Clustering Techniques

STEPHEN FANALE

What is Clustering? Why is Clustering?

Structure in data

- Concentration of points
- Segmentation of data

Unsupervised Learning is Key

Clustering is similar but not the same as classification

No prior information about the numbers or properties of clusters

No score or error function to evaluate effectiveness

Background: Density

Gaussian Mixed Method

Models underlying density of points as sums of Gaussians

BIC is a useful tool for finding how many statistically significant clusters – but note this is different from how many clusters there actually are

Further research- 6.3 Parametric density estimation

Different methods

Sum squares minimization

Radius minimization

Clustering around peaks

Clustering procedurally

Clustering by Sum-of-Squares Minimization: *K*-Means

- Chose centroid of each of K clusters
- Assign each point to closest cluster
- Update locations of centroids of each cluster by recalculating μ

$$\sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k||^2$$

```
import numpy as np
from sklearn.cluster import KMeans

X = np.random.normal(size=(1000, 2)) # 1000 pts in 2
    #dims
clf = KMeans(n_clusters=3)
clf.fit(X)
centers=clf.cluster_centers_ # locations of the
        # clusters
labels=clf.predict(X) # labels for each of the
       # points
```

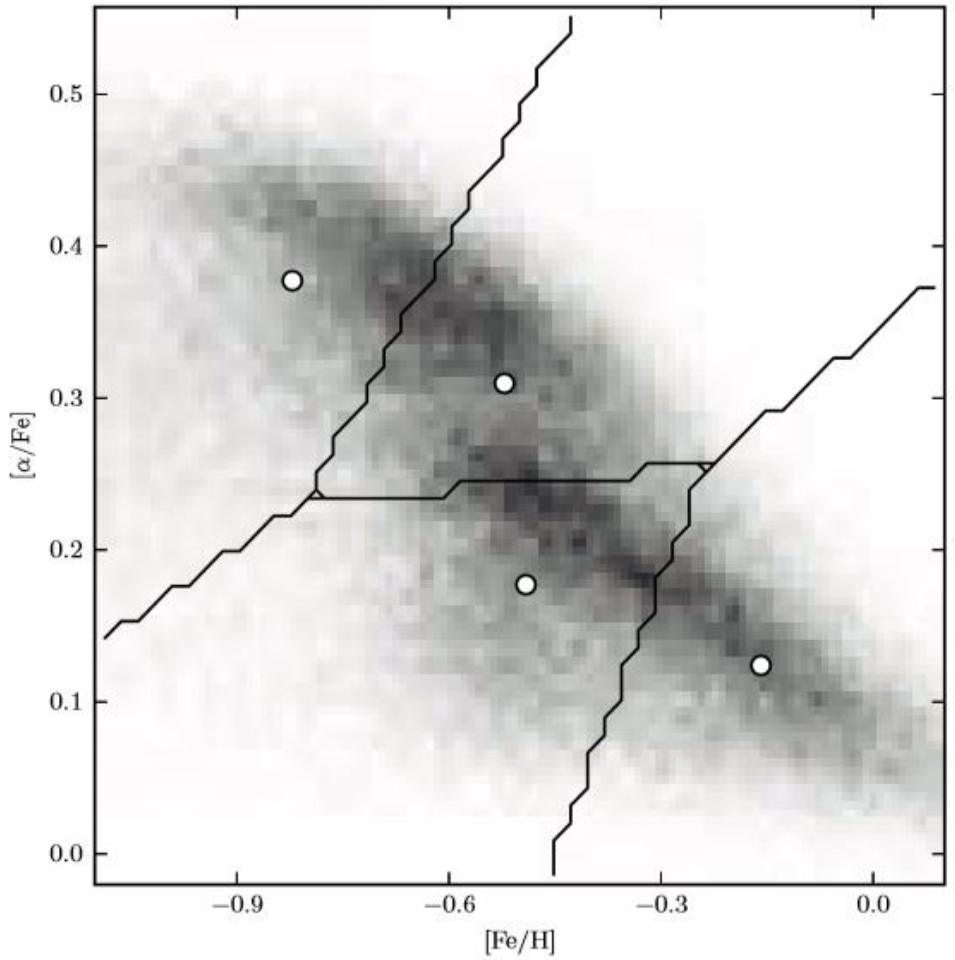

Figure 6.13. The *K*-means analysis of the stellar metallicity data used in figure 6.6. Note how the background distribution "pulls" the cluster centers away from the locus where one would place them by eye. This is why more sophisticated models like GMM are often better in practice.

Clustering by Max-Radius Minimization: The Gonzalez Algorithm

- Similar to K-Means
- *The Gonzalez Algorithm*
  - Starting with no clusters, progressively add one cluster with arbitrary center
  - Find point X that maximizes the distance from the centers of existing clusters and make that the next cluster
  - Continues until K clusters

$$\min_{k} \max_{x_i \in C_k} ||x_i - \mu_k||$$

Clustering by Nonparametric Density Estimation: Mean Shift

- Move the data points in the direction of the log of the gradient of the density of the data, until they converge at the peaks
- For Epanechnikov kernel and the value $a = \frac{h^2}{D+2}$, the update rule reduces to $x_i^{m+1} =$
  $mean\ position\ of\ points\ x_i^m\ within\ distance\ h\ of\ x_i^m$

```
import numpy as np
from sklearn.cluster import MeanShift

X = np.random.normal(size=(1000, 2)) # 1000 pts in 2
    # dims
ms = MeanShift(bandwidth=1.0)
     # if no bandwidth is specified,
     # it will be learned from data
ms.fit(X) # fit the data
centers = ms.cluster_centers_ # centers of clusters
labels = ms.labels_ # labels of each point X
```

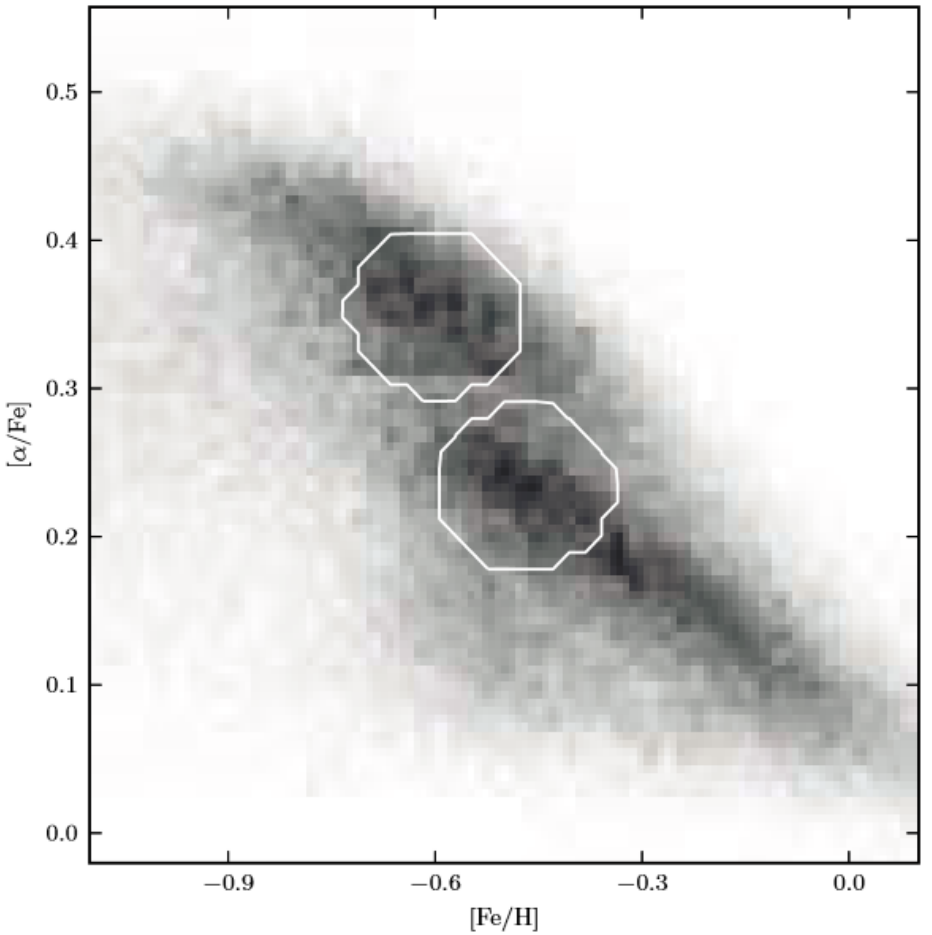

Figure 6.14. Mean-shift clustering on the metallicity data set used in figures 6.6 and 6.13. Th method finds two clusters associated with local maxima of the distribution (interior of th circles). Points outside the circles have been determined to lie in the background. The mea shift does not attempt to model correlation in the clusters: that is, the resulting clusters are ax aligned.

Clustering Procedurally: Hierarchical Clustering

- Partition into N clusters
- Join clusters until Nth partition contains one cluster
- If two points are in the same cluster at level m and remain together they are hierarchically clustered

$$d_{\min}(C_k, C_{k'}) = \min_{x \in C_k, x' \in C_{k'}} ||x - x'||,$$

$$d_{\max}(C_k, C_{k'}) = \max_{x \in C_k, x' \in C_{k'}} ||x - x'||,$$

$$d_{\text{avg}}(C_k, C_{k'}) = \frac{1}{N_k N_{k'}} \sum_{x \in C_k} \sum_{x' \in C_{k'}} ||x - x'||,$$

$$d_{\text{cen}}(C_k, C_{k'}) = ||\mu_k - \mu_{k'}||,$$

```
from scipy.sparse.csgraph import\
     minimum_spanning_tree
from sklearn.neighbors import kneighbors_graph

X = np.random.random((1000, 2)) # 1000 pts in 2 dims
G = kneighbors_graph(X, n_neighbors=10,
                     mode='distance')
T = minimum_spanning_tree(G)
```

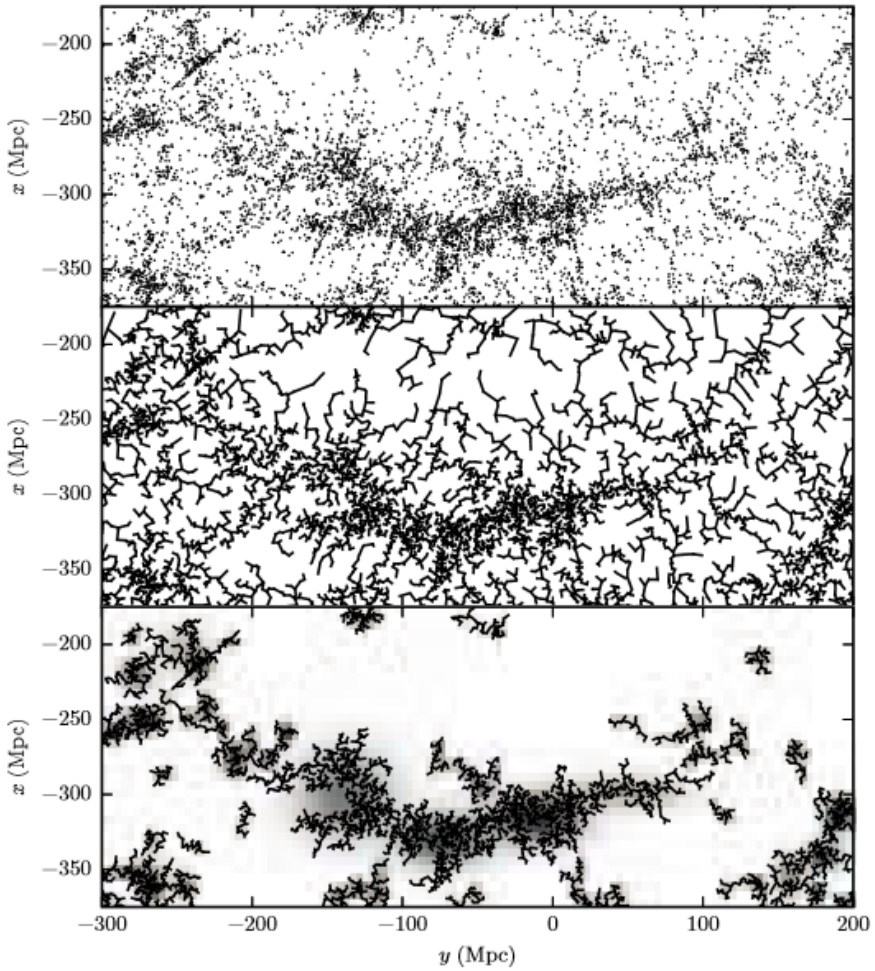

Figure 6.15. An approximate Euclidean minimum spanning tree over the two-dimensional projection of the SDSS Great Wall. The upper panel shows the input points, and the middle panel shows the dendrogram connecting them. The lower panel shows clustering based on this dendrogram, created by removing the largest 10% of the graph edges, and keeping the remaining connected clusters with 30 or more members. See color plate 4.

A Brief Comparison

K-means and max-radius are parametric in theory, however murky in practice

Mean shift is nonparametric in spirit, though final step is similar to K-means

Hierarchical is nonparametric

Accuracy of clustering methods are not well known

K-means and max-radius are fast and simple but fiddly

Hierarchical is accurate but slow, can be sped up using algorithms

Further Comparison

TABLE 6.1.
Summary of the practical properties of different unsupervised techniques.

Method	Accuracy	Interpretability	Simplicity	Speed
K-nearest-neighbor	H	H	H	M
Kernel density estimation	H	H	H	H
Gaussian mixture models	H	M	M	M
Extreme deconvolution	H	H	M	M
K-means	L	M	H	M
Max-radius minimization	L	M	M	M
Mean shift	M	H	H	M
Hierarchical clustering	H	L	L	L
Correlation functions	H	M	M	M

Simple summary Table 6.1 is a quick summary of our assessment of each of the methods considered in this chapter, in terms of high (H), medium (M), and low (L). Note that the table mixes methods for different tasks (density estimation, clustering, correlation functions), so they are not all directly comparable to each other.

Sources

Ivezic 6.4, 6.6