Clustering 3: Hierarchical clustering (continued); choosing the number of clusters

Optional reading: ISL 10.3, ESL 14.3

Even more linkages

Last time we learned about hierarchical agglomerative clustering, basic idea is to repeatedly merge two most similar groups, as measured by the linkage

Three linkages: single, complete, average linkage. Properties:

- Single and complete linkage can have problems with chaining and crowding, respectively, but average linkage doesn't
- Cutting an average linkage tree provides no interpretation, but there is a nice interpretation for single, complete linkage trees
- Average linkage is sensitive to a monotone transformation of the dissimilarities d_{ij}, but single and complete linkage are not
- All three linkages produce dendrograms with no inversions

Actually, there are many more linkages out there, each having different properties. Today: we'll look at two more

Reminder: linkages

Our setup: given $X_1, \dots X_n$ and pairwise dissimilarities d_{ij}. (E.g., think of $X_i \in \mathbb{R}^p$ and $d_{ij} = \|X_i - X_j\|_2$)

Single linkage: measures the closest pair of points

$$d_{\text{single}}(G,H) = \min_{i \in G,\, j \in H} d_{ij}$$

Complete linkage: measures the farthest pair of points

$$d_{\text{complete}}(G,H) = \max_{i \in G,\, j \in H} d_{ij}$$

Average linkage: measures the average dissimilarity over all pairs

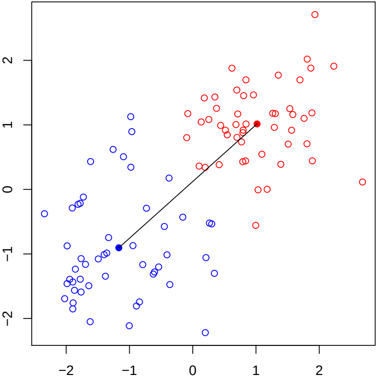
$$d_{\text{average}}(G,H) = \frac{1}{n_G \cdot n_H} \sum_{i \in G,\, j \in H} d_{ij}$$

Centroid linkage

Centroid linkage[1] is commonly used. Assume that $X_i \in \mathbb{R}^p$, and $d_{ij} = \|X_i - X_j\|_2$. Let $\bar{X}_G, \bar{X}_H$ denote group averages for G, H. Then:

$$d_{\text{centroid}}(G, H) = \|\bar{X}_G - \bar{X}_H\|_2$$

Example (dissimilarities d_{ij} are distances, groups are marked by colors): centroid linkage score $d_{\text{centroid}}(G, H)$ is the distance between the group centroids (i.e., group averages)

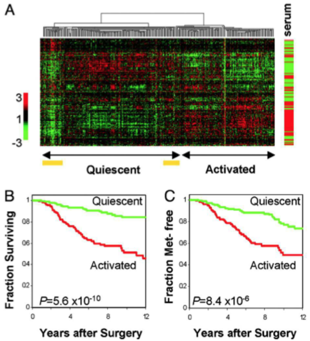


[1] Eisen et al. (1998), "Cluster Analysis and Display of Genome-Wide Expression Patterns"

Centroid linkage is the standard in biology

Centroid linkage is simple: easy to understand, and easy to implement. Maybe for these reasons, it has become the standard for hierarchical clustering in biology

Fig. 1. Performance of a "wound response" gene expression signature in predicting breast cancer progression

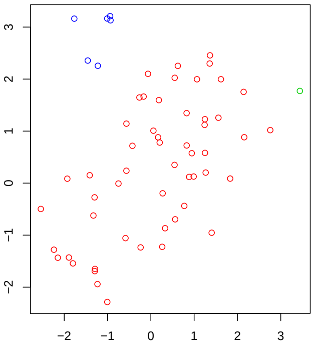


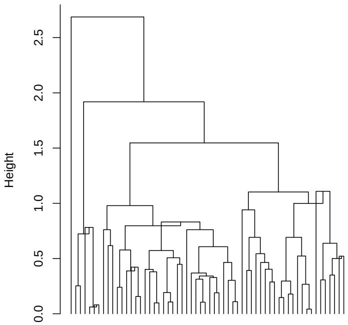
Chang, Howard Y. et al. (2005) Proc. Natl. Acad. Sci. USA 102, 3738-3743

PNAS

Centroid linkage example

Here $n = 60$, $X_i \in \mathbb{R}^2$, $d_{ij} = \|X_i - X_j\|_2$. Cutting the tree at some heights wouldn't make sense ... because the dendrogram has inversions! But we can, e.g., still look at ouptut with 3 clusters

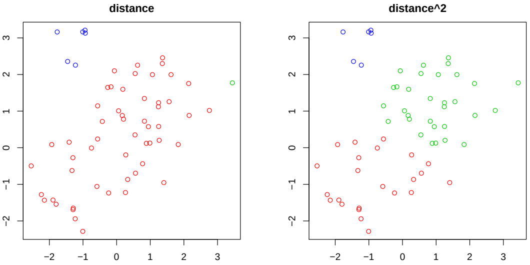




Cut interpretation: there isn't one, even with no inversions

Shortcomings of centroid linkage

- Can produce dendrograms with inversions, which really messes up the visualization
- Even if were we lucky enough to have no inversions, still no interpretation for the clusters resulting from cutting the tree
- Answers change with a monotone transformation of the dissimilarity measure $d_{ij} = \|X_i - X_j\|_2$. E.g., changing to $d_{ij} = \|X_i - X_j\|_2^2$ would give a different clustering

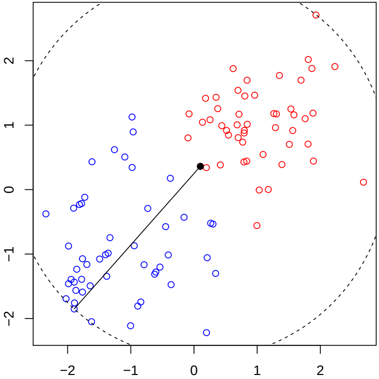


Minimax linkage

Minimax linkage[2] is a newcomer. First define radius of a group of points G around X_i as $r(X_i, G) = \max_{j \in G} d_{ij}$. Then:

$$d_{\text{minimax}}(G, H) = \min_{i \in G \cup H} r(X_i, G \cup H)$$

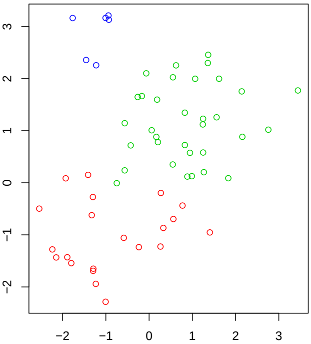
Example (dissimilarities d_{ij} are distances, groups marked by colors): minimax linkage score $d_{\text{minimax}}(G, H)$ is the smallest radius encompassing all points in G and H. The center X_c is the black point

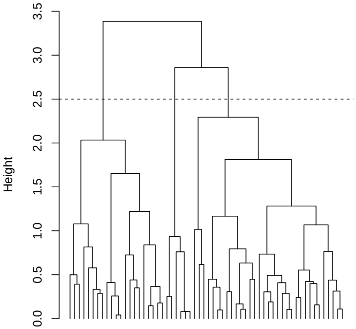


[2] Bien et al. (2011), "Hierarchical Clustering with Prototypes via Minimax Linkage"

Minimax linkage example

Same data s before. Cutting the tree at $h = 2.5$ gives clustering assignments marked by the colors



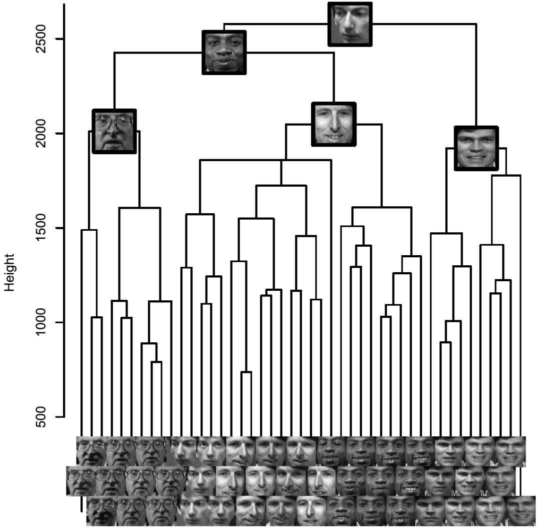


Cut interpretation: each point X_i belongs to a cluster whose center X_c satisfies $d_{ic} \leq 2.5$

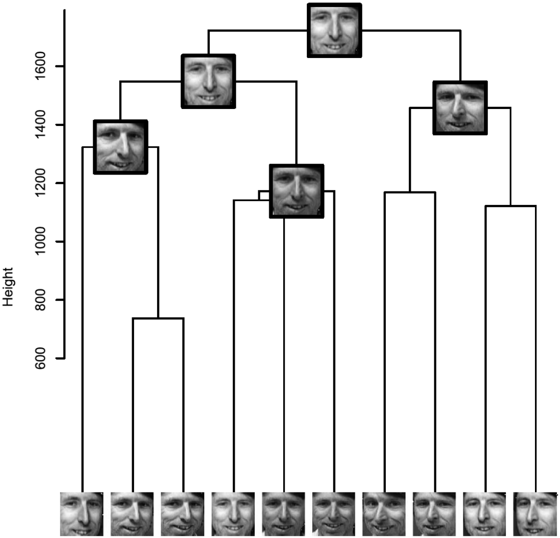
Properties of minimax linkage

- Cutting a minimax tree at a height h a nice interpretation: each point is $\leq h$ in dissimilarity to the center of its cluster. (This is related to a famous set cover problem)
- Produces dendrograms with no inversions
- Unchanged by monotone transformation of dissimilarities d_{ij}
- Produces clusters whose centers are chosen among the data points themselves. Remember that, depending on the application, this can be a very important property. (Hence minimax clustering is the analogy to K-medoids in the world of hierarchical clustering)

Example: Olivetti faces dataset



(From Bien et al. (2011))



(From Bien et al. (2011))

Centroid and minimax linkage in R

The function `hclust` in the base package performs hierarchical agglomerative clustering with centroid linkage (as well as many other linkages)

E.g.,

```
d = dist(x)
tree.cent = hclust(d, method="centroid")
plot(tree.cent)
```

The function `protoclust` in the package `protoclust` implements hierarchical agglomerative clustering with minimax linkage

Linkages summary

Linkage	No inversions?	Unchanged with monotone transformation?	Cut interpretation?	Notes
Single	✓	✓	✓	chaining
Complete	✓	✓	✓	crowding
Average	✓	×	×	
Centroid	×	×	×	simple
Minimax	✓	✓	✓	centers are data points

Note: this doesn't tell us what "best linkage" is

What's missing here: a detailed empirical comparison of how they perform. On top of this, remember that choosing a linkage can be very situation dependent

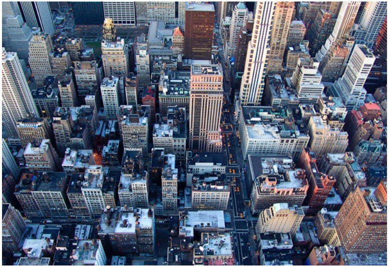
Designing a clever radio system (e.g., Pandora)

Suppose we have a bunch of songs, and dissimilarity scores between each pair. We're building a clever radio system—a user is going to give us an initial song, and a measure of how "risky" he is going to be, i.e., maximal tolerable dissimilarity between suggested songs

How could we use hierarchical clustering, and with what linkage?

Placing cell phone towers

Suppose we are helping to place cell phone towers on top of some buildings throughout the city. The cell phone company is looking to build a small number of towers, such that no building is further than half a mile from a tower



How could we use hierarchical clustering, and with what linkage?

How many clusters?

Sometimes, using K-means, K-medoids, or hierarchical clustering, we might have no problem specifying the number of clusters K ahead of time, e.g.,

- Segmenting a client database into K clusters for K salesman
- Compressing an image using vector quantization, where K controls the compression rate

Other times, K is implicitly defined by cutting a hierarchical clustering tree at a given height, e.g., designing a clever radio system or placing cell phone towers

But in most exploratory applications, the number of clusters K is unknown. So we are left asking the question: what is the "right" value of K?

This is a hard problem

Determining the number of clusters is a hard problem!

Why is it hard?

- Determining the number of clusters is a hard task for humans to perform (unless the data are low-dimensional). Not only that, it's just as hard to explain what it is we're looking for. Usually, statistical learning is successful when at least one of these is possible

Why is it important?

- E.g., it might mean a big difference scientifically if we were convinced that there were $K = 2$ subtypes of breast cancer vs. $K = 3$ subtypes
- One of the (larger) goals of data mining/statistical learning is automatic inference; choosing K is certainly part of this

Reminder: within-cluster variation

We're going to focus on K-means, but most ideas will carry over to other settings

Recall: given the number of clusters K, the K-means algorithm approximately minimizes the within-cluster variation:

$$W = \sum_{k=1}^{K} \sum_{C(i)=k} \|X_i - \bar{X}_k\|_2^2$$

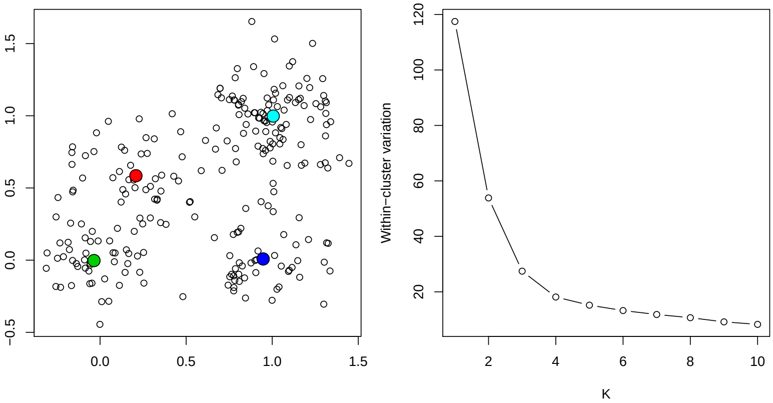
over clustering assignments C, where $\bar{X}_k$ is the average of points in group k, $\bar{X}_k = \frac{1}{n_k} \sum_{C(i)=k} X_i$

Clearly a lower value of W is better. So why not just run K-means for a bunch of different values of K, and choose the value of K that gives the smallest $W(K)$?

That’s not going to work

Problem: within-cluster variation just keeps decreasing

Example: $n = 250$, $p = 2$, $K = 1, \ldots 10$



Between-cluster variation

Within-cluster variation measures how tightly grouped the clusters are. As we increase the number of clusters K, this just keeps going down. What are we missing?

Between-cluster variation measures how spread apart the groups are from each other:

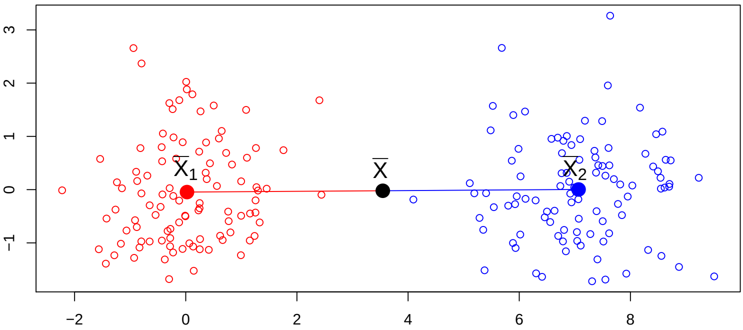
$$B = \sum_{k=1}^{K} n_K \|\bar{X}_k - \bar{X}\|_2^2$$

where as before $\bar{X}_k$ is the average of points in group k, and $\bar{X}$ is the overall average, i.e.

$$\bar{X}_k = \frac{1}{n_k} \sum_{C(i)=k} X_i \quad \text{and} \quad \bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Example: between-cluster variation

Example: $n = 100$, $p = 2$, $K = 2$



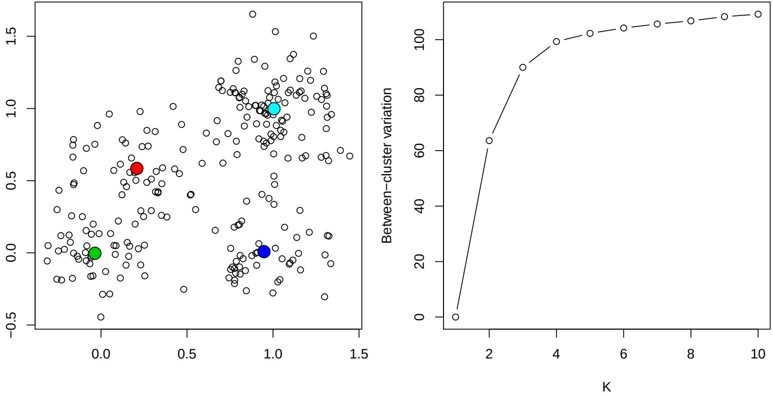
$$B = n_1 \|\bar{X}_1 - \bar{X}\|_2^2 + n_2 \|\bar{X}_2 - \bar{X}\|_2^2$$

$$W = \sum_{C(i)=1} \|X_i - \bar{X}_1\|_2^2 + \sum_{C(i)=2} \|X_i - \bar{X}_2\|_2^2$$

Still not going to work

Bigger B is better, can we use it to choose K? Problem: between-cluster variation just keeps increasing

Running example: $n = 250$, $p = 2$, $K = 1, \ldots 10$



CH index

Ideally we'd like our clustering assignments C to simultaneously have a small W and a large B

This is the idea behind the CH index.[3] For clustering assignments coming from K clusters, we record CH score:

$$\mathsf{CH}(K) = \frac{B(K)/(K-1)}{W(K)/(n-K)}$$

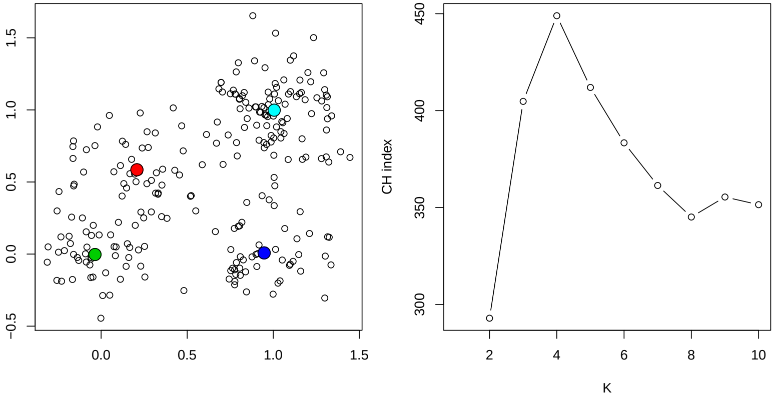
To choose K, just pick some maximum number of clusters to be considered K_{max} (e.g., $K = 20$), and choose the value of K with the largest score $\mathsf{CH}(K)$, i.e.,

$$\hat{K} = \underset{K \in \{2, \ldots K_{\mathsf{max}}\}}{\operatorname{argmax}} \ \mathsf{CH}(K)$$

[3] Calinski and Harabasz (1974), "A dendrite method for cluster analysis"

Example: CH index

Running example: $n = 250$, $p = 2$, $K = 2, \dots 10$.



We would choose $K = 4$ clusters, which seems reasonable

General problem: the CH index is not defined for $K = 1$. We could never choose just one cluster (the null model)!

Gap statistic

It's true that $W(K)$ keeps dropping, but how much it drops at any one K should be informative

The gap statistic[4] is based on this idea. We compare the observed within-cluster variation $W(K)$ to $W_{\mathsf{unif}}(K)$, the within-cluster variation we'd see if we instead had points distributed uniformly (over an encapsulating box). The gap for K clusters is defined as

$$\mathsf{Gap}(K) = \log W_{\mathsf{unif}}(K) - \log W(K)$$

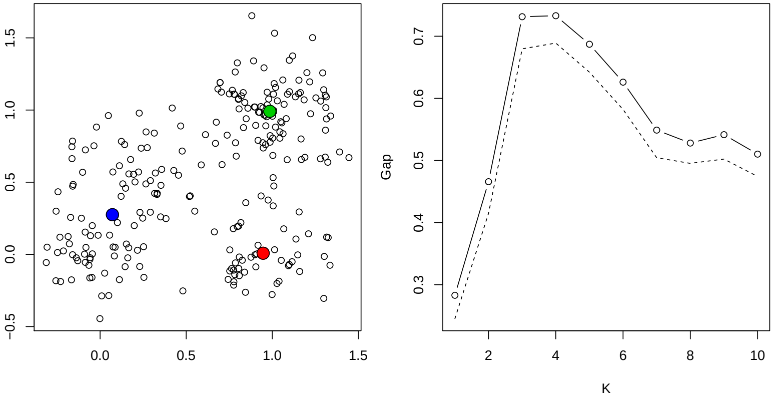
The quantity $\log W_{\mathsf{unif}}(K)$ is computed by simulation: we average the log within-cluster variation over, say, 20 simulated uniform data sets. We also compute the standard error of $s(K)$ of $\log W_{\mathsf{unif}}(K)$ over the simulations. Then we choose K by

$$\hat{K} = \min\Big\{K \in \{1, \ldots K_{\mathsf{max}}\} : \mathsf{Gap}(K) \geq \mathsf{Gap}(K+1) - s(K+1)\Big\}$$

[4] Tibshirani et al. (2001), "Estimating the number of clusters in a data set via the gap statistic"

Example: gap statistic

Running example: $n = 250$, $p = 2$, $K = 1, \ldots 10$



We would choose $K = 3$ clusters, which is also reasonable

The gap statistic does especially well when the data fall into one cluster. (Why? Hint: think about the null distribution that it uses)

CH index and gap statistic in R

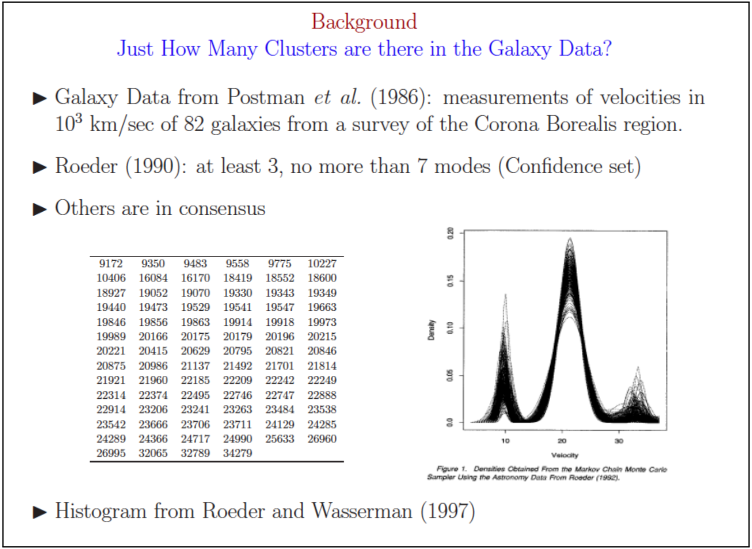
The CH index can be computed using the `kmeans` function in the base distribution, which returns both the within-cluster variation and the between-cluster varation (Homework 2)

E.g.,

```
k = 5
km = kmeans(x, k, alg="Lloyd")
names(km)
# Now use some of these return items to compute ch
```

The gap statistic is implemented by the function gap in the package `lga`, and by the function gap in the package `SAGx`. (Beware: these functions are poorly documented ... it's unclear what clustering method they're using)

Once again, it really is a hard problem



Background

Just How Many Clusters are there in the Galaxy Data?

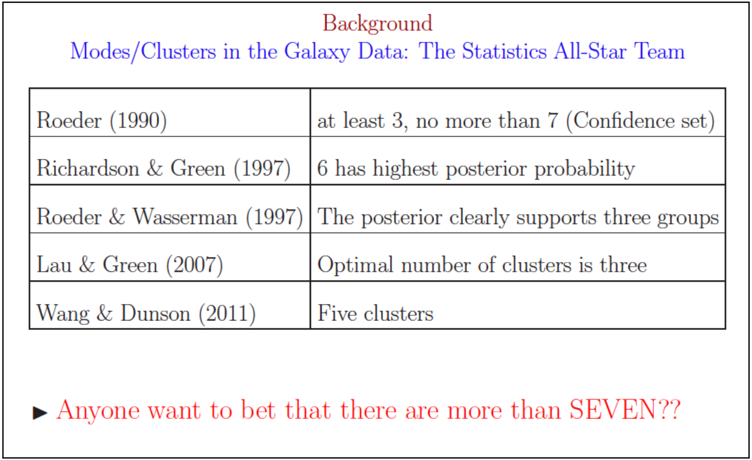
- Galaxy Data from Postman *et al.* (1986): measurements of velocities in 10^3 km/sec of 82 galaxies from a survey of the Corona Borealis region.
- Roeder (1990): at least 3, no more than 7 modes (Confidence set)
- Others are in consensus

9172	9350	9483	9558	9775	10227
10406	16084	16170	18419	18552	18600
18927	19052	19070	19330	19343	19349
19440	19473	19529	19541	19547	19663
19846	19856	19863	19914	19918	19973
19989	20166	20175	20179	20196	20215
20221	20415	20629	20795	20821	20846
20875	20986	21137	21492	21701	21814
21921	21960	22185	22209	22242	22249
22314	22374	22495	22746	22747	22888
22914	23206	23241	23263	23484	23538
23542	23666	23706	23711	24129	24285
24289	24366	24717	24990	25633	26960
26995	32065	32789	34279		

Figure 1. Densities Obtained From the Markov Chain Monte Carlo Sampler Using the Astronomy Data From Roeder (1992).

- Histogram from Roeder and Wasserman (1997)

(Taken from George Cassella's CMU talk on January 16 2011)



Background

Modes/Clusters in the Galaxy Data: The Statistics All-Star Team

Roeder (1990)	at least 3, no more than 7 (Confidence set)
Richardson & Green (1997)	6 has highest posterior probability
Roeder & Wasserman (1997)	The posterior clearly supports three groups
Lau & Green (2007)	Optimal number of clusters is three
Wang & Dunson (2011)	Five clusters

- Anyone want to bet that there are more than SEVEN??

(From George Cassella's CMU talk on January 16 2011)

Recap: more linkages, and determining K

Centroid linkage is commonly used in biology. It measures the distance between group averages, and is simple to understand and to implement. But it also has some drawbacks (inversions!)

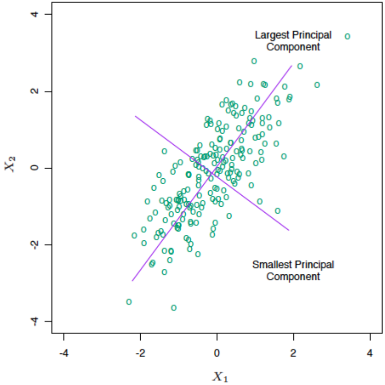
Minimax linkage is a little more complex. It asks the question: "which point's furthest point is closest?", and defines the answer as the cluster center. This could be useful for some applications

Determining the number of clusters is both a hard and important problem. We can't simply try to find K that gives the smallest achieved within-class variation. We defined between-cluster variation, and saw we also can't choose K to just maximize this

Two methods for choosing K: the CH index, which looks at a ratio of between to within, and the gap statistic, which is based on the difference between within-class variation for our data and what we'd see from uniform data

Next time: principal components analysis

Finding interesting directions in our data set



(From ESL page 67)