Exploring Clustering with Astronomical Data

An exploration of the effectiveness of clustering algorithms on large astronomical data

April 22, 2015

Authors

NICK HOCKENSMITH
KEVIN PARK
DANE SKINNER
Oregon State University
Corvallis

KHP

2015
KIDDER HALL PRESS

Contents

1 Introduction

The project focuses on clustering roughly 52,000 observations of variable star data broken into a training data set with about 1500 observations and a test data set with about 50,000 observations. There are several approaches to clustering that each have their benefits and drawbacks, and since there is no correct method, nor is there any means of determining which method will ultimately be superior, we employ 5 different clustering methods on the training data and compare the resulting clusters against the known clusters. We primarily focus on the number of observations assigned to each cluster as a means of checking the effectiveness of the clustering algorithm.

2 Clustering Approaches and Results

2.1 Preparing the Data

The training data contains 87 columns that include an index number of the star, several measurements regarding the variable light emitted from the star, statistical calculations based on those measurements, and finally, a single class designation out of 25 possibilities. Problems with the data consist of observations without certain values and columns containing only zeros.

It is necessary to address these problems before performing any clustering, so we first remove the index number and the class designation since one is not a measurement and the other is a non-numerical response value. Further preparation could include removing the statistical data or removing values that do not contribute a significant proportion of the total variation. These would benefit computation time, but ultimately, all the data was left intact because computation time remained small.

Of the roughly 1500 observations in the training data, 1100 remain after removing the observations with NAs present. The 100 observations still span a total of 25 known classes of variable stars, and the total number in each cluster are as follows.

Table 1: Number of Stars of Each Type

Class	1	2	3	4	5	6	7	8	9	10	11	12	13
Obs.	45	39	21	31	51	191	109	13	28	15	13	12	25
Class	14	15	16	17	18	19	20	21	22	23	24	25	
Obs.	55	23	1	25	124	6	6	42	29	11	41	12	

2.2 K-means

With the training data properly prepared, the first algorithm we employ is K-means clustering. We know that there are 25 distinct classes of stars in the training data, so we set $k = 25$. Without scaling the data, clustering according to K-means results in the four largest clusters containing 860 observations: a seemingly clear case of concentrating the data into too few clusters. Upon scaling the data, the data is more balanced to a fault. Now the largest cluster has only 101 observations while several other clusters contain between 30 and 80 observations. In the same way that not scaling the data gave too much weight to few variables, scaling the data gave too much weight to variables that did not originally have much influence on cluster assignment.

Table 2: Cluster Sizes Produced by K-Means with Scaling

Cluster Id	1	2	3	4	5	6	7	8	9	10	11	12	13
Obs.	83	53	28	86	73	28	6	62	42	47	5	48	57
Cluster Id	14	15	16	17	18	19	20	21	22	23	24	25	
Obs.	112	44	82	3	43	30	47	37	41	16	25	2	

2.3 K-medoids

Upon applying this approach to the test data, it becomes clear that the chosen metric influences the resulting clusters and their respective sizes. As shown in the table below, using the Manhattan Metric, otherwise known as the ℓ_1 metric, produces more balanced data while the Euclidean metric (ℓ_2) partitions much of the data into fewer of the clusters. Considering that the Manhattan Metric produces clusters more similar in size to the true data, further exploration on the data using the Manhattan Metric might produce more desirable results.

Table 3: Cluster Sizes Produced by K-Medoids with Manhattan Metric

Cluster Id	1	2	3	4	5	6	7	8	9	10	11	12	13
Obs.	89	27	24	42	122	108	100	78	43	43	50	64	51
Cluster Id	14	15	16	17	18	19	20	21	22	23	24	25	
Obs.	29	102	46	1	19	51	2	1	2	4	1	1	

2.4 K-means ++

K-means++ should provide an improved clustering result over the standard K-means approach since it thoughtfully chooses k cluster centers before proceeding with the standard K-means algorithm. Using scaled data, the five largest clusters produced with the K-means++ algorithm contain 635 observations: very close to the true data where the 5 largest clusters contain 623 observations. The drawback is the large number of single or two observation clusters (6), and the significantly increased computation time.

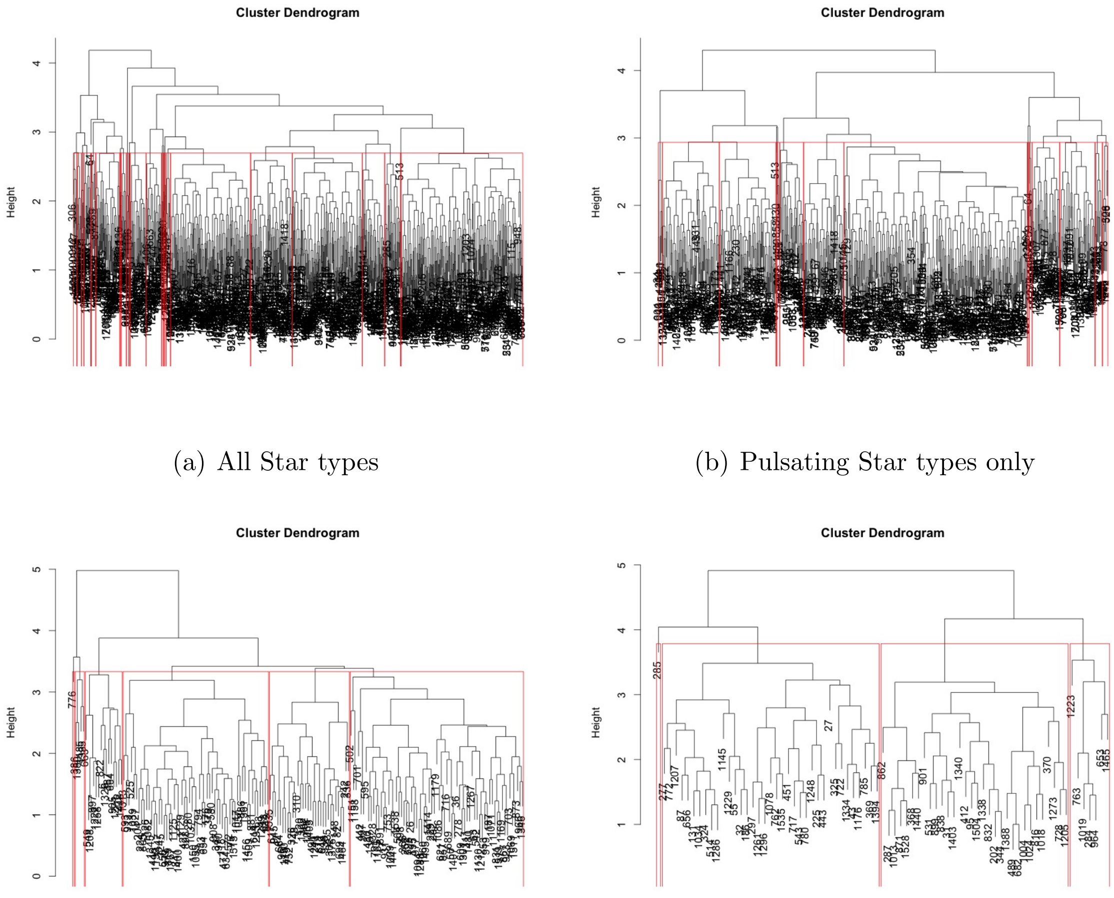
Table 4: Cluster Sizes Produced by K-means++

Cluster Id	1	2	3	4	5	6	7	8	9	10	11	12	13
Obs.	124	1	1	33	170	21	7	2	111	9	105	104	63
Cluster Id	14	15	16	17	18	19	20	21	22	23	24	25	
Obs.	50	14	16	16	20	125	1	6	23	75	1	2	

2.5 Hierarchical Clustering

We consider three hierarchical clustering methods — the “complete,” the “single,” and the “average” linkages. Additionally, we chose to cluster on the entire training data set as well as within the three main subcategories of variable stars. Lastly, all 85 variables are normalized to the interval $[0, 1]$.

After running the normalized data through the “complete,” “single,” and “average” linkages, we focus on analyzing the complete linkage. In essence, the single and average linkages tend to cluster a fairly sizeable amount of observations into a single cluster compared to the complete linkage. This result is further exacerbated if the explanatory variables are either left unchanged or standardized rather than normalized. For example, if the data is standardized and then clustered, cluster #1 contains roughly 850 observation. Furthermore, this number grows to more than a thousand without standardizing. Overall, it would appear that performing the farthest-neighbors clustering on the full set of standardized variables is better able to differentiate the more “tightly” clustered stars. Below is a table for the clusters on the non-subsetted training data as well as graphs for both subsetted and non-subsetted training data sets.



(a) All Star types

(b) Pulsating Star types only

(c) Eruptive Star types only

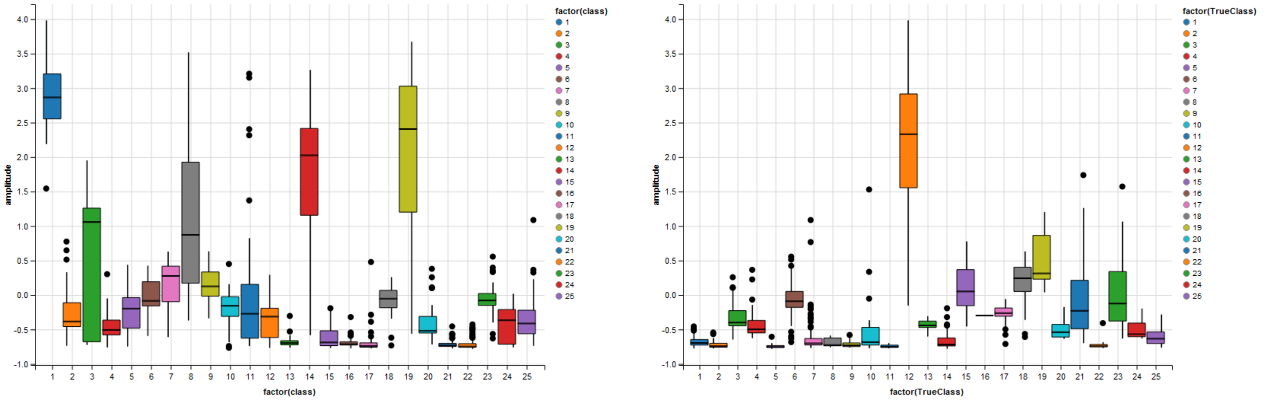
(d) Multi-Star types only

Table 5: Farthest-Neighbor on Non-subsetted Training Data

Cluster Id	1	2	3	4	5	6	7	8	9	10	11	12
Obs.	196	59	11	171	298	102	39	9	38	39	55	1
Cluster Id	13	14	15	16	17	18	19	20	21	22	23	24
Obs.	13	11	17	3	6	11	4	4	1	3	6	3

2.6 Spectral Clustering

Another method we apply is spectral clustering to determine the different clusters. For this method, we did not perform any form of dimensional reduction besides removing the covariates (around 10) that largely contained 0 entries or NAs.



On the left represents the estimated distribution of the clusters and on the right represents the true distribution of stars with respect to the amplitude data. Compared to the other methods, spectral clustering more evenly distributed the sizes between the 25 groups. An important note, the numbers along the x-axis do not represent the same star group as in the true classes. Instead it is the group designated by the spectral clustering.

3 Discussion and Future Work

Clustering the data proved a difficult task due to the number of variables and to the way different star classes showed variable values over similar ranges. Even with each method considered, we would be cautious about applying these methods to actual data for fear of excessive misclassification of observations.

Future work would include further refinements to the algorithms with the goal of reducing misclassification as much as possible. Furthermore, exploring larger data sets would likely require a reduction in the total number of variables; thus we would likely need to perform Principle Component Analysis to reduce our data to a manageable size without eliminating the components necessary for effective clustering.