# Re-clustering of Constellations through Machine Learning

Shanshan Xu,[1] Kaifeng Chen,[2] and Yao Zhou[3]
[1] *Stanford University, Department of Physics*
[2] *Stanford University, Department of Applied Physics*
[3] *Stanford University, Department of Materials Science and Engineering*
(Dated: December 12, 2014)

## INTRODUCTION

Since thousands of years ago, people around the world have been looking up into the sky, trying to find patterns of visible stars' distribution, and dividing them into different groups called constellations. Originally, constellations are recognized and organized by people's imaginations based on the shapes of the star distribution. The most two famous groups of stars is the "Big Dipper" and the "Orion". In modern astronomy, the International Astronomical Union (IAU) has defined constellations as specific areas of the celestial sphere. These areas have their origins in star patterns from which the constellations take their names. In total, there are 88 officially recognized constellations.

On the other hand, certain stars are grouped together primarily because they are close to each other and far away from other stars. In other word, one can approximate constellations as the clusters of stars on the celestial sphere. Then it would be quite interesting to see what would constellations (clusters) look like if one uses some totally objective clustering methods regardless of traditions and human imaginations. For example, would the seven stars in the famous constellation "Big Dipper" still be classified into the same cluster? This gives us an inspiration of re-clustering of constellations using *unsupervised* machine learning techniques.

## DATA

In this project, we used data from BSC5P database table, which contains data derived from the Bright Star Catalog, 5th Edition, preliminary. It is widely used as a source of basic astronomical and astrophysical data for stars brighter than magnitude 6.5, roughly every star visible to the naked eye from Earth. The database contains the identifications of included 9110 stars in several other widely-used catalogs, double- and multiple-star identifications, indication of variability and variable-star identifiers, equatorial positions for B1900.0 and J2000.0, galactic coordinates, UBVRI photoelectric photometric data when they exist, spectral types on the Morgan-Keenan (MK) classification system, proper motions (J2000.0), parallax, radial- and rotational-velocity data, and multiple-star information (number of components, separation, and magnitude differences) for known non-single stars. For our purpose, we extracted star names, positions in the form of galactic coordinate and magnitudes information from this database.

The feature is star's (galactic) coordinate on the celestial sphere. In addition, we set a threshold apparent magnitude $M_c$ and only analyze the stars with magnitude $\leq M_c$. (A brighter star has smaller magnitude.) In the following, we choose $M_c = 2.6$ and 4.6 respectively. For $M_c = 2.6$, there are101 stars which belong to 41 real constellations. For $M_c = 4.6$, there are 973 stars classified into 86 real constellations. The formal serves the purpose to gain a quick result of the clustering algorithm while the latter is the approximation of the stars in the sky.

## CLUSTERING METHODS

### Spherical K-means++

First, we run the spherical K-means algorithm for constellation re-clustering. Standard K-means uses the Euclidean distance as the measure. However, in our problem, since all the stars are on the celestial sphere, sphere distance seems to be a better measure than ordinary Euclidean distance. The spherical K-means [1] algorithm, which was initially developed for clustering large document collections, fits our problem perfectly. In this algorithm, the distance or dissimilarity measure is

$$d(x^{(i)}, x^{(j)}) = 1 - \cos\langle x^{(i)}, x^{(j)}\rangle \,, \tag{1}$$

where $x^{(i)}$ is the unit vector indicating the star $i$'s coordinate on the celestial sphere. $\langle x^{(i)}, x^{(j)}\rangle$ is the angle between two unit vectors $x^{(i)}$ and $x^{(j)}$. The measure (1) is also the distance or dissimilarity used in our other clustering algorithms for constellation classification. In company with this measure, the centroid coordinate $\mu_i$ associated with the cluster $\mathcal{C}_i$ is then updated by

$$\mu_i = \frac{\sum_{j:x^{(j)}\in\mathcal{C}_i} x^{(j)}}{\left\|\sum_{j:x^{(j)}\in\mathcal{C}_i} x^{(j)}\right\|_2} \,. \tag{2}$$

For our purpose, we let the algorithm specify $K$ automatically. Ref. [2] proposed a function $f(K)$ to select K and compared its performance to other methods of $K$ selection including AIC, BIC and the gap statistic [3].

Here, we use $f(K)$ defined in [2] as the criterion but replace the Euclidean distance by the sphere measure (1). The other improvement we made is about the initialization of $K$ centroids. Rather than the uniformly random sampling, we adopted the seeding method in K-means++ [4].

We apply the spherical K-means++ algorithm to the 101 brightest stars when $M_c = 2.6$. We iterate $K$ from 1 to 20 and compute the corresponding $f(K)$, which is summarized in Fig. 1.

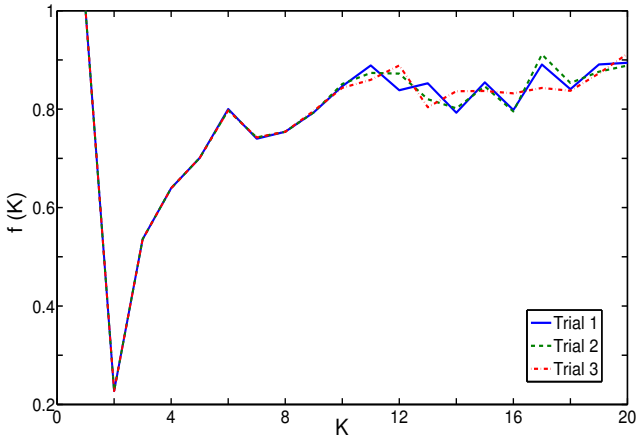


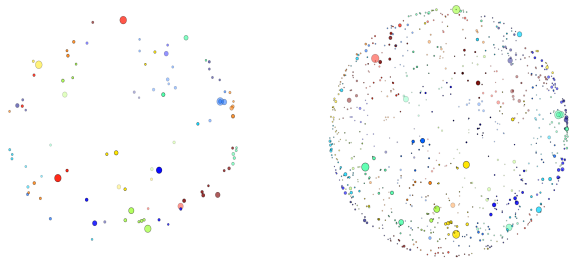

FIG. 1: $f(K)$ values obtained by spherical K-means++. Only stars with magnitude $\leq 2.6$ are considered.

• According to $f(K)$, the optimal $K$ is 2 because $f(2)$ is significantly smaller than other values. This suggests that the brightest 101 stars on the sphere actually form two clusters. However, $K = 2$ is obviously too small for the purpose of re-clustering constellations.

• In a more "reasonable" range like $K \in [8, 20]$, $f(K)$ does not change significantly. This could be the sign that constellations may not own clear clustering structure on the celestial sphere.

• The number of real official constellations for these stars is $K = 41$. As shown in Fig. 2, if we choose $K = 41$ and $K = 86$ corresponding to the real values when $M_c = 2.6$ and $M_c = 4.6$, respectively, the stars in the "Big Dipper" constellation are split into two clusters. Compared with other constellations, the seven stars in the "Big Dipper" are relatively isolated from its surroundings and thus are more recognizable. Indeed, all the civilizations in history group these seven stars together. Therefore, if we take the criterion that a good clustering should keep "Big Dipper" unseparated, then the real value $K = 41$ is too large.

In sum,one problem when running K-means algorithms is to select appropriate $K$. We have shown that selection $K$ by the function $f(K)$ and by the real value are not good choices. Therefore, we would better adopt an algorithm without the need to prespecify $K$.



(a) $M_c = 2.6$. Input:$K = 41$.

(b) $M_c = 4.6$. Input:$K = 86$.

FIG. 2: Results of spherical K-means++. Stars with the same color are in the same cluster. The radius of the circle represents the magnitude, i.e. larger circle means brighter.

### Affinity Propagation

Affinity propagation (AP)[5] is a clustering algorithm that does not require the number of clusters to be determined or estimated before running the algorithm. Unlike $K$-means, AP simultaneously considers all data points as potential cluster centroids (exemplars). To find appropriate exemplars, AP updates two evidence matrices as:

$$
\begin{aligned}
r(i,k) &\leftarrow s(i,k) - \max_{k' \neq k}(r(i,k') + s(i,k')) \\
a(i,k) &\leftarrow \min\left(0, r(k,k) + \max_{i' \notin \{i,k\}}(0, r(i',k))\right) \quad i \neq k \\
a(k,k) &\leftarrow \max_{i' \neq k}(0, r(i',k))\,. \qquad (3)
\end{aligned}
$$

Finally, larger the $r(:,k) + a(:,k)$, more probability the point $k$ as a final cluster center. The algorithm involves three matrices:

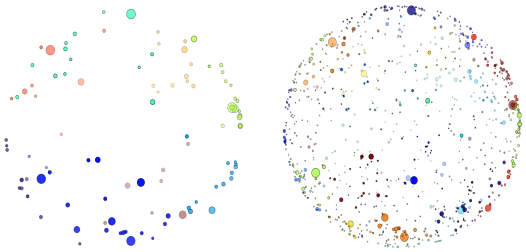
• The similarity $s(i,k)$, input matrix indicating how well the data point with index $k$ is suited to be the exemplar for data point $i$. For our problem, we set

$$s(i,k) = 2 - d(x^{(i)}, x^{(k)}) = 1 + \cos\langle x^{(i)}, x^{(k)}\rangle\,. \qquad (4)$$

• The "responsibility" matrix $r(i,k)$, sent from data point $i$ to candidate exemplar point $k$, reflects the accumulated evidence for how well-suited point $k$ is to serve as the exemplar for point $i$, taking into account other potential exemplars for point $i$.

• The "availability" matrix $a(i,k)$, sent from candidate exemplar point $k$ to point $i$, reflects the accumulated evidence for how appropriate it would be for point $i$ to choose point $k$ as its exemplar, taking into account the support from other points that point $k$ should be an exemplar.

The clustering results are shown in Fig. 3(a) and (b). 8 clusters are obtained in Fig. 3(a) and 33 for Fig. 3(b). The number of clusters given by AP is much smaller than the real values and the seven stars in the "Big Dipper" stay in the same group.



(a) $M_c = 2.6$. Output: $K = 8$ (b) $M_c = 4.6$. Output: $K = 33$

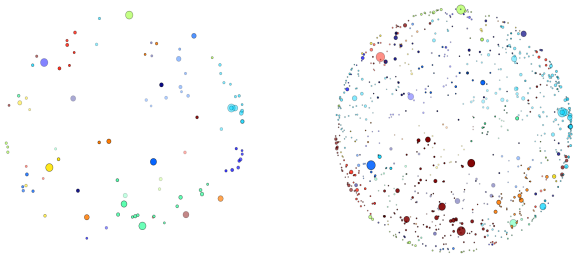
FIG. 3: Results of Affinity propagation with damping factor 0.5. Preferences are the median of the input similarities

Although we need not prespecify $K$ when running AP, the number of outcome exemplars (clusters) is influenced by the values of the input preferences $s(k, k)$, the diagonal elements of the input similarity matrix. In our setting, preferences are set to the median of the input similarities, resulting in a moderate number of clusters.

## DBSCAN

DBSCAN [6], short for Density-Based Spatial Clustering of Applications with Noise, is another clustering algorithm that does not need to presepcify $K$. It is based on local densities of the data and requires inputs Eps and minDist values to determine within what distance a point will belong to a cluster (reachable), and how the points that belong the same cluster are connected. But DBSCAN will also detect the noise data from the dataset, namely the points that don't belong to any cluster.

As stated in Ref.[6], the suggested optimal minDist value for two-dimensional data would be 4. However, in our problem, we found that minDist = 4 is not the best choice. The number of the resulting clusters is so small that most of the stars are treated as noises. Instead, we searched the parameter space of minDist and Eps to reach the number of clusters as close as the real number. For $M_c = 2.6$, the best pair Eps = 1.3 and minDist = 1 will give 40 clusters, which is close to 41. When $M_c = 4.6$, running DBSCAN algorithm for the chosen stars for Eps = 2.02 and minDist = 2 will give 89 clusters with 45 noise points. The following two figures show the result for the clustering of the "Big Dipper" and "Orion". We can see that the "Big Dipper" could be easily recognized except for one star in the neighboring cluster. The constellation "Orion" belongs to a large cluster with a high density of stars. Since the algorithm sets a global density based on input arguments Eps and minDist, it tends to choose the stars in high density region as the clusters and those in low density region as the noises.



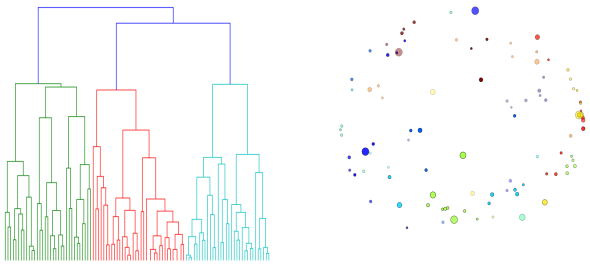
(a) $M_c = 2.6$ (b) $M_c = 4.6$

FIG. 4: Results of DBSCAN.

## Other Methods

Just for completeness, we also call other clustering methods in scikit-learn[7], hierarchical clustering and spectral clustering. As $K$-means, these two methods need to prespecify $K$.

The resulting dendrogram and clusters for stars when $M_c = 2.6$ by agglomerative approach of the spectral clustering are shown as follows: The number of intersects



(a) Dendrogram (b) Clustering Results

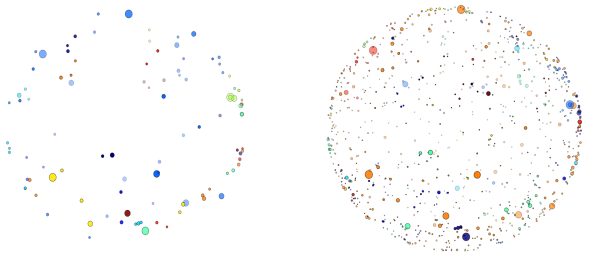
FIG. 5: Results of hierarchical clustering when $M_c = 2.6$ with Ward linkage and input $K = 41$ and Ward linkage.

when a vertical line is drew across the dendrogram will be the number of clusters. Inputing the real value $K = 41$ as one of the arguments of hierarchical clustering results in Fig. 5 (b). Again, as $K$-means, the "Big Clippers" are no longer classified in the same cluster.

Spectral clustering techniques first make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction, then followed by a $K$-Means in the low dimensional space[8]. Although still running $K$-means, spectral clustering has the ability to detect non-convex boundaries of the clusters. However, for the constellation clustering, we do not need the non-convex boundaries.

## GUI IMPLEMENTATION

In order to facilitate the clustering process, we also build up a GUI to call the algorithms in scikit-learn [7]



(a) $M_c = 2.6$. Input:$K = 41$.

(b) $M_c = 4.6$. Input:$K = 86$.

FIG. 6: Results of spectral clustering.

as well as a 3D visualization. The screenshot is seen in Fig. 7.

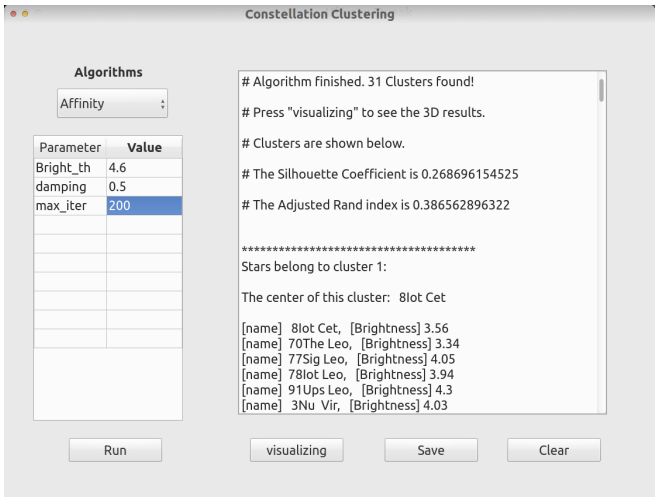



FIG. 7: GUI screenshot

## COMPARISON OF ALL ALGORITHMS

### Adjusted Rand Index

Adjusted Rand Index (ARI) [7] is to compare the result with the ground truth labels. ARI has a bounded range $[-1, 1]$, where negative values are bad (independent labelings) and similar clusterings have a positive ARI. Random label assignments have a ARI score close to 0. while 1.0 is the perfect match score. In this work, we use the real constellations as the ground truth labels. The Mathematical formulation is as follows.

If $C$ is a ground truth class assignment and $K$ is the clustering, we define $a$ as the number of pairs of elements that are in the same set in $C$ and in the same set in $K$, and $b$ as the number of pairs of elements that are in different sets in $C$ and in different sets in $K$. To guarantee that random label assignments will get a value close to zero, we then discount the expected Rand Index of random labelings:

$$\mathrm{RI} = \frac{a+b}{C_2^{n_{\mathrm{samples}}}} \tag{5}$$

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathrm{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathrm{E}[\mathrm{RI}]} \tag{6}$$

where $C_2^{n_{\mathrm{samples}}}$ is the total number of possible pairs in the dataset.

### Silhouette Coefficient

If the ground truth labels are not known, evaluation must be performed using the model itself. The Silhouette Coefficient (SC) [7] is an example. The SC score is bounded between $-1$ for incorrect clustering and $+1$ for highly dense clustering. Scores around zero indicate overlapping clusters. The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster. The SC is defined for each sample and is composed of two scores $a$ and $b$. $a$ is the mean distance between a sample and all other points in the same class and $b$ is the mean distance between a sample and all other points in the next nearest cluster. The SC for a single sample is then given as:

$$s = \frac{b-a}{\max(a,b)} \tag{7}$$

The SC for a set of samples is given as the mean of the SC for each sample.

### Comparison Table

| Algorithms | $M_c = 2.6$ | | $M_c = 4.6$ | |
|---|---|---|---|---|
| | ARI | SC | ARI | SC |
| K-means | 0.61 | 0.60 | 0.43 | 0.53 |
| Hierarchical | 0.67 | 0.53 | 0.45 | 0.50 |
| Spectral | 0.18 | 0.25 | 0.04 | -0.08 |
| Affinity | 0.32 | 0.59 | 0.41 | 0.48 |
| DBSCAN | 0.57 | 0.34 | 0.23 | -0.03 |

To compare with the ground truth labels, all the prespecified $K$, if needed, are taken to be the same as the number of real constellations. From the comparison table, we found that spherical K-means, affinity propagation, and hierarchical clustering have better performance. Especially, they have similar SC score. Note that the reason for the relatively low ARI score of affinity propagation is the difference between the number of generated clusters and the real number of constellations.

## FUTURE WORK

In this work, we use various clustering algorithms to re-clustering constellations. For each algorithm, we hope to provide further justifications to pick appropriate input parameters.

Stars magnitude is simply treated as a threshold. There maybe a more reasonable way to deal with the magnitude. For example, in spherical K-means, magnitude can be used as a weight factor in the dissimilarity measure. Or in affinity propagation, we may relate the input preference to each star's magnitude.

Finally, we can improve visualization of the clustered constellations by methods like 3D rendering. It would provide us with a real life view of how the generated constellations distribute in the universe.


[1] I. S. Dhillon, J. Fan, and Y. Guan, in *Data mining for scientific and engineering applications* (Springer, 2001) pp. 357–381.

[2] D. T. Pham, S. S. Dimov, and C. Nguyen, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science **219**, 103 (2005).

[3] R. Tibshirani, G. Walther, and T. Hastie, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **63**, 411 (2001).

[4] D. Arthur and S. Vassilvitskii, in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (Society for Industrial and Applied Mathematics, 2007) pp. 1027–1035.

[5] B. J. Frey and D. Dueck, Science **315**, 972 (2007).

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, in *Kdd*, Vol. 96 (1996) pp. 226–231.

[7] "http://scikit-learn.org/stable/modules/clustering,".

[8] A. Y. Ng, M. I. Jordan, and Y. Weiss, in *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS* (MIT Press, 2001) pp. 849–856.