusters, say k, making the set of clusters denoted as $j = \{1, 2, ..., k\}$, and $k \leq m$. Then the dataset of observations, $\mathbf{X}$, is randomly divided into clusters. Find the mean of each cluster to create

the cluster centers, denoted as $\{c_1, c_2, ..., c_j\}$. For the clustering assignment, find the Euclidean distance, $E(x_i, c_j)$, between each observation and every cluster center. The Euclidean distance is defined as:

$$E(x_i, c_j) = \sqrt{(x_i - c_j)^2} \tag{1}$$

Here, x_i is observation i, where $i = \{1, ..., m\}$ and c_j is the cluster center for cluster j, where $j = \{1, ..., k\}$. This distance is then used to redefine the clusters based on the criteria that the observations are assigned to the cluster with the smallest distance. These steps are then repeated until the clusters and cluster centers converge.

The total within sum of squares is minimized to choose the optimal k number of clusters. The K-means algorithm defines the total within sum of squares as:

$$WSS = \sum_{l=1}^{j} \sum_{h=1}^{m_j} (x_{lh} - c_l)^2 \tag{2}$$

Here, m_j denotes the number of subjects in each cluster j, $j = \{1...k\}$.

Then this WSS is penalized based on a pseudo Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC). In the formal definitions of AIC and BIC, the log likelihood is replaced with the total within sum of squares plus the penalty for complex clusters. The pseudo AIC has a penalty of 2 times the number of clusters, and the pseudo BIC has a penalty of using $log(n)$ times the number of clusters, where n is the number of subjects. The cluster with the smallest AIC or BIC is chosen as the

optimal number of clusters for the dataset.

Benefits and Limitations: K-means

Some benefits of the K-means algorithm include its efficiency and adaptability to types of datasets. The K-means algorithm and its extensions have quick and efficient computational time. While K-means is used for continuous datasets, its extensions K-modes and K-proto account for categorical datasets. Following the same general set up as K-means, K-modes finds the modes of the categorical datasets for the cluster centers, and K-proto uses both the means and modes for mixed datasets. Hartigan and Wong's K-means algorithm in the R package contains an option called n-start. This allows for n number of random starts for the initial cluster center assignments which reduces some error in each run of the algorithm. However, even with these extensions and adaptations to the K-means algorithm, there are still some limitations. Because K-means is only looking at a set number of clusters defined by the user's initial cluster guess and n-starts, the true optimal number of clusters can not be found, only the local optimal number of clusters. K-means is also restricted to equal sizes or dimensions of its data where each vector in the n-dimensions have be of the same size dimension. For example, K-means can cluster a dataset of 3 variables as long as each variable has 1 dimension and is of equal length.

4 Vector in Partition

The Vector in Partition algorithm aims to bridge the gap in clustering analysis for datasets with both continuous and categorical observations where each vector in the n-dimensions can be multidimensional and of different sizes. The idea behind the VIP algorithm stems from the general K-means concept in that it partitions observations into clusters where the sum of squares from each observation to the assigned cluster center is minimized. However, the VIP algorithm can handle variables that are of multiple dimensions and of various lengths. Dr. Ray's goal for the VIP algorithm is to have developed "a novel non-parametric clustering approach for multidimensional gene-related variables" [5]. In gene analysis and other industries dealing with complex datasets, there are often variables with multiple dimensions that need to be looked at simultaneously. Yet, because there are currently no clustering tools to analyze the joint effects of these factors, they can only be analyzed separately and then compared after.

The VIP algorithm was originally modeled for multidimensional gene analysis in relation to identifying patterns across the variables, gene expression (GE), single nucleotide polymorphisms (SNP) and DNA methylation (DNAm or CPG), in association with the allergic skin disease, eczema. The factors, SNP and CPG, can cause genetic variation which can affect how the gene is expressed. Thus, these genetic factors are dependent on each other when determining the genetic makeup of a person.

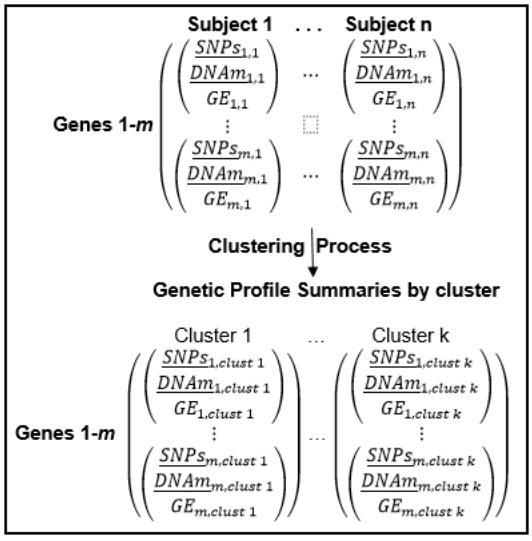

Figure 2: Illustration of the clustering of the genetic factors as vectors for each subject and each gene within a dataset

Figure 2 shows an illustration of how the VIP algorithm clusters data with different sized multidimensional vectors [5]. The top image is of a dataset of genetic factors. For each gene with a subject there exists an SNP vector, a DNA methylation vector (DNAm or CPG), and one GE. The line under each label represents a vector, and the vectors can be of different lengths. Through the VIP clustering process, the data is clustered by subject across all genes. The VIP algorithm looks at the joint effect of these genetic factors by using a new distance measure that will take into account each dimension of each variable equally and simultaneously.

Algorithm: VIP

The setup for the VIP algorithm is as follows.

Suppose there is some observed genetic data in a matrix, $\mathbf{X}$, of size n x m, where n is number of subjects and m is the number of genes. Let $\mathbf{x}_{i,j}$ represents the ith subject and the jth gene in dataset $\mathbf{X}$, where $i = \{1, ..., n\}$ and $j = \{1, ..., m\}$. Each $\mathbf{x}_{i,j}$ is a collection of three vectors of data for SNP, CPG, and GE of each subject.

$$\mathbf{x}_{i,j} = \{\mathbf{x}^c_{j,i,1}, \mathbf{x}^r_{j,i,2}, \mathbf{x}^r_{j,i,3}\} \tag{3}$$

Here $\mathbf{x}^c_{j,i,1}$ represents the categorical vector for SNP; $\mathbf{x}^r_{j,i,2}$ represents the continuous vector for CPG, and $\mathbf{x}^r_{j,i,3}$ represents the continuous vector for GE. Note, the superscript c represents categorical data and r represents continuous data. The goal is to cluster n subjects into k clusters.

Now that the observed data is defined, the number of clusters, the initial cluster centers, the distance measure, and the center updating method need to be defined. For the number of clusters, the range of possible clusters to be evaluated is $s = \{1, ..., k\}$, where k is user defined and $k \leq n$. The initial cluster centers are based on a random cluster assignment for each subject. Let $\mathbf{q}_s$ be the cluster centers. These are the means of the continuous observations and the modes of the categorical observations.

$$\mathbf{q}_s = \{\mathbf{q}^c_{s,1}, \mathbf{q}^r_{s,2}, \mathbf{q}^r_{s,3}\} \tag{4}$$

Here $\mathbf{q}^c_{s,1}$ is a vector that represents the SNP cluster centers, $\mathbf{q}^r_{s,2}$ is a vector that represents the CPG cluster centers, and $\mathbf{q}^r_{s,3}$ is a vector that represents the GE cluster centers for cluster s across all m genes.

These cluster centers, $\mathbf{q}_s$, and the observed data, $\mathbf{x}_{i,j}$, will be used in the distance measure. A redefined distance measure will be used to assign observations to clusters with the smallest distance. This distance measure can handle both continuous and categorical data types of multidimensional vectors so that each is expressed equally and simultaneously to get the joint effects. Let $d(x_{i,j}, q_s)$ denote the distance measure.

$$d(\mathbf{x}_{i,j}, \mathbf{q}_s) = \sum_{j=1}^{m} Y_j * [E(\mathbf{x}^r_{j,i,2}, \mathbf{q}^r_{j,s,2}) + E(\mathbf{x}^r_{j,i,3}, \mathbf{q}^r_{j,s,3})] \tag{5}$$

Where

$$Y_j = \frac{1}{\sum_{p=1}^{l} P(\mathbf{x}^c_{p,i,1}, \mathbf{q}^c_{p,i,1})} \tag{6}$$

Here, l is the number of SNPs in $\mathbf{x}^c_{j,i,1}$ vector, and let

$$P(\mathbf{x}^c_{p,i,1}, \mathbf{q}^c_{p,i,1}) \begin{cases} 1 & \mathbf{x}^c_{p,i,1} = \mathbf{q}^c_{p,i,1} \\ 0 & \mathbf{x}^c_{p,i,1} \neq \mathbf{q}^c_{p,i,1} \end{cases} \tag{7}$$

The distance measure is the sum across all genes of the Euclidean distance, $E()$, of the observed data and cluster centers for the continuous genetic factors, CPG and GE. This is proportionally weighted by Y which accounts for the number of matching observations of the observed SNP

data and cluster centers. The distance measure takes into account the similarities between CPG and GE while weighting it with the matching SNPs across all genes. For example, suppose there is some gene j and there are 2 SNPs in the vector $\mathbf{x}^c_{j,i,1}$, so $l = 2$. If both the observed SNPs match the cluster center SNP, then Y=1/2 and the similarity between CPG and GE has less weight. If CPG and GE are similar to the cluster centers patterns, then the Euclidean distance will also be small. Thus, if the SNP, CPG, and GE are all similar to the cluster centers, then the distance measure will subsequently be small.

Based on this distance measure summed across all genes, each subject is then assigned to the cluster in which it is closest. Similar to K-means, the centers are then updated, and the distance is measured again until there is no more movement between clusters.

The VIP algorithm chooses the optimal number of clusters from a user-specified range based on minimizing the variation within clusters and penalizing complex clusters. The information criteria used for calculating this variation is an objective measure that takes the total variation along with a penalty that avoids overfitting. Similar to K-means, a pseudo AIC and BIC are used. The pseudo AIC has a penalty of 2 times the number of clusters times the degrees of freedom from the observed data for SNP, CPG, and GE, denoted by S_{df}, C_{df}, G_{df}. The pseudo BIC has a similar penalty with the exception of using $log(n)$, where n is the number of sub-

jects. The pseudo AIC and BIC is defined as:

$$AIC = WSS + (2 * s * (S_{df} + C_{df} + G_{df})) \tag{8}$$

$$BIC = WSS + (log(n) * s * (S_{df} + C_{df} + G_{df})) \tag{9}$$

Here, both equations use the total within sum of squares (WSS). This is the objective measure. The VIP algorithm defines WSS as the distance measure, $d(x_{i,j}, q_s)$, summed across the subjects in each cluster, n_j, and summed across the clusters $s = \{1...k\}$.

$$WSS = \sum_{s=1}^{k} \sum_{i=1}^{n_s} d(x_{i,j}, q_s) \tag{10}$$

The cluster with the smallest AIC or BIC is chosen as the optimal number of clusters for the dataset.

Benefits and Limitations: VIP

While still in the research and testing phases, the VIP algorithm and R package are still being improved for efficiency and simplicity. The preliminary testing has shown that the VIP algorithm correctly clusters data where each variable is a multidimensional vector. The main benefit of the VIP algorithm is the new distance measure that "outperforms existing distance definitions in that it has the ability to assess joint effect of factors in each variable vector composed of different types of variables"[5]. This allows for the analysis of multidimensional variable vectors to get the

joint effects in the data. Some limitations of the VIP algorithm include the specialization of the algorithm, and the possible effect of the size of the datasets on the output. Currently the VIP algorithm is very specialized in its research and testing which is geared towards analyzing GE, SNP, and CPG in relation to genes and eczema status. However, the flexibility of the algorithm can easily be changed to fit several different multidimensional scenarios. The size of the datasets, more specifically the number of subjects and genes, may affect the output for clustering the subjects correctly, but while the algorithm is still in the testing phases of research, no known patterns or relations have been confirmed. Also, the bigger the dataset and the larger the range to look for the optimal cluster, the longer the computational time.

5 Example

Now, a clustering analysis will be presented on a simulated dataset example where the dataset will be analyzed with both the K-means algorithm and the VIP algorithm. In an effort to show the advantages of the VIP algorithm, the dataset will be of multidimensional variable vectors related to gene analysis. One-hundred datasets of the same scenario will run through each algorithm, and the output will be summarized to compare the results. The simulations will be summarized on correct clustering with accuracy, and sensitivity and specificity per cluster.

To summarize the output, the final clustering assignments for all 100

datasets will be recorded and labeled as true positive, true negative, false positive, false negative. True positive means the subject belongs in the cluster and they were assigned to that cluster. True negative means the subject does not belong in the cluster and they were not assigned to that cluster. False positive means the subject does not belong in the cluster and they were assigned to that cluster. False negative means the subject belongs in the cluster and they were not assigned to that cluster. Then using those labels, the accuracy of clustering across all clusters and the sensitivity and specificity of clustering in each cluster will be calculated. Accuracy shows the proportion of subjects clustered correctly. It is the true positives and true negative, divided by the total number of subjects. The sensitivity of each cluster is the number of true positives divided by the number of true positives and false negatives. This shows the probability that the subject is put in a cluster given they belong in that cluster. The specificity is the number of true negatives divided by the number of true negatives and false positives. This is the probability that a subject is not put in the wrong cluster. For each category in the results, the goal is to have a clustering median of 3 and the percentages of accuracy, sensitivity, and specificity to be high with a low standard deviation.

Simulating the Data

The data will be simulated using R. Suppose there is some genetic data in a matrix, X, with 100 subjects, n, and 100 genes, m. Let $\mathbf{x}_{i,j}$ rep-

resent the ith subject and the jth gene in the dataset X, where $i = \{1, ..., n\}$ and $j = \{1...m\}$. Each $\mathbf{x}_{i,j}$ is a collection of vectors containing data from each subject's SNP, CPG, and GE. The data will be simulated with the true number of clusters being 3.

		Gene 1 - 33			Gene 34 - 50			Gene 51 - 66			Gene 66 - 100		
	Subjects	Distibution		length of dimension	Distibution		length of dimension	Distibution		length of dimension	Distibution		length of dimension
Cluster 1	1 - 33	SNP :	M(0.33,0.33,0.34)	3	SNP :	M(0.33,0.33,0.34)	3	SNP :	M(0.33,0.33,0.34)	5	SNP :	M(0.33,0.33,0.34)	5
		CPG:	N(4, 0.5)	5	CPG:	N(4, 0.5)	10	CPG:	N(4, 0.5)	10	CPG:	N(4, 0.5)	15
		GE:	N(5, 0.5)	1	GE:	N(5, 0.5)	1	GE:	N(5, 0.5)	1	GE:	N(5, 0.5)	1
Cluster 2	34 - 66	SNP :	M(0.4,0.25,0.35)	3	SNP :	M(0.4,0.25,0.35)	3	SNP :	M(0.4,0.25,0.35)	5	SNP :	M(0.4,0.25,0.35)	5
		CPG:	N(8, 0.5)	5	CPG:	N(8, 0.5)	10	CPG:	N(8, 0.5)	10	CPG:	N(8, 0.5)	15
		GE:	N(10, 0.5)	1	GE:	N(10, 0.5)	1	GE:	N(10, 0.5)	1	GE:	N(10, 0.5)	1
Cluster 3	67 - 100	SNP :	M(0.35,0.45,0.20)	3	SNP :	M(0.35,0.45,0.20)	3	SNP :	M(0.35,0.45,0.20)	5	SNP :	M(0.35,0.45,0.20)	5
		CPG:	N(12, 0.5)	5	CPG:	N(12, 0.5)	10	CPG:	N(12, 0.5)	10	CPG:	N(12, 0.5)	15
		GE:	N(15, 0.5)	1	GE:	N(15, 0.5)	1	GE:	N(15, 0.5)	1	GE:	N(15, 0.5)	1

Figure 3: Distribution of Simulated Data. M stands for multinomial distribution and N stands for Normal distribution.

In Figure 3, there are 3 clusters, 100 subjects and 100 genes for each subject. Vectors for SNP, CPG, and GE are sampled from unique distributions for the subjects in each cluster. SNPs are sampled from a multinomial distribution, and CPGs and GE are sampled from normal distributions where CPG is sampled from a range of values $(-\infty, \infty)$ and GE is sampled from a range of values $(0, \infty)$. The subjects in each cluster have data sampled from different proportions for the categorical SNP data and from different means for the continuous CPG and GE data where the standard deviation is 0.5. The clusters are simulated to represent a gene where there are several SNPs and CPGs per gene. Although not noted, an additional file is needed to denote which CPGs and SNPs are located within

specific genes.

Once datasets have been simulated, the 100 subjects will be clustered using both algorithms.

K-means Example

Start with a cluster analysis using the K-means algorithm. Since the datasets are of varying lengths, the datasets will be run separately using K-means on the continuous datasets for GE and CPG and K-modes on the categorical dataset for SNP. Run the same scenario for datasets $k = \{1...100\}$, and summarize the data by looking at each separate dataset and summarizing based on cluster assignment for each subject finding the accuracy, the sensitivity, and the specificity.

clustering		sensitivity					
		cluster 1		cluster 2		cluster 3	
median	sd	mean	sd	mean	sd	mean	sd
3	0	0.193	0.23	0.5054	0.22	0.6644	0.21
		specificity					
accuracy		cluster 1		cluster 2		cluster 3	
median	sd	mean	sd	mean	sd	mean	sd
0.46	0.11	0.8834	0.14	0.8065	0.09	0.7839	0.1

Figure 4: Summary results of K-modes for SNP dataset

The results from K-modes for the categorical SNP data shows the median cluster assignment is 3. While this is correct, the accuracy of clustering the subjects into the correct number of clusters is only 46%. The sensitivity and the specificity of each cluster is also low compared to the K-means results. Specifically, the sensitivity for cluster 1 is very low at 0.193,

meaning K-modes clustered the subjects who belonged to cluster 1 into cluster 1 only about 19% of the time.

clustering		sensitivity					
		cluster 1		cluster 2		cluster 3	
median	sd	mean	sd	mean	sd	mean	sd
3	0	1	0	0.99	0.1	0.99529	0.05
accuracy		specificity					
		cluster 1		cluster 2		cluster 3	
median	sd	mean	sd	mean	sd	mean	sd
1	0.05	0.995	0.05	1	0	1	0

Figure 5: Summary results of K-means for CPG dataset

clustering		sensitivity					
		cluster 1		cluster 2		cluster 3	
median	sd	mean	sd	mean	sd	mean	sd
3	0	1	0	1	0	1	0
accuracy		specificity					
		cluster 1		cluster 2		cluster 3	
median	sd	mean	sd	mean	sd	mean	sd
1	0	1	0	1	0	1	0

Figure 6: Summary results of K-means for GE dataset

The K-means for CPG and GE have very good results with the correct clustering assignment and high percentages of accuracy, sensitivity, and specificity. Specifically, the K-means algorithm clustered the GE dataset perfectly. While looking at each group separately seems to cluster the subjects well, since the joint effects of the genetic factors is not represented, the results for GE, CPG, and SNP do not give much information on clustering the subjects based on their genes. Typically, in genetic research K-means cluster analysis would not be used to look at the joint effects of GE, CPG, and SNP because the information separately does not give the whole

picture. The computational time for running K-means and K-modes for 100 datasets is about 3 minutes total.

VIP Example

Now run the same datasets in R using the VIP algorithm. Running the same scenario for each k in $k = \{1...100\}$, the inputs are the observed datasets as well as the indices and the names datasets that keep track of which SNPs, CPGs, and GEs belong to which gene for each subject. The algorithm checks clusters in the user defined range, here let the range be $\{2...10\}$, and chooses the optimal cluster based on the pseudo AIC/BIC. The output for each scenario gives a table of the cluster assignment, the mean of each center by gene, the cluster assignment of each subject, the sum of squares within, and the AIC/BIC for the optimal number of clusters chosen. In order to summarize the data, the cluster assignment for each subject will be recorded and the accuracy, the sensitivity, and the specificity will be calculated.

clustering		sensitivity					
		cluster 1		cluster 2		cluster 3	
median	sd	mean	sd	mean	sd	mean	sd
3	0.7	0.9339	0.17	0.9387	0.18	0.9611	0.12
accuracy		specificity					
		cluster 1		cluster 2		cluster 3	
median	sd	mean	sd	mean	sd	mean	sd
0.94	0.08	0.9959	0.09	0.99507	0.05	1	0

Figure 7: Summary results for 100 runs of 3 clusters for data of 100 subjects (n) and 100 genes (m)

The summarized results from the VIP algorithm show the median

cluster assignment is 3. Thus, out of the 100 runs overall, the algorithm clustered the subjects correctly. The accuracy shows the subjects were correctly clustered 93% of the time. The sensitivity, in cluster 1, which is the mean probability that a subject is clustered into cluster 1 given they belong in cluster 1 is 93.15%. For cluster 2, its mean sensitivity is 87.39%, and cluster 3 has a mean sensitivity of 99.47%. The specificity for cluster 1, which is the mean probability that the subjects who do not belong in cluster 1 are not put in cluster 1, is 97.04%. Cluster 2 also has a mean specificity of 97.04%, and cluster 3 has a mean specificity of 98%. The standard deviations for accuracy, sensitivity, and specificity are all low. Overall, these are good results that show the correct number of clusters and have high accuracy, sensitivity, and specificity. The computational time for running the VIP algorithm for 100 datasets was about 6 minutes total.

6 Conclusion

The K-means algorithm and the Vector in Partition algorithm are two exploratory cluster analysis methods for finding patterns in observed datasets without any prior information. While these methods are similar in their setup partitioning observations into clusters by minimizing the within cluster variation, there is difference between the distance measure each algorithm uses to determine the clustering of the observations. The traditional K-means algorithm clusters continuous datasets with its extensions,

K-modes and K-proto clustering categorical and mixed datasets. While the K-means clustering methods are efficient, the distance measure can only handle data where each vector in the n-dimensions is of the same sized dimension, leading to limitations when it comes to more complex datasets. As industries of innovation and research expand with technology and there is more information needing to be processed at high volumes and complexities, the missing sector in clustering analysis is the ability to cluster datasets where observations within a subject consist of vectors of varying lengths and data type. The VIP algorithm fills this gap, allowing for researchers to look at the joint effects of each multidimensional vector in the datasets at the same time. It can handle these complex datasets using its redefined distance measure that accounts for each dimension of each variable equally and simultaneously.

In the example of genetic data with the factors, SNP, CPG, and GE, both K-means and VIP were used to cluster the simulated datasets. Even though K-means and its extensions were able to correctly cluster the datasets separately, there was not much use for the results because the joint effects of the genetic factors were not shown. This highlighted the need for the VIP algorithm and its flexibility to handle the analysis of datasets with several moving parts. The new distance measure takes into account each dimension of the dataset equally and simultaneously. The VIP algorithm is a non-parametric clustering approach which correctly clustered the subjects in the example resulting in a "good" cluster analysis with high percentages

of accuracy, sensitivity, and specificity.

The point of the VIP algorithm is to present a novel approach to mathematically combine these three factors in gene analysis. It accomplishes this by using the new distance measure to cluster mixed datasets where each vector in the n-dimensions can be multidimensional and of different lengths. As analysis and research of the VIP algorithm continues, there will be more information on the limitations including time and size restrictions of genes and subjects, and correct clustering effects in relation to sensitivity and specificity. The next steps for the VIP algorithm is to go into real data testing and use the resulting patterns of clustering to see if there is any relation between the clustered subject's genes and the prevalence of eczema. The VIP algorithm can be easily modified for several scenarios be it more variables, different sizes of vectors, all continuous, all categorical, and any mix in between. This flexibility will allow the VIP algorithm to impact multiple industries in future research.

7 Bibliography

[1] J. A. Hartigan and M. A. Wong. *A K-means Clustering Algorithm.* Yale University, 1979.

https://www.labri.fr/perso/bpinaud/userfiles/downloads/hartigan_1979_kmeans.pdf.

[2] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition, Second Edition.* pg 1-2. Academic Press Inc, 1990.

[3] *K-means clustering.* Retrieved October 2018, from

https://en.wikipedia.org/wiki/K-means_clustering

[4] MacQueen, James. *Some Methods for Classification and Analysis of Multivariate Observations.* Univeristy of California, 1967. https://pdfs.semanticscholar.org/ac8a/b51a86f1a9ae74dd0e4576d1a019f5e654ed.pdf.

[5] Meredith Ray. (MRay_RO3_Research_Strategy_PA-16-162.pdf)