Real Lecture 20: Hierarchical and K-means Clustering

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

Clustering

Goal:

- Identify groups of related points in a dataset
  - points in each group are more similar to each other than to points in other groups

Examples

- market segmentation
- recommendation systems
- Natural language processing

Hierarchical Clustering

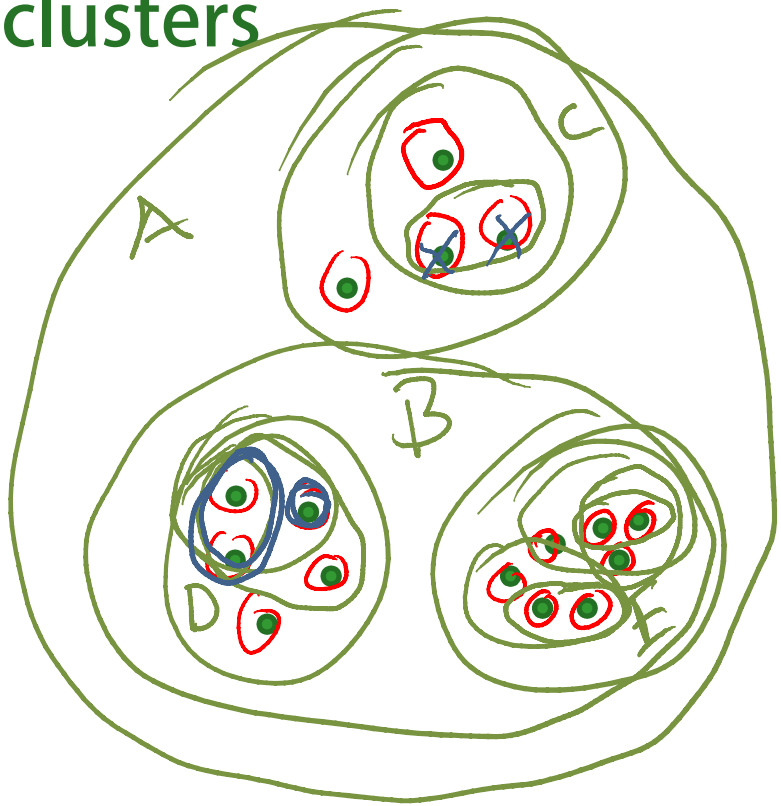
Results in a Tree of Clusters

Make every point a cluster

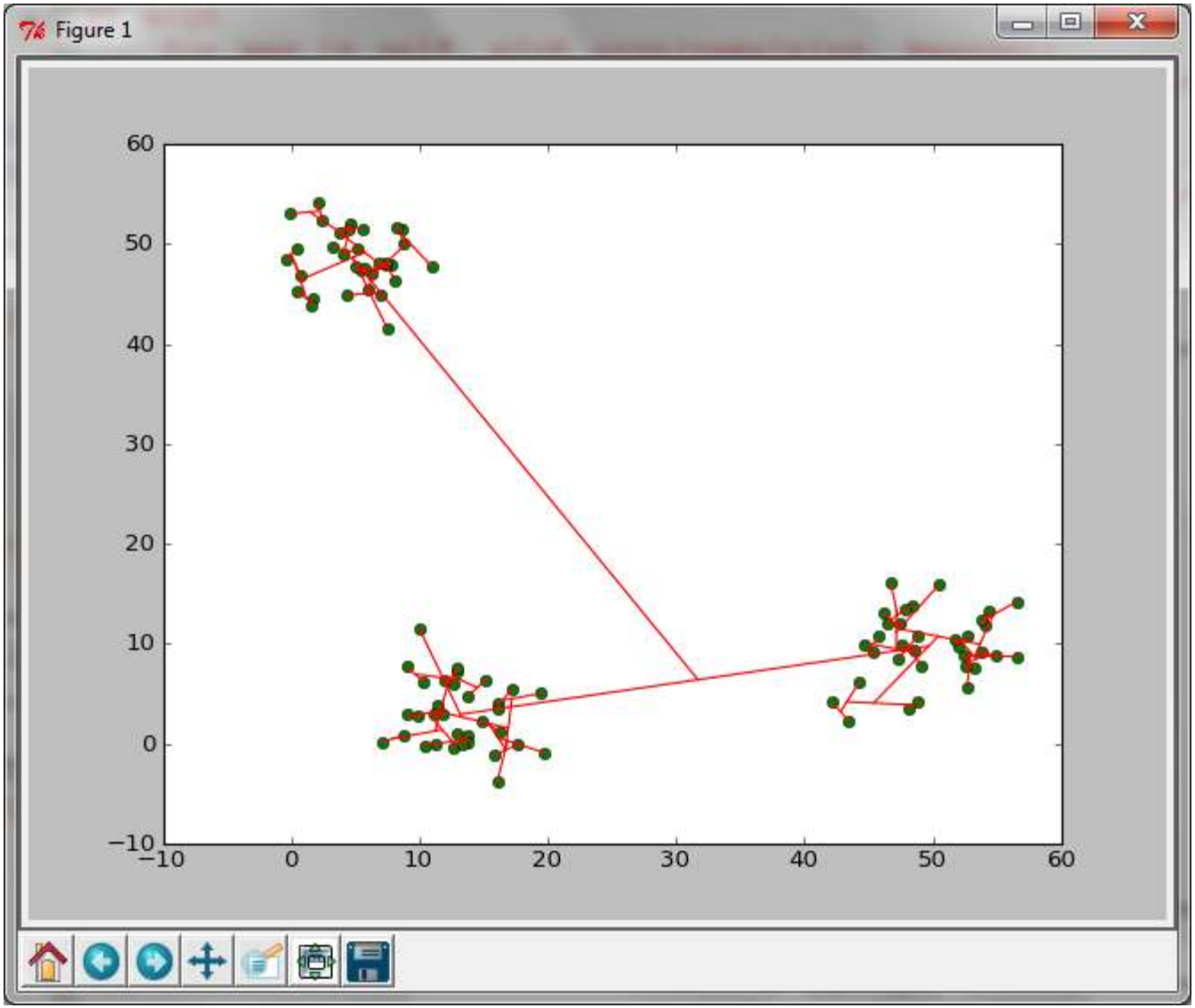
while(#clusters > 1):

pick the two *closest* clusters

merge them



Example printed from code



What do we mean by closest

Need to define distance between 2 points

- General question for any clustering algorithm
- Somewhat easy when data are points in $\mathbb{R}^n$
  - although even then there are many choices

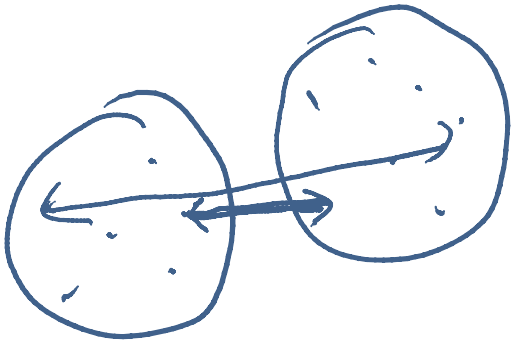
(age, wealth)

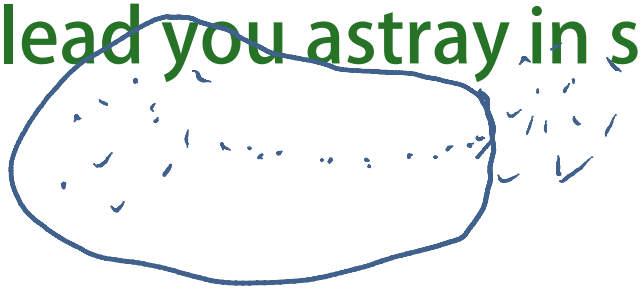
- How do we define the distance between …
  - pairs of purchase histories
  - pairs of documents
  - pairs of images
  - pairs of Facebook profiles
  - …
- This is the secret sauce for good clustering

What do we mean by closest

What is the distance between two clusters?

- this question is more specific to hierarchical clustering
- some choices:
  - distance between the means
  - distance between the closest points
  - distance between the farthest points
- each of these choices may lead you astray in some situations





Aside: List Comprehensions

Simple shorthand notation for building lists

syntax: [expr for var in list]

- Shorthand for

```
rv = []
for var in list:
    rv.append(expr)
```

Hierarchical Clustering

Pros:

- Helpful when number of clusters is unknown
- Can help uncover structure in the data

Cons:

- No theoretical guarantees

K-means clustering

Divide a set of points into exactly K clusters

- minimize the sum-of-squares of the distances to the mean for each cluster
- naïve algorithm is exponential
- iterative algorithm is fast and effective in practice
  - but it's not guaranteed to converge to the right result

k-means clustering algorithm

X =set of k points to represent the partitions

while(not converged):

-partition the points by assigning each data point to the closest point in X

-update the representative points X to correspond to the mean of each partition

What if one of the partitions is empty?

reassign that representative to a new random point.

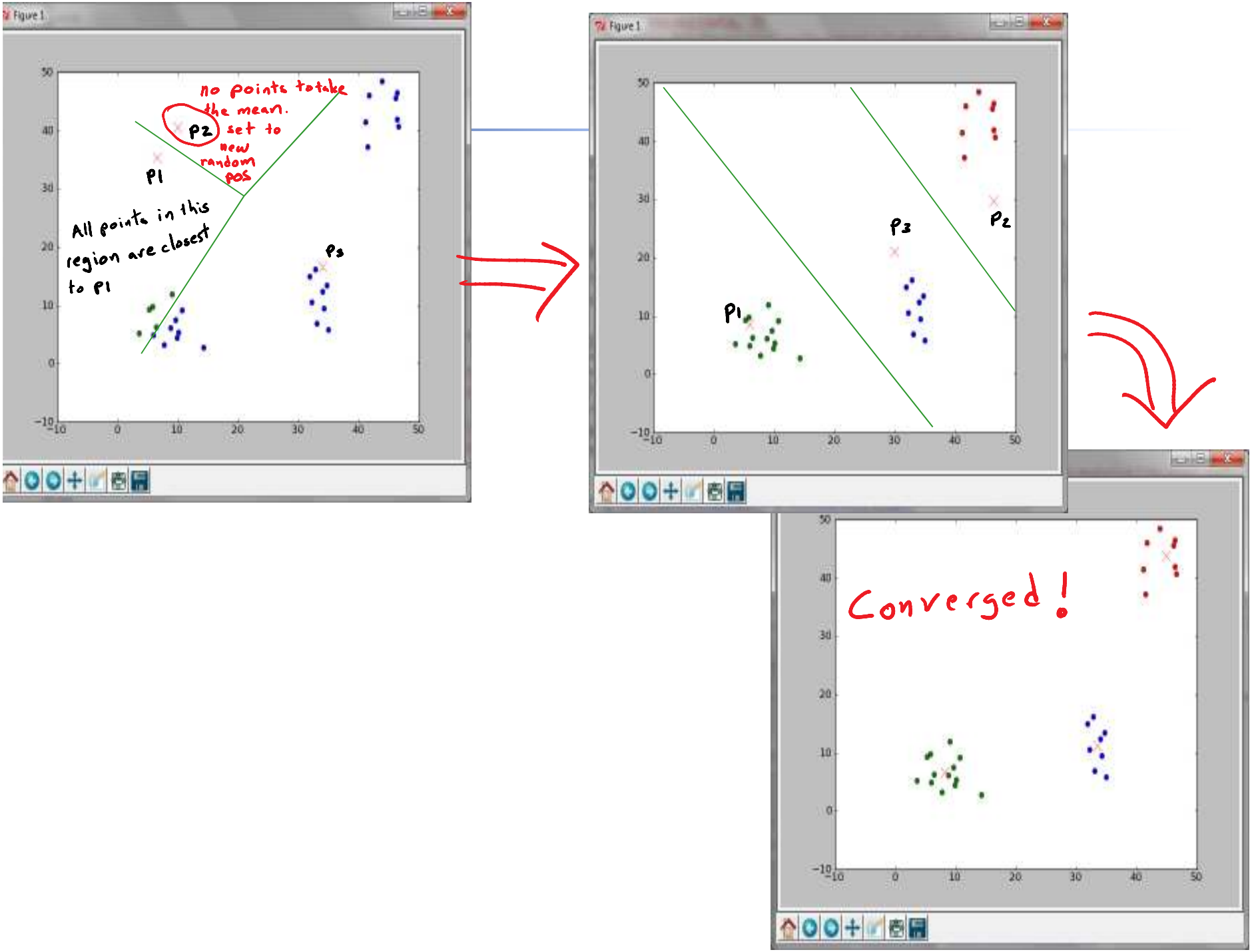


Figure 1
no points to take the mean. set to new random pos
P2
P1
All points in this region are closest to P1
P3
Figure 1
P1
P3
P2
Converged!