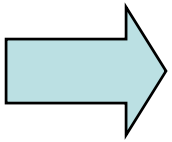
Lecture 15
Clustering

Oct – 31 – 2008

Unsupervised learning and pattern discovery

So far, our data has been in this form:

$x_1^1, x_2^1, x_3^1, \ldots, x_m^1$	y^1
$x_1^2, x_2^2, x_3^2, \ldots, x_m^2$	y^2
…	
…	
$x_1^n, x_2^n, x_3^n, \ldots, x_m^n$	y^n



We will be looking at ***unlabeled data:***

$x_1^1, x_2^1, x_3^1, \ldots, x_m^1$
$x_1^2, x_2^2, x_3^2, \ldots, x_m^2$
…
…
$x_1^n, x_2^n, x_3^n, \ldots, x_m^n$

What do we expect to learn from such data?

I have tons of data and need to:

organize it better – e.g., find subgroups

understand it better – e.g., understand interrelationships

find regular trends in it – e.g., If A and B, then C

The web organized by topic into categories.

Arts
Movies, Music, Television, ...

Business
Companies, Finance, Jobs, ...

Computers
Internet, Hardware, Software, ...

Games
Board, Roleplaying, Video, ...

Health
Alternative, Fitness, Medicine, ...

Home
Consumers, Homeowners, Family, ...

Kids and Teens
Computers, Entertainment, School, ...

News
Media, Newspapers, Current Events, ...

Recreation
Food, Outdoors, Travel, ...

Reference
Education, Libraries, Maps, ...

Regional
Asia, Europe, North America, ...

Science
Biology, Psychology, Physics, ...

Shopping
Autos, Clothing, Gifts, ...

Society
Issues, People, Religion, ...

Sports
Basketball, Football, Soccer, ...

World
Deutsch, Español, Français, Italiano, Japanese, Korean, Nederlands, Polska, Svenska, ...

Music

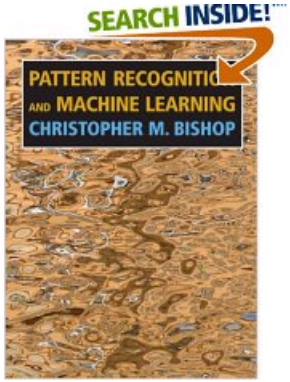
Arts > Music

Categories

Anime (61)
Anti-Music (172)
Arranging (2)
Awards (52)
Bands and Artists (47326)
Business (5322)
Charts (109)
Chats and Forums (242)
Classifieds (37)
Clubs and Venues (435)
Collecting (220)
Comedians (128)
Composition (16836)
Computers (38)
Concerts and Events (507)
Directories (120)
Disabled (17)
DJs (665)
Education (346)
History (107)
Humor (99)
Instruments (11418)
Living History (56)
Lyrics (787)
Marching (1379)
Movies (335)
Museums (46)
Music Videos (129)
Musical Theatre (1831)
Musicology (264)
News and Media (197)
Organizations (127)
People (34)
Personal Pages (194)
Photography (136)
Radio (36)
Record Labels (2900)
Regional (11)
Resources (173)
Reviews (685)
Shopping (2935)
Songwriting (405)
Sound Files (1484)
Styles (31892)
Technology (96)
Television Shows (178)
Theory (174)
Trading (328)
Video Games (153)
Vocal (2952)
Weblogs (105)
Weddings (13)
Women in Music (1716)

Hierarchical clustering as a way to organize web pages

Finding association patterns in data …



Share your own customer images
Search inside this book

Pattern Recognition and Machine Learning (Information Science and Statistics) (Hardcover)
by Christopher M. Bishop (Author)
★★★★☆ (22 customer reviews)

List Price: ~~$74.95~~
Price: **$50.11** & this item ships for **FREE with Super Saver Shipping**. Details
You Save: $24.84 (33%)

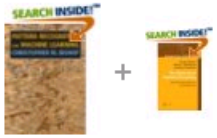
Upgrade this book for $14.99 more, and you can read, search, and annotate every page online. See details
Availability: In Stock. Ships from and sold by **Amazon.com**. Gift-wrap available.

Want it delivered Monday, October 29? Order it in the next 0 hours and 33 minutes, and choose **One-Day Shipping** at checkout. See details

58 used & new available from $43.59

Better Together

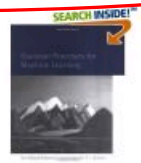
Buy this book with The Elements of Statistical Learning by T. Hastie today!



Buy Together Today: $118.84
Buy both now!

Customers Who Bought This Item Also Bought

Pattern Classification (2nd Edition) by



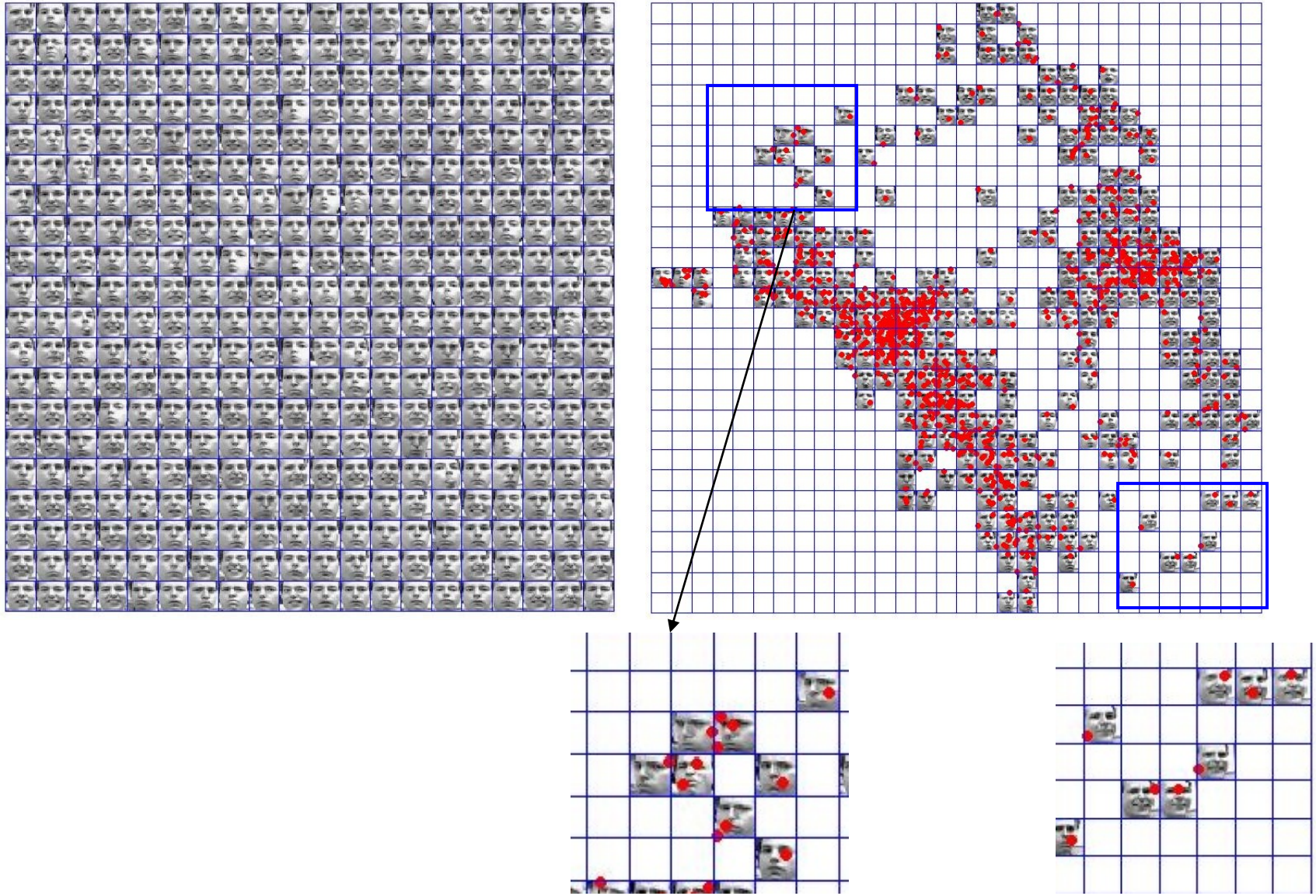
Gaussian Processes for Machine Learning by

Data Mining by Ian H. Witten

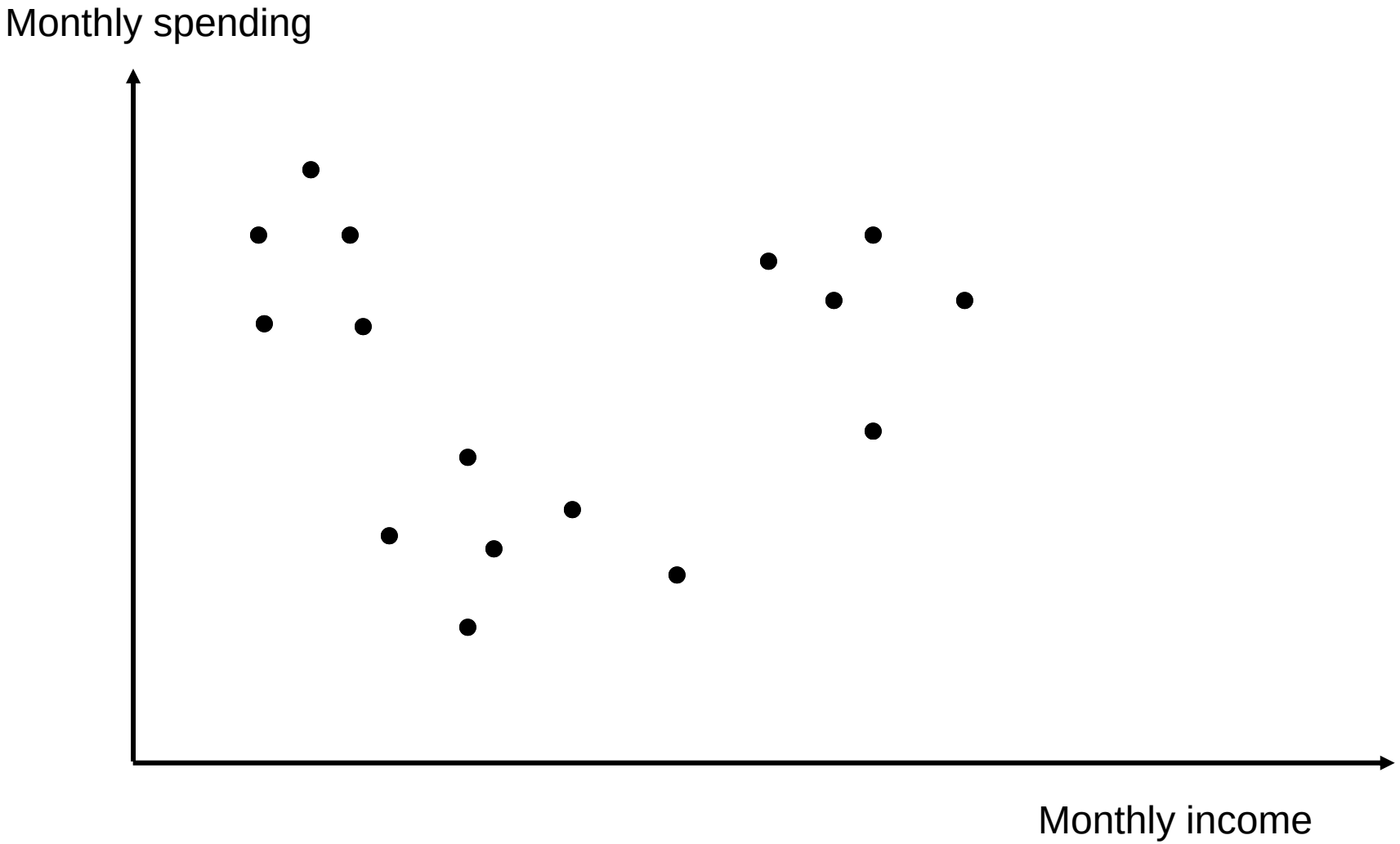
Machine Learning by Tom M. Mitchell

Computer Manual in MATLAB to Accompany

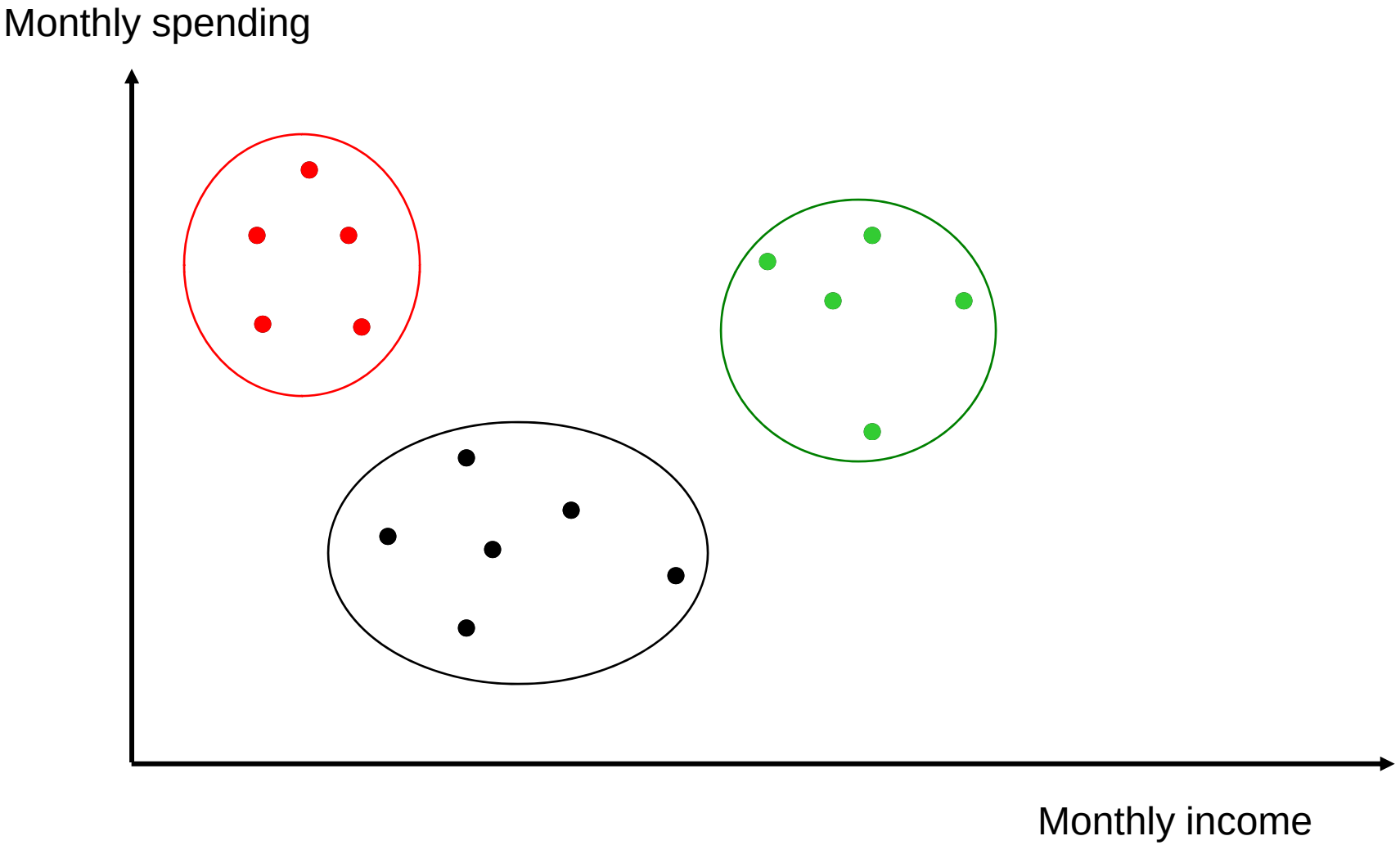
Dimension reduction for visualization



Clustering Example



Clustering Example



What is Clustering

- In general clustering can be viewed as an exploratory procedure for finding interesting subgroups in given data
- Very commonly, we focus on a special kind of clustering:
  - Group ***all*** given examples into ***disjoint*** subsets *of* clusters, such that:
    - Examples within a cluster are (very) similar
    - Examples in different clusters are (very) different

Example Applications

- Information retrieval – cluster retrieved documents to present more organized and understandable results
- Consumer market analysis – cluster consumers into different interest groups so that marketing plans can be specifically designed for each individual group
- Image segmentation: decompose an image into regions with coherent color and texture
- Vector quantization for data (i.e., image) compression: group vectors into similar groups, and use group mean to represent group members
- Computational biology: group gene into co-expressed families based on their expression profile using different tissue samples and different experimental conditions

Vector quantization for image compression

701,554 bytes

127,292 bytes

Important components in clustering

- Distance/similarity measure
  - How to measure the similarity between two objects?
- Clustering algorithm
  - How to find the clusters based on the distance/similarity measures
- Evaluation of results
  - How do we know that our clustering result is good

Distance Measures

- One of the most important question in unsupervised learning, often more important than the choice of clustering algorithms
- There are a set of commonly used distance measures
- Let’s look at them and think about when each of them might be appropriate or inappropriate

Common distance/similarity measures

- Euclidean distance $L_2(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{i=1}^{d} (x_i - x_i')^2}$
- City block distance (Manhattan distance)

$$L_1(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{d} |x_i - x_i'|$$

- Cosine similarity

$$cos(\mathbf{x}, \mathbf{x}') = \frac{<\mathbf{x} \cdot \mathbf{x}'>}{|\mathbf{x}| \cdot |\mathbf{x}'|}$$

- More flexible measures: $D(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{i=1}^{d} w_i (x_i - x_i')^2}$

one can learn the appropriate weights given user guidance

Note: We can always transform between distance and similarity using a monotonically decreasing function

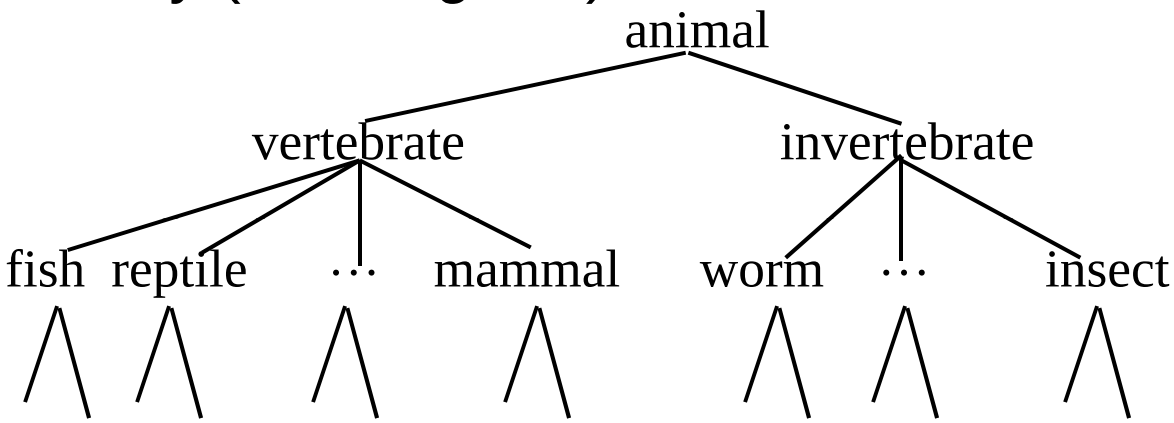
How to decide which to use?

- Usually need to consider the application domain, you need to ask questions such as:
  - What does it mean for two consumers to be similar to each other? Or for two genes to be similar to each other?
- Ideally we'd like to learn a distance ftn from user input
  - Ask users to provide things like object A is similar to object B, dissimilar to object C
  - Learn a distance function to correctly reflect these relationships
  - This is a more advanced topic that we will not cover in this class, but nonetheless important
- When we can not afford to learn a distance measure, and don't have a clue about which distance function is appropriate, what should we do?
- Remember, clustering is an exploratory procedure and it is important to explore – i.e., try different options

- Now we have seen different ways that one can use to measure distances or similarities among a set of unlabeled objects, how can we use them to group the objects into clusters?
- People have looked at many different approaches and they can be categorized into two distinct types

Hierarchical and non-hierarchical Clustering

- **Hierarchical clustering** builds a tree-based hierarchical taxonomy (*dendrogram*)

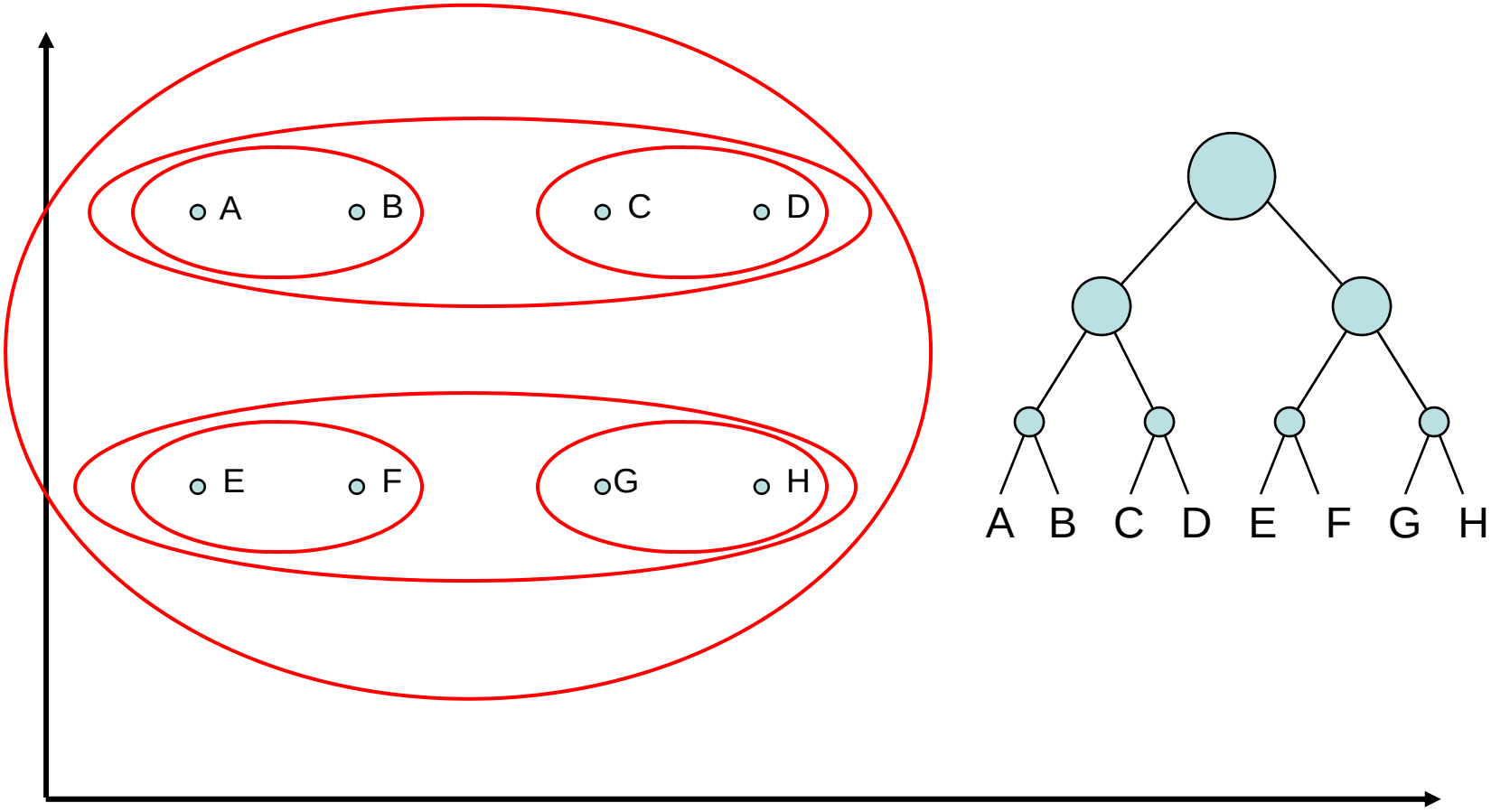


- **A non-hierarchical (flat) clustering** produces a single partition of the unlabeled data
- Choosing a particular level in the hierarchy produces a flat clustering
- Recursive application of flat clustering can also produce a hierarchical clustering.

Hierarchical Agglomerative Clustering (HAC)

- Assumes a *distance function* for determining the similarity of two instances.
  - One can also assume a similarity function, and reverse many of the operations in the algorithm to make it work for similarities
- Starts with each object in a separate cluster and then repeatedly joins the two closest clusters until only one cluster is left
- The history of merging forms a binary tree (or a hierarchy).

HAC Example



HAC Algorithm

Start with all objects in their own cluster.
Until there is only one cluster:
    Among the current clusters, determine the two clusters, c_i and c_j, that are closest
    Replace c_i and c_i with a single cluster $c_i \cup c_i$

Problem: our distance function computes distance between instances, but we also need to compute distance between clusters.

How?

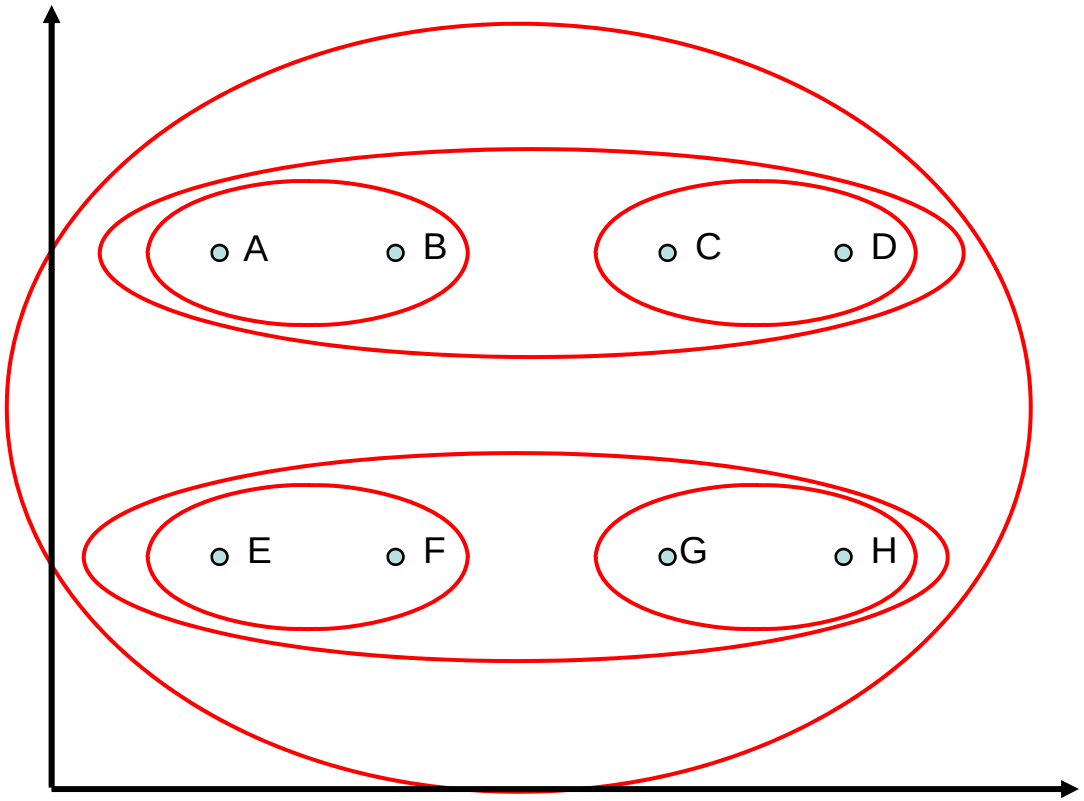
Distance Between Clusters

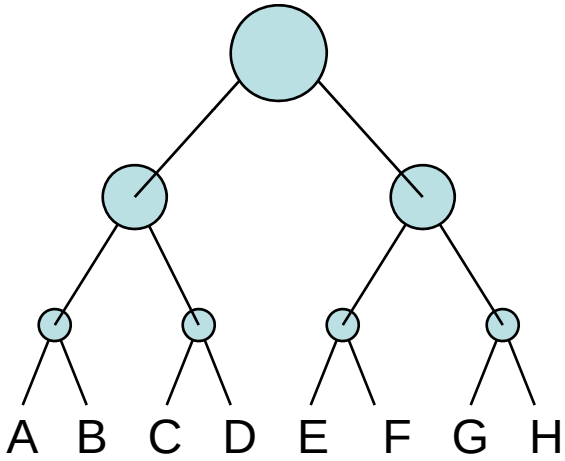
- Assume a distance function that determines the distance of two objects: $D(x,x')$.

- There are multiple way to turn a distance function into a cluster distance function:
  - Single Link: distance of two closest members of clusters
  - Complete Link: distance of two furthest members of clusters
  - Average Link: average distance

Single Link Agglomerative Clustering

- Use minimum distance of all pairs: $D(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{x}' \in C_j} D(\mathbf{x}, \mathbf{x}')$

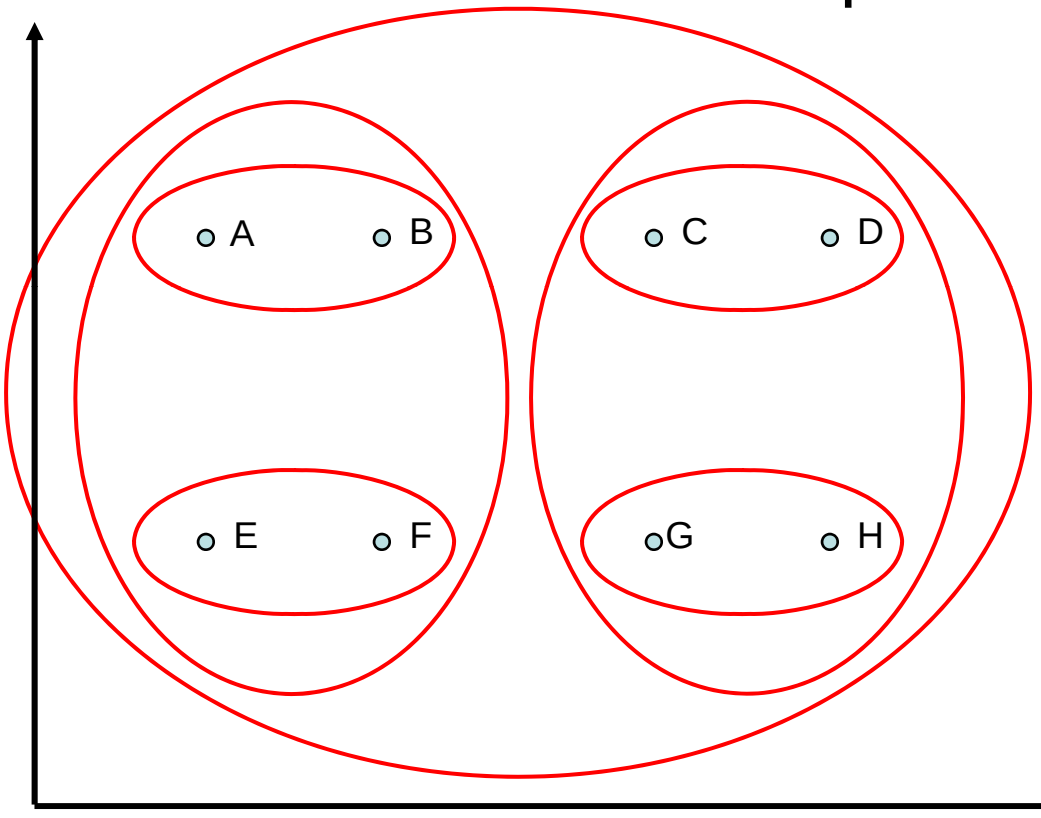


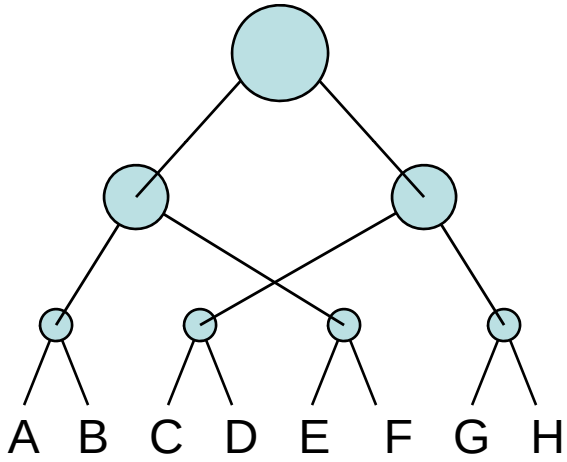


- Forms straggly clusters – long islands

Complete Link

- Maximum distance of all pairs: $D(C_i, C_j) = \max_{x \in c_i, x' \in c_j} D(x, x')$





- Makes “tight,” spherical clusters

Updating the Cluster Distances after merging is a piece of

- After merging c_i and c_j, the distance of the resulting cluster to any other cluster, c_k, can be computed by:
  - Single Link:

$$D((c_i \cup c_j), c_k) = \min(D(c_i, c_k), D(c_j, c_k))$$

  - Complete Link:

$$D((c_i \cup c_j), c_k) = \max(D(c_i, c_k), D(c_j, c_k))$$

- This is **constant** time given previous distances

Average Link

- Basic idea: average similarity between members of the two clusters
- Two options are available:
  - Averaged across all ordered pairs in the merged cluster

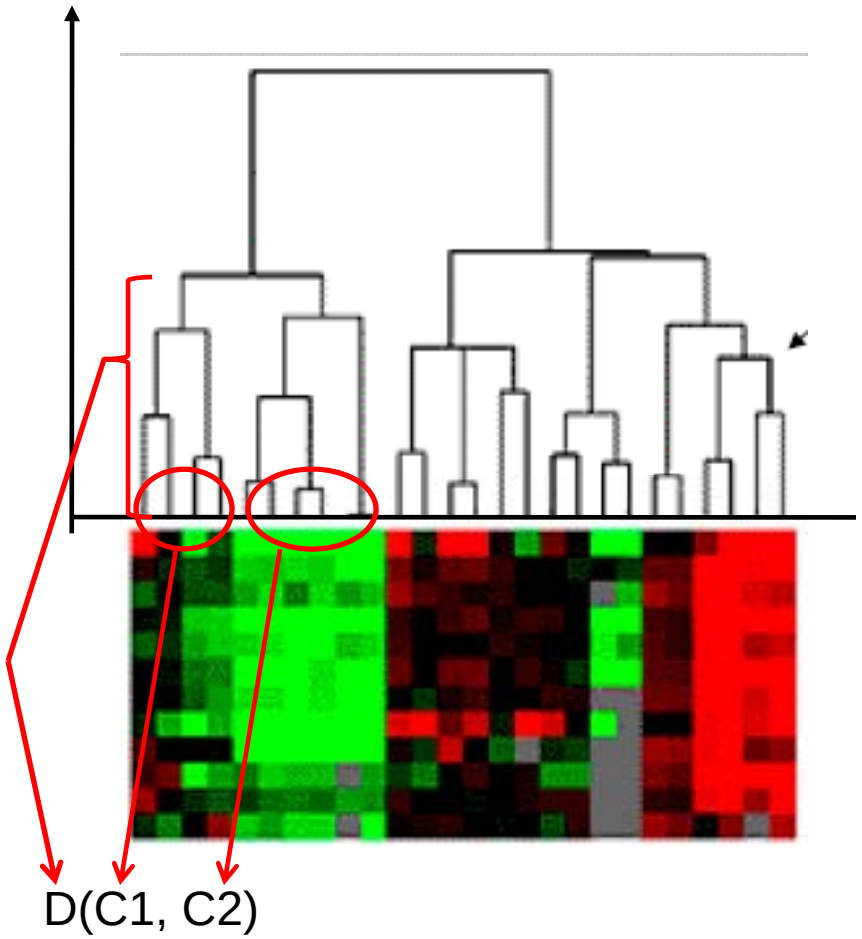
$$D(c_i, c_j) = \frac{1}{\left|c_i \cup c_j\right|\left(\left|c_i \cup c_j\right| - 1\right)} \sum_{\mathbf{x} \in (c_i \cup c_j)} \sum_{\mathbf{x}' \in (c_i \cup c_j): \mathbf{x}' \neq \mathbf{x}} D(\mathbf{x}, \mathbf{x}')$$

  - Averaged across all pairs between the original two clusters

$$D(c_i, c_j) = \frac{1}{\left|c_i\right|\left|c_j\right|} \sum_{\mathbf{x} \in c_i} \sum_{\mathbf{x}' \in c_j} D(\mathbf{x}, \mathbf{x}')$$

  - No clear difference in performance
- Compared to single link and complete link:
  - Computationally more expensive – $O(n_1 n_2)$
  - Achieves a compromise between single and complete link

HAC creates a Dendrogram



- Dendrogram draws the tree such that the height of a tree branch = the distance between the two merged clusters at that particular step
- The distances are always monotonically increasing
- This can provide some understanding about how many natural groups there are in the data
- A drastic height change indicates that we are merging two very different clusters together