CS109 – Data Science

Verena Kaynig-Fittkau

vkaynig@seas.harvard.edu
staff@cs109.org

AWS Clusters

- New and updated instructions for Spark 1.5 are on Piazza:

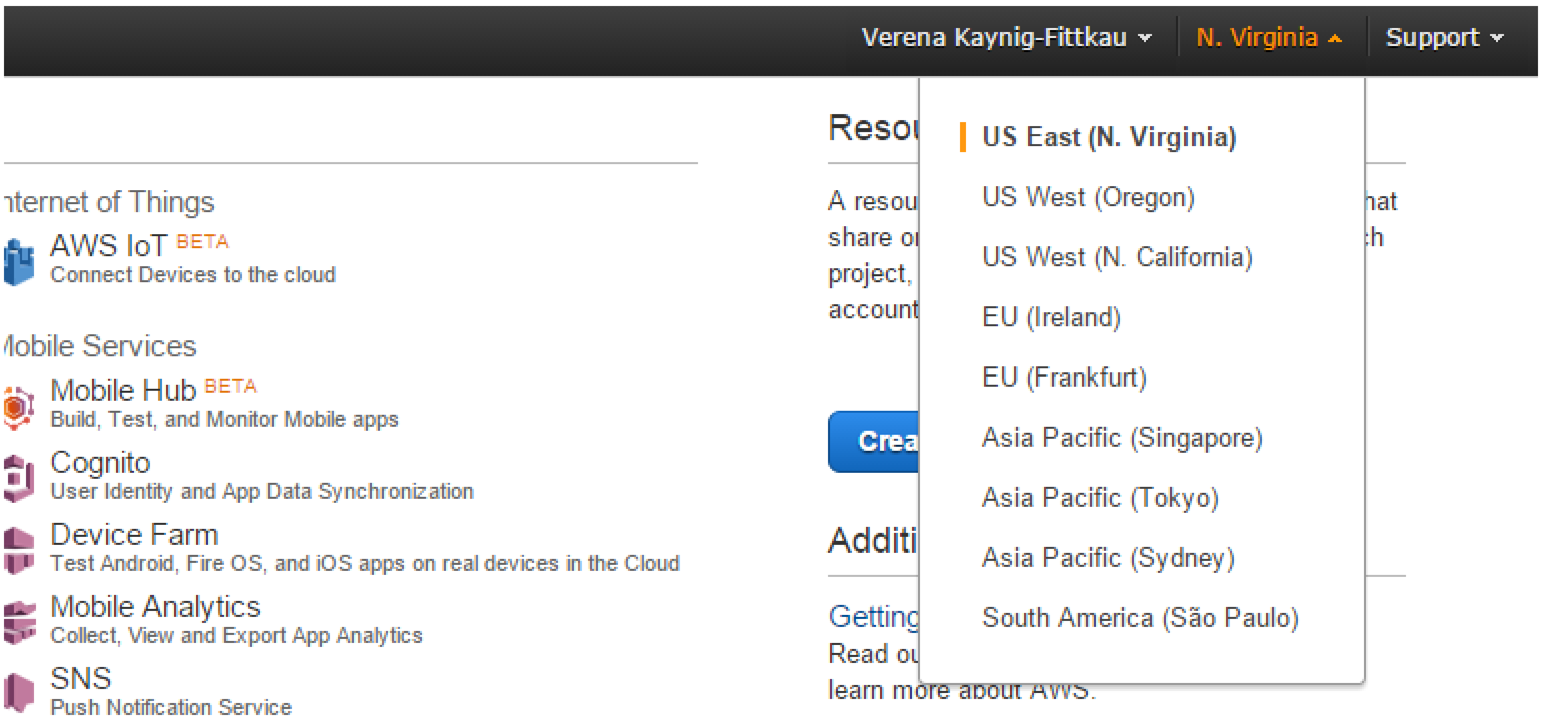
https://piazza.com/class/icf0cypdc3243c?cid=1369

Avoid Unnecessary Charges!

- Look at AWS console > Services > EMR
- There should be some terminated clusters there
- Check the region on the top right corner
- Make sure to change it to US East

https://piazza.com/class/icf0cypdc3243c?cid=1256

Region Setting in AWS



Announcements

- Final project
  - Team assignments have been posted to piazza
  - Make sure you are in a 3-4 person team
  - Try and date on the piazza thread
  - If you have problems write to staff@cs109.org

  - Project proposals are due on Thursday

  https://piazza.com/class/icf0cypdc3243c?cid=1317

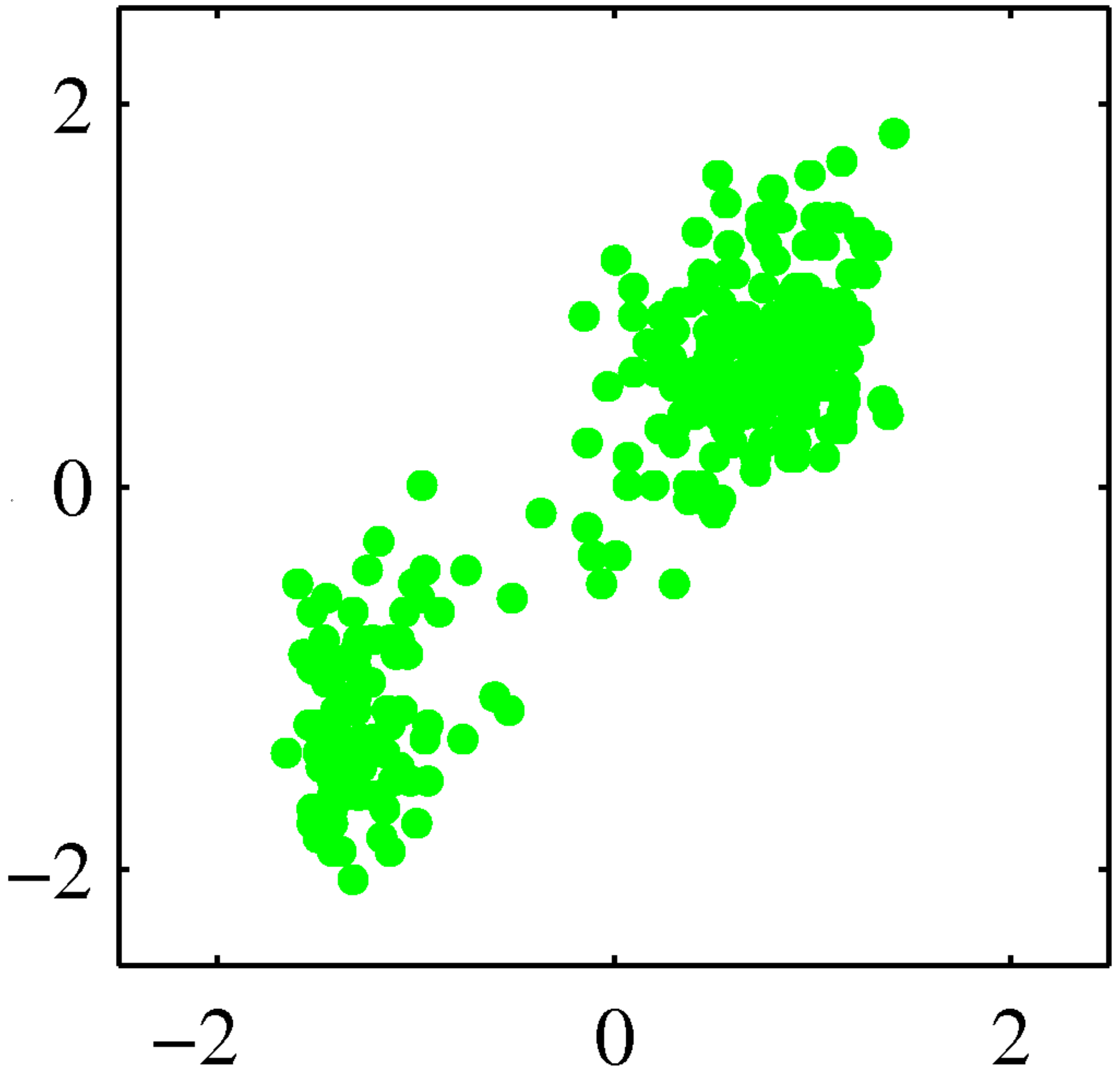
Final Project Proposal

- Submit just **one form per team**.
- Do it as **early as possible**!
- No project approval until you meet your TF

https://piazza.com/class/icf0cypdc3243c?cid=1317

Where before we had the y, or 'labels', now we don't have any of that and the task becomes much more difficult, because we cannot use the y/label to guide us defining the hyperplane.

Unsupervised Setting



Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006

Unsupervised Learning

- Find patterns in unlabeled data
- Sometimes used for a supervised setting in which labels are hard to get
- Can identify new patterns that you were not aware of.

Clustering Applications

In clustering, you're trying to find a pattern that you don't already know ahead of time.

- Google image search categories
- Author Clustering: http://academic.research.microsoft.com/VisualExplorer#1048044
- Opening a new location for a hospital, police station, etc.
- Outlier detection

In this scenario, some institutions throw out nearly all of their information and only keep the outlier data or only the significant events data.

Unsupervised Learning

- K-means
- Mean-shift
- Hierarchical Clustering

- Rand index, stability

Because we don't have y labels, this is how to evaluate how well the above methods performed.

K-means – Algorithm

Where before k = number of neighbors, here it's the number of random positions

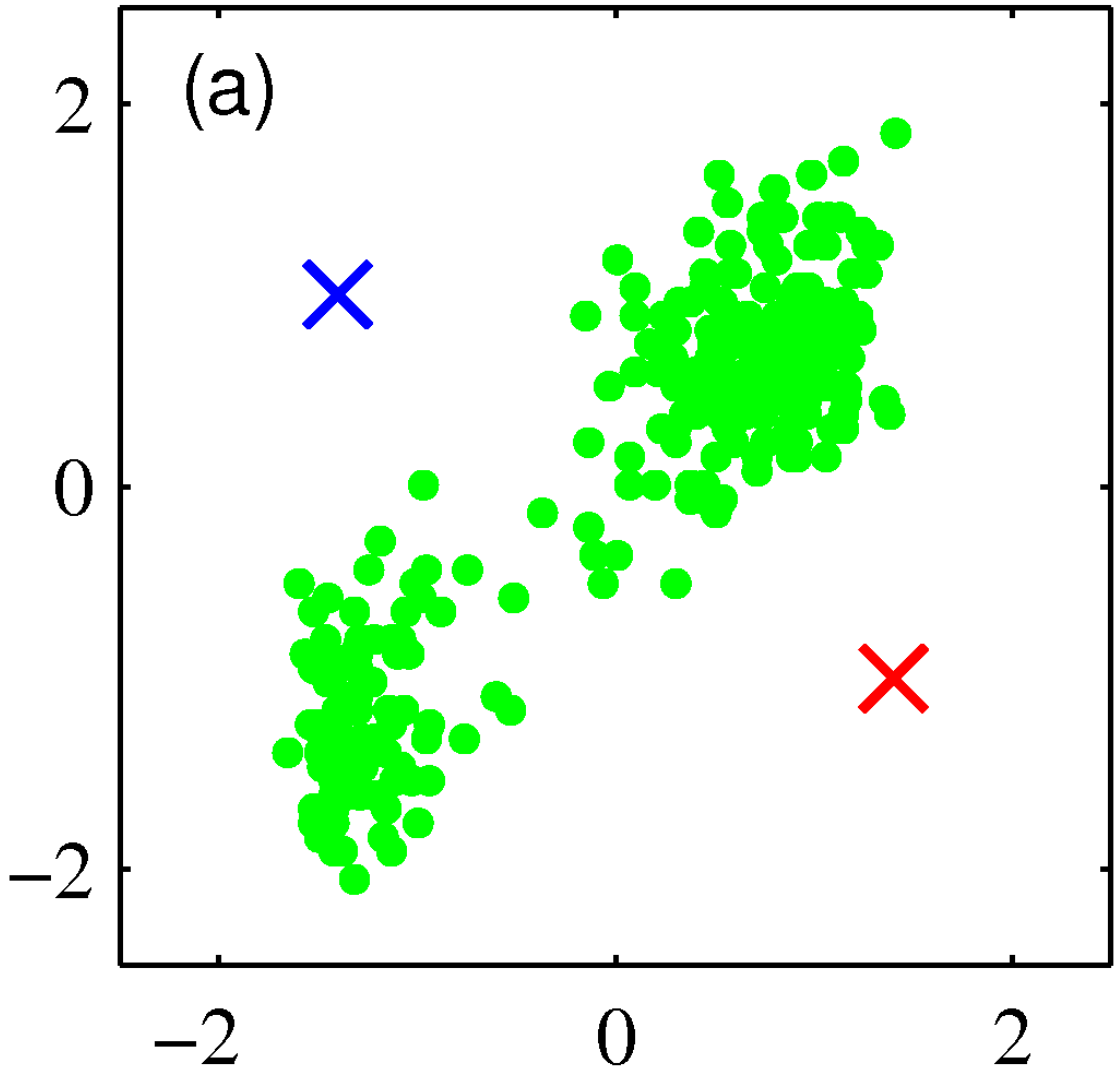
Again, k is just a number that we have to predefine.

- Initialization:
  - choose k random positions
  - assign cluster centers $\mu^{(j)}$ to these positions

K-means

Here, you randomly choose two points, and say this is now the center of the cluster.

We initialize this algorithm by choosing two arbitrary centers.



Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006

K-means

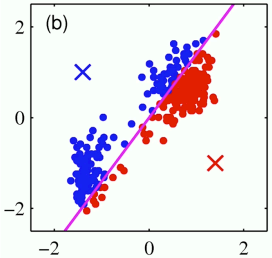
- Until Convergence:
  - Compute distances $||x^{(i)} - \mu^{(j)}||$

Now we compute the distances of all the data points we have, to these clusters and assign the points to the nearest cluster center.

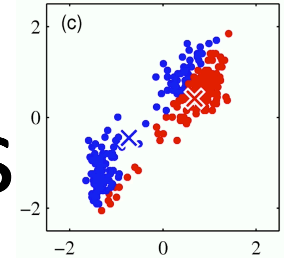
  - Assign points to nearest cluster center
  - Update Cluster centers:

$$\mu^{(j)} = \frac{1}{N_j} \sum_{x_i \in C_j} x_i$$

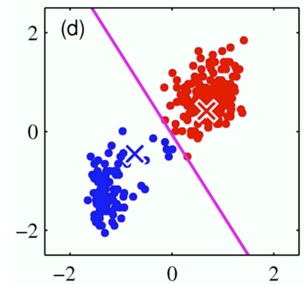
K-means



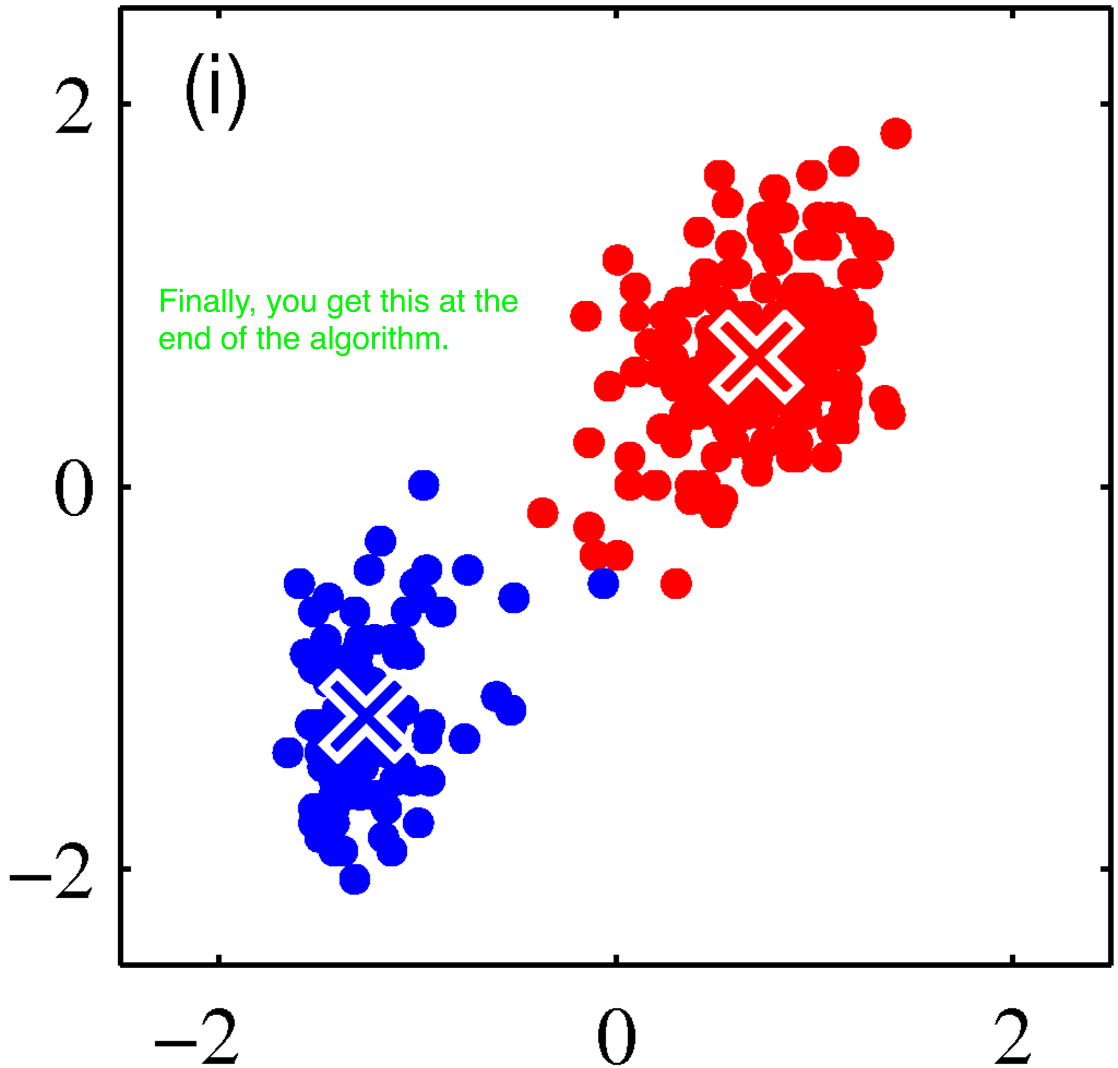
Before updating the cluster centers, the data points nearest to each of the randomly-assigned cluster centers get classified as such.



Here, the centers of the clusters get assigned to be in the middle of all the data points that were assigned to it.

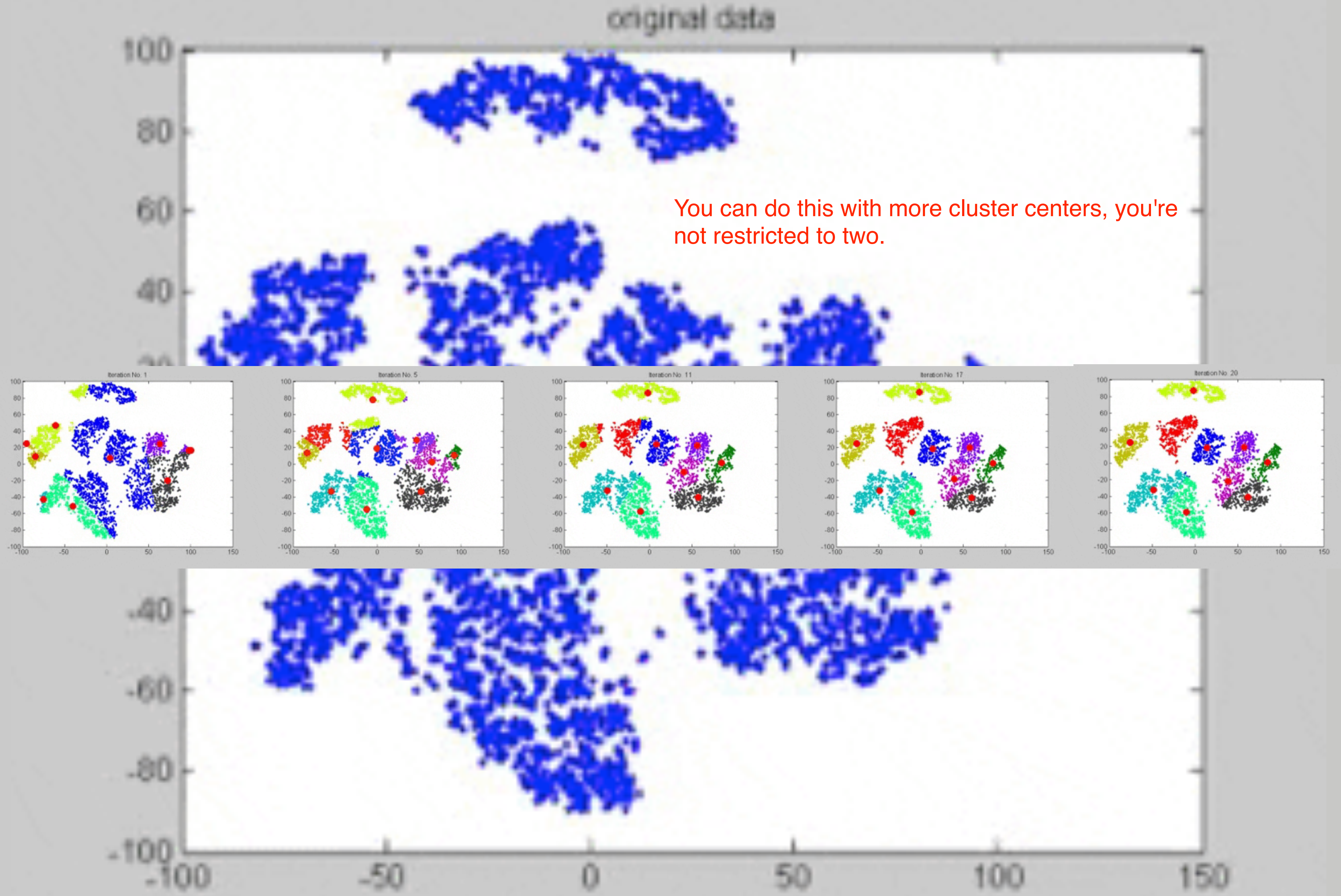


The whole thing starts again and the distances are calculated to determine the boundary

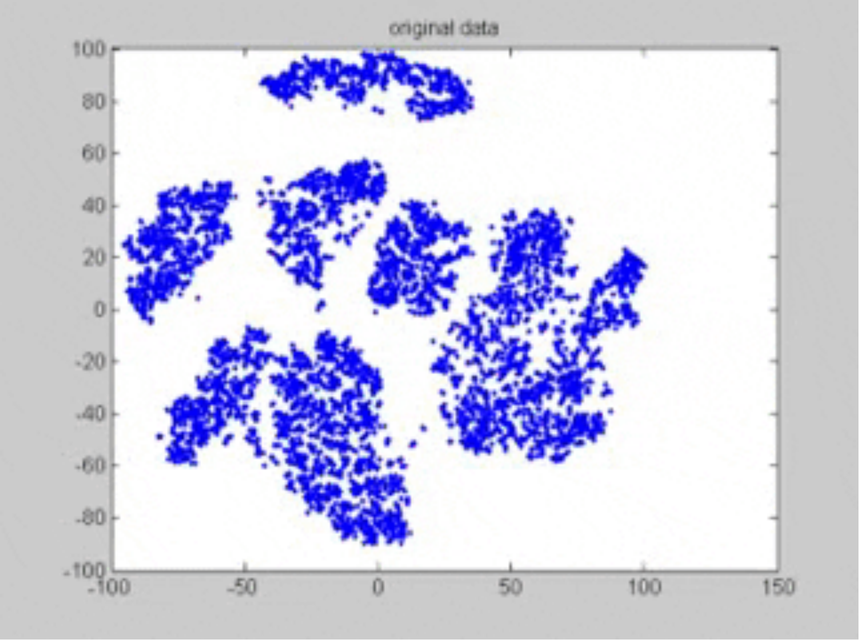


Finally, you get this at the end of the algorithm.

Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006



You can do this with more cluster centers, you're not restricted to two.



original data
100
80
60
40
20
0
-20
-40
-60
-80
-100
-100
-50
0
50
100
150

K-means Example

R

G

B

K-means Example

K-means Example

Despite both images using k = 10 (random) positions, they appear different.

The difference is due to the initializing with a random position for the cluster centers.

K-means Summary

How do you know when you've converged? Set a value for Epsilon, and once your cluster centers have moved less than epsilon, then you're done.

- Guaranteed to converge
- Result depends on initialization

This makes it hard to decide which pattern is the one you ought to be choosing/looking for.

- Number of clusters is important
- Sensitive to outliers
  – Use median instead of mean for updates

You can mediate some of that sensitivity to outliers by using median instead of the mean during the update phase from the centers of your clusters.

Initialization Methods

- Random Positions

Instead of going into the feature space and picking two positions, this says the points have to belong to a cluster so it makes sense to pick random data points as your initial cluster centers.

- Random data points as Centers
- Random Cluster assignment to data points

First do the random cluster assignment, then do the update step, and see where the centers end up.

- Start several times

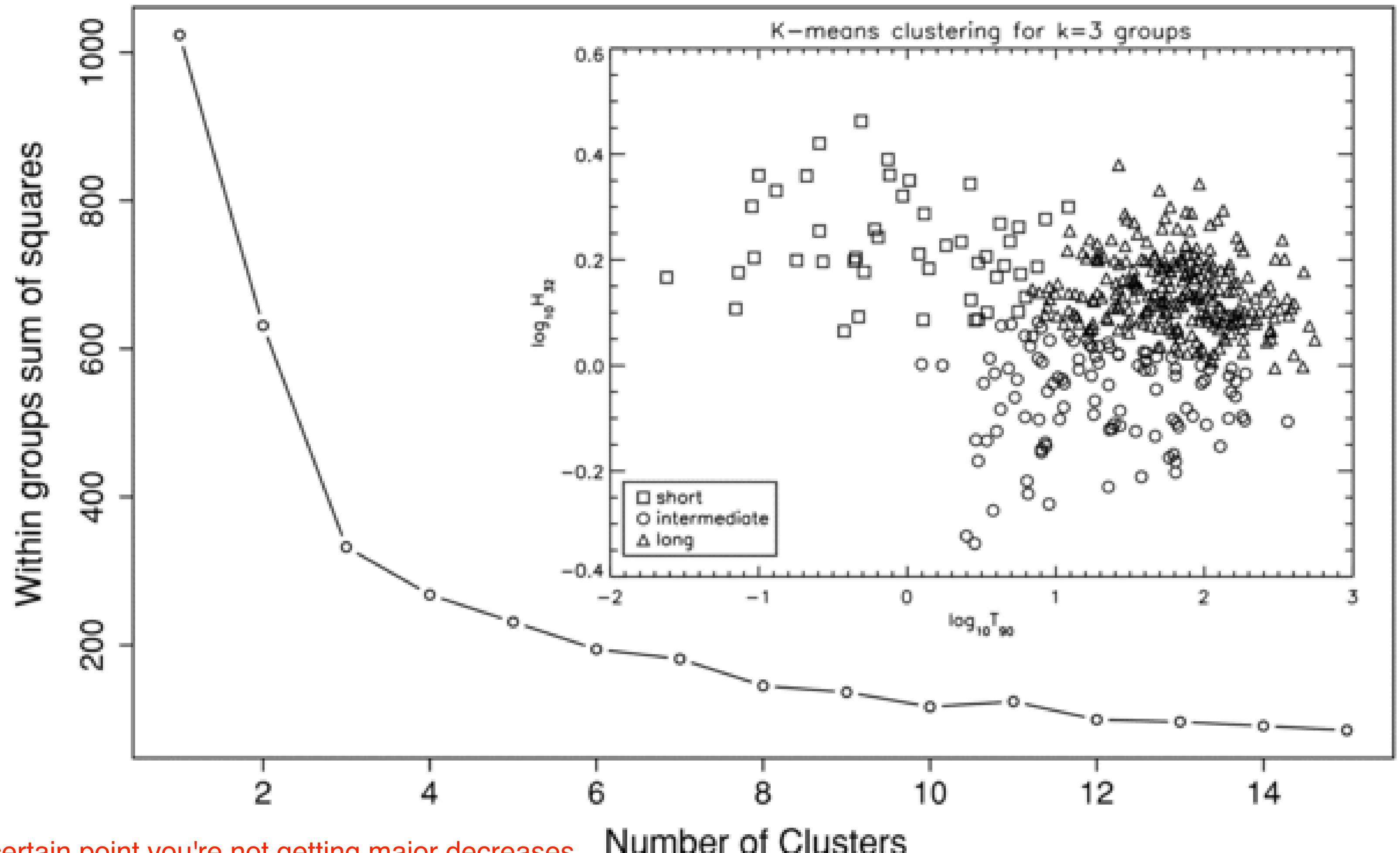
With clustering, there is this idea of stability. If you do 100 runs of k=10 and you get a solution that posp up 90 times and another that comes up 10 times, you'd want to go with the 90x solution since it's more strongly held to by the data.

How to find K

- Extreme cases:
  - K=1
  - K=N
- Choose K such that increasing it does not model the data much better.

“Knee” or “Elbow” method

Here, within groups sum of squares measures the distance from the data points to their cluster group's center.



So, at a certain point you're not getting major decreases
in sum of squares for the number of K/random cluster centers.

http://m.iopscience.iop.org/article/10.1088/0004-637X/725/2/1955;jsessionid=9B96FCAF94036AA723BA5F9C78FB33D6.c1.iopscience.cld.iop.org

Cross Validation

If you want to be able to generalize to new data, then cross validation will help you pick the best k/number of cluster centers to avoid overfitting.

- Use this if you want to apply your clustering solution to new unseen data

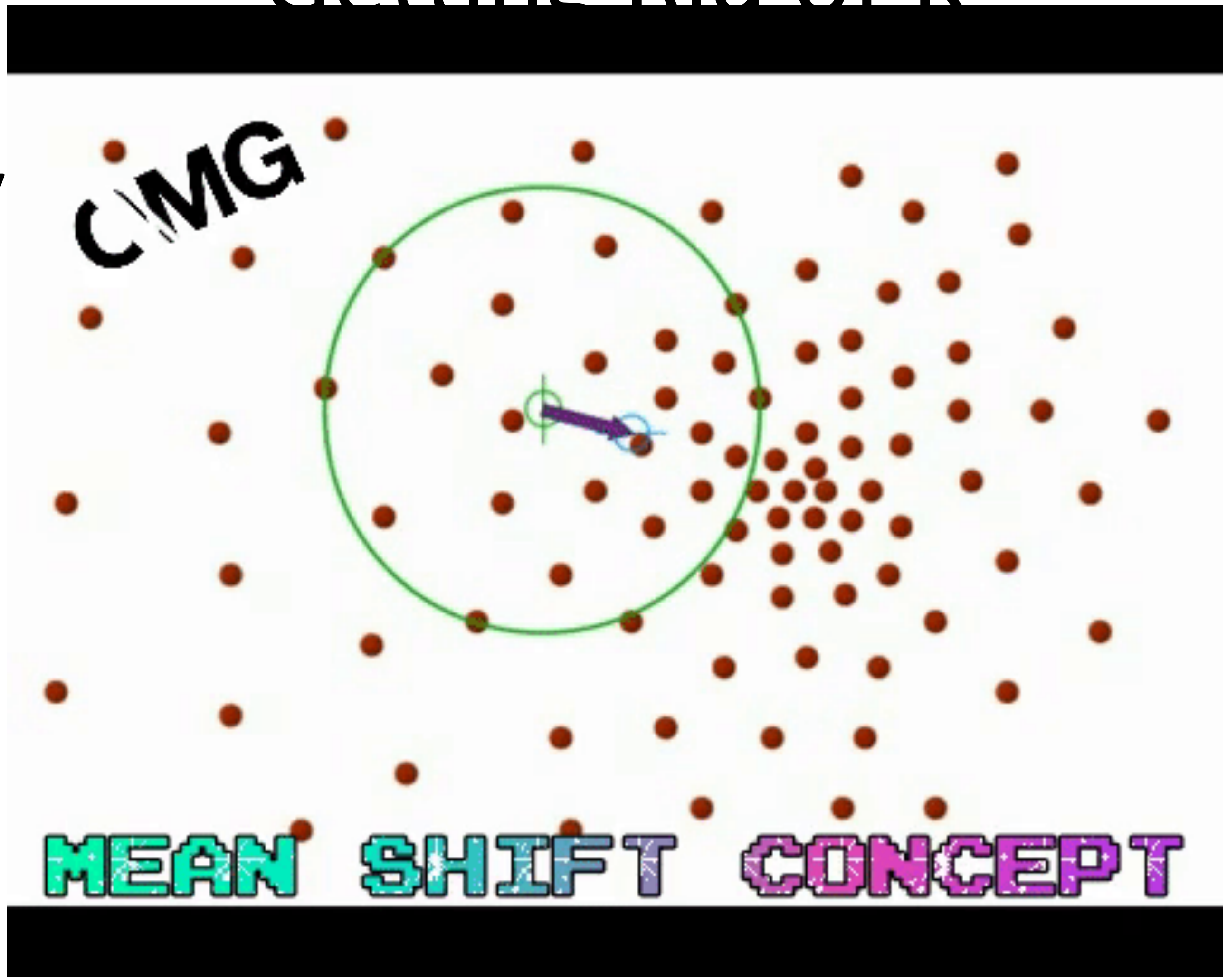
- Partition data into n folds
- Cluster on n-1 folds
- Compute sum of squared distances to centroids for validation set

Interad of using the same data to compute the sum of squares, you have training data that you use to solve your kmeans and then you have the validation data coming in where you compute all the distances to the groups that you just identified.

Cross-validation gives you confidence about generalization or about the new data coming in or that you haven't seen yet.

Getting Rid of K

- Hav
- Can



For each point in the dataset, you specify the window around it. And that window is
the same size for all points. Now this point is essentially becomes a cluster center
and you look for all the points that are in a specific distance to that point, inside this window.

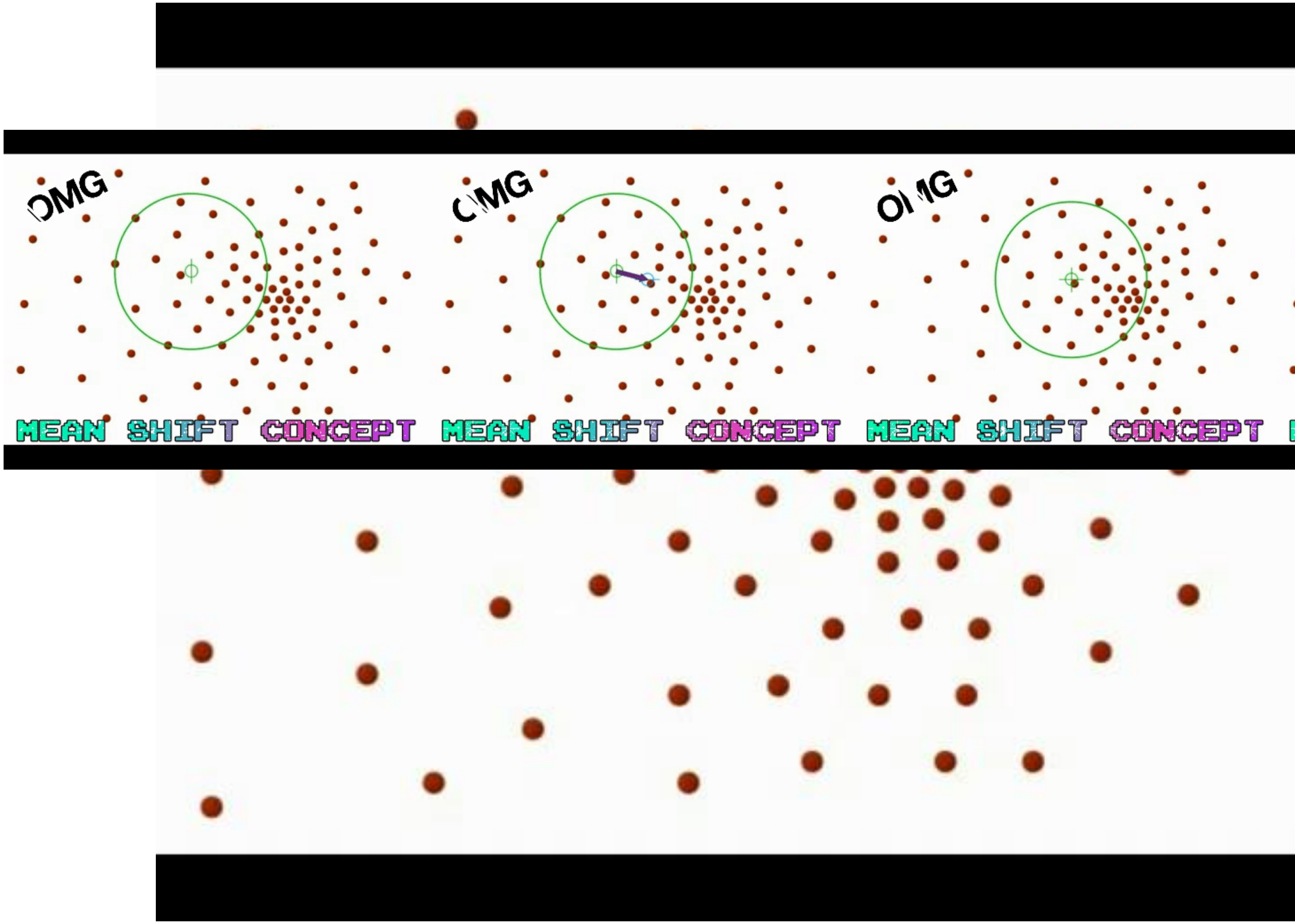
Mean Shift

1. Put a window around each point
2. Compute mean of points in the frame.
3. Shift the window to the mean
4. Repeat until convergence

https://www.youtube.com/watch?v=kmaQAsotT9s

Mean Shift

You end up doing this for each individual data point, and shift, shift, shift.



http://www.youtube.com/watch?v=kmaQAsotT9s

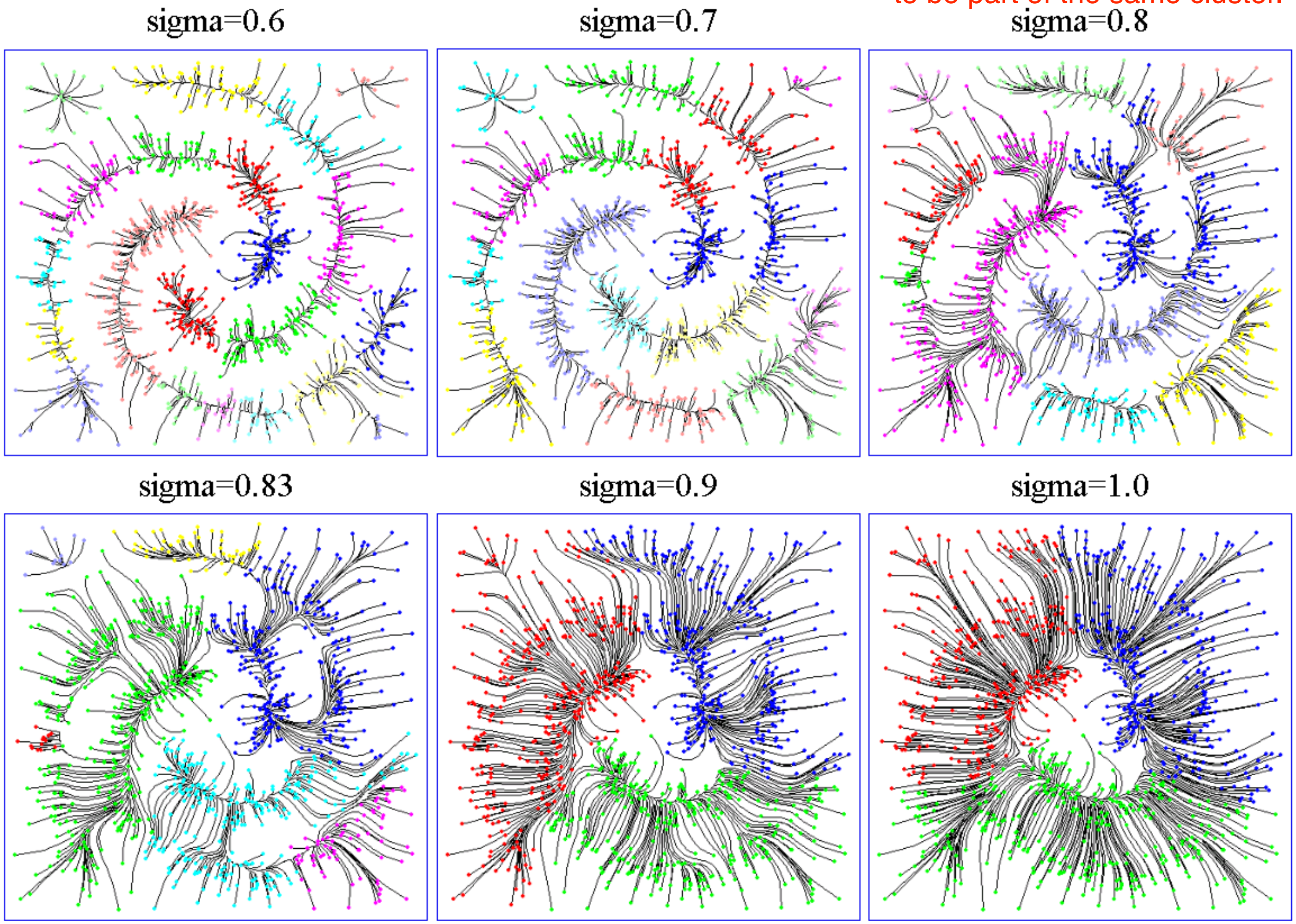
When you have a gradient of points, where they're more clustered/denser, the window will shift towards that density of points. All the points eill end up with the same center in the end because of this, and that center is your cluster.

This is a demonstration for different window sizes and the lines here show the path that the window took in each iteration.

Here the number of clusters hasn't been defined, just the size of the window.

Mean Shift

All the points where the window shifted to the window shifted to the same place are assumed to be part of the same cluster.



The size of the window has a lot to do with the number of clusters that're being found.

Fischer et al., “Clustering with the Connectivity Kernel“, NIPS (2003)

Mean Shift Summary

- Does not need to know number of clusters
- Can handle arbitrary shaped clusters
- Robust to initialization
- Needs bandwidth parameter (window size)
- Computationally expensive

The reason why is because you have to do it for each and every single data point.

Calculating the mean for each of the data points within the window is computationally expensive.

- Very good article:

http://saravananthirumuruganathan.wordpress.com/2010/04/01/introduction-to-mean-shift-algorithm/

Multi-feature object trajectory clustering for video analysis

Nadeem Anjum   Andrea Cavallaro

Parameters parameters

Again, there is no 'free lunch' paramaters, meaning one of them has to be set for the model to function.

- For K means we need K and result depends on initialization
- For mean shift we need the window size and a lot of computation

- Hierarchical Clustering keeps a history of all possible cluster assignments

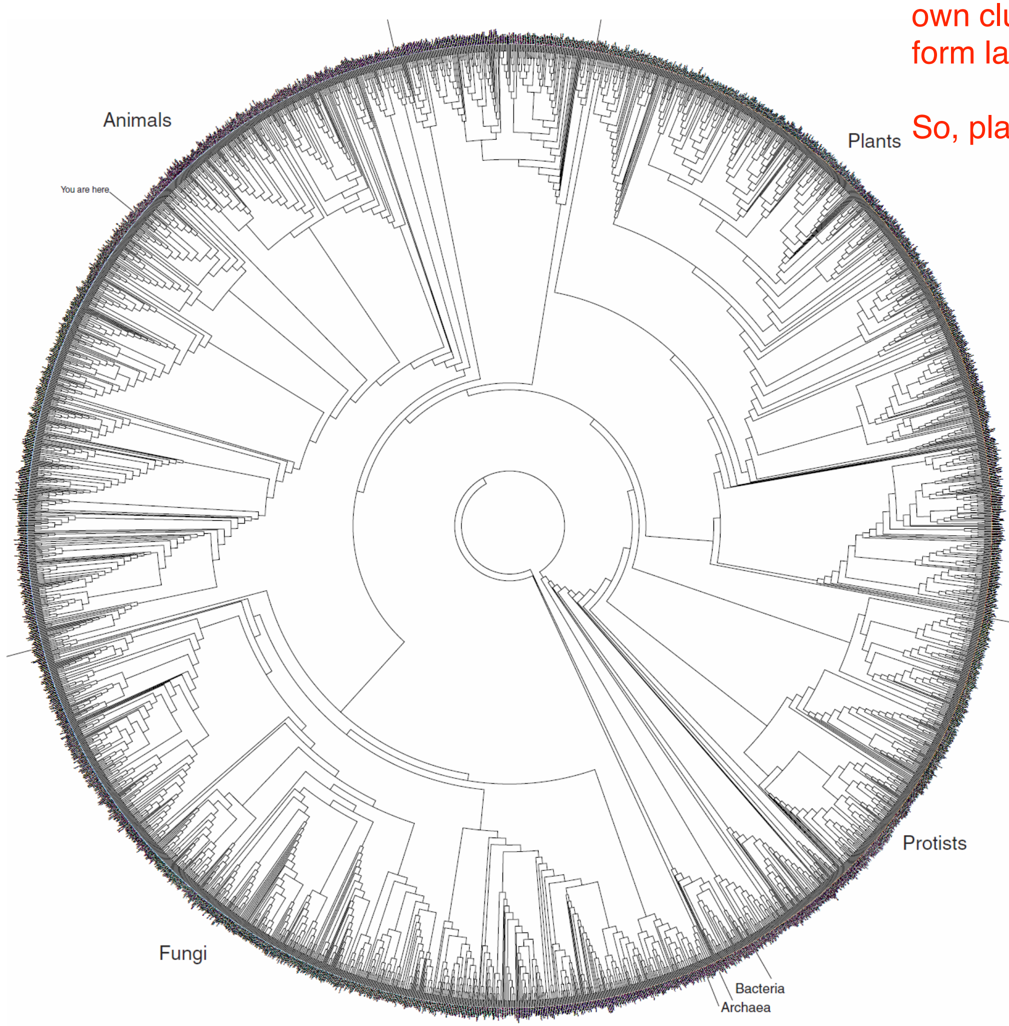
This approach uses neither k nor windows to function.

This is a listing of organisms and how similar they are in terms of evolvement.

Tree of Life

On the lowest level, every organism is its own cluster. Then the groupings start to form larger and larger clusters of points.

So, plants are in one cluster then fungi, etc.

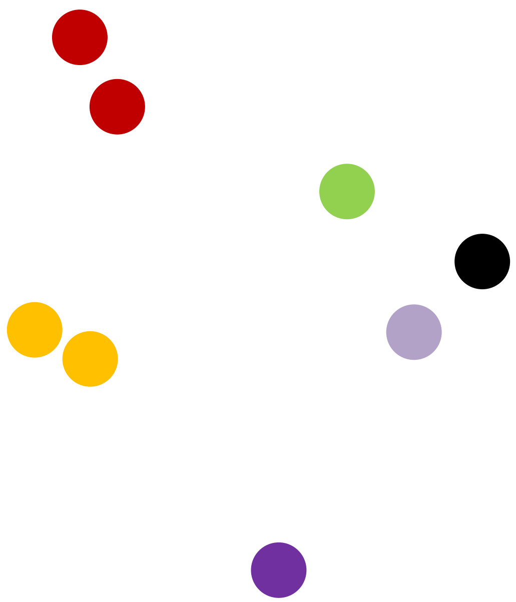


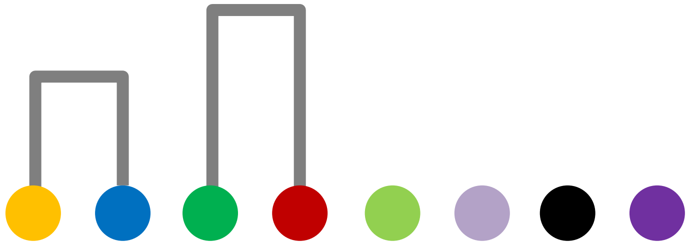
http://www.zo.utexas.edu/faculty/antisense/DownloadfilesToL.html

Hierarchical Clustering

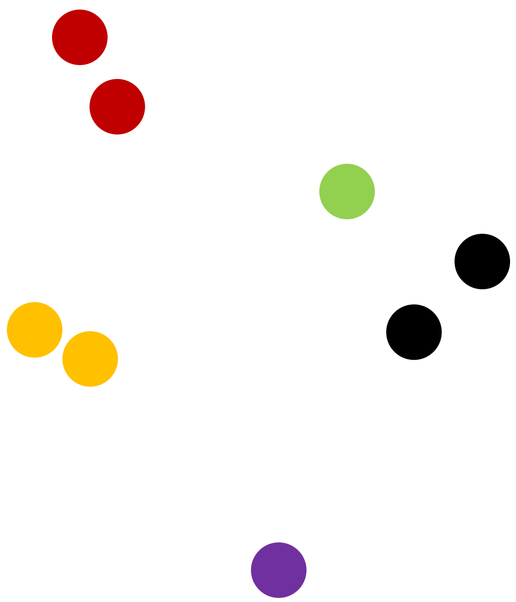
Hierarchical Clustering

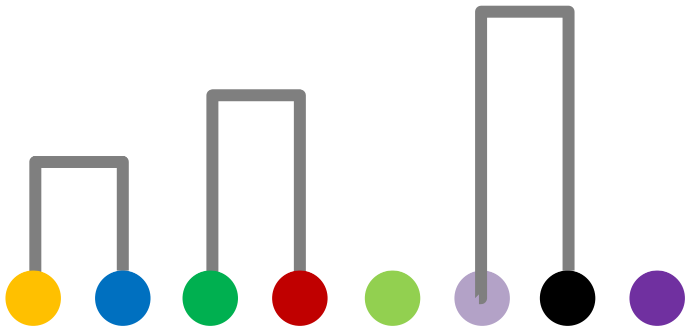
Hierarchical Clustering



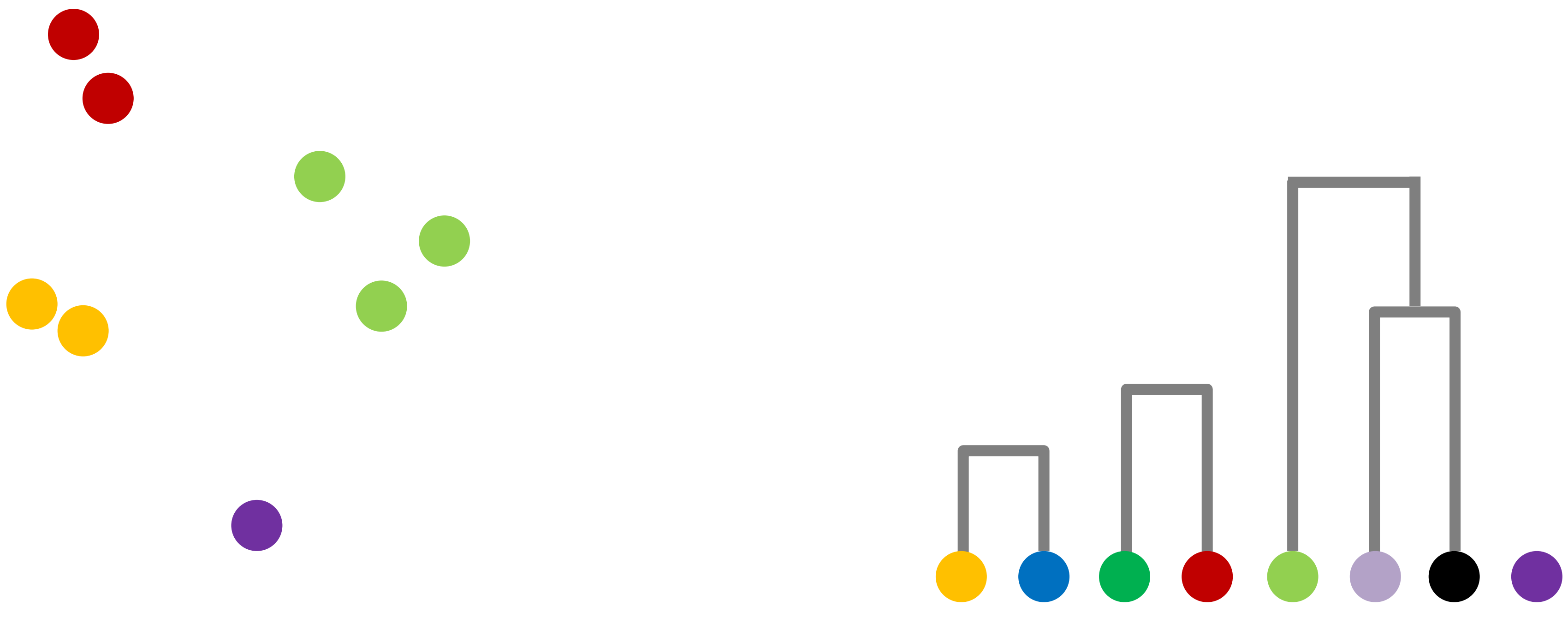


Hierarchical Clustering

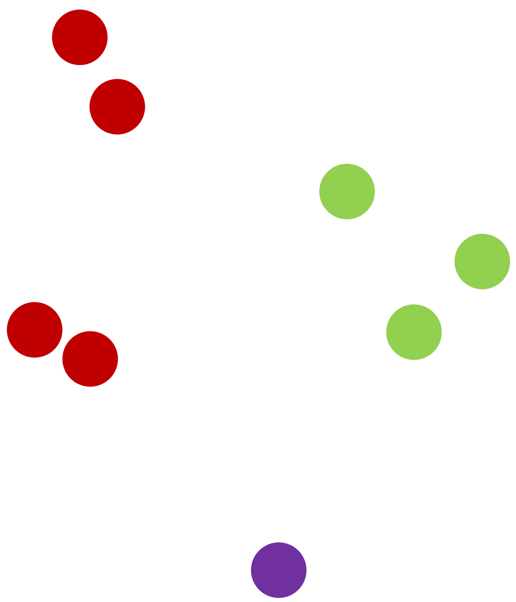


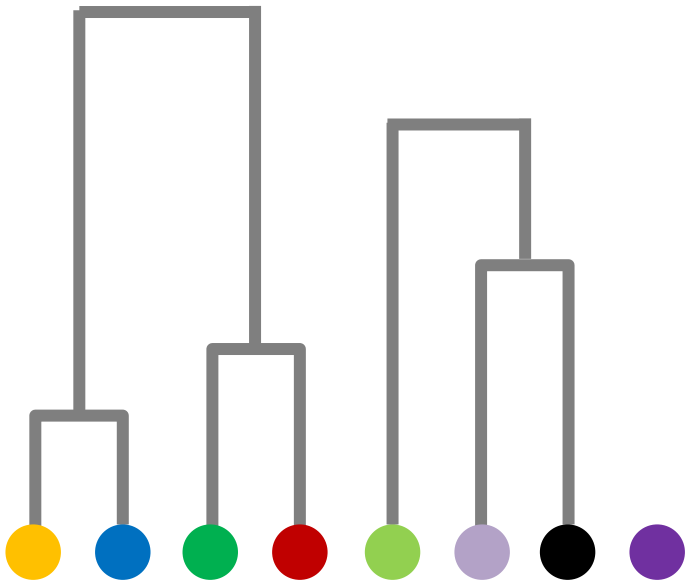


Hierarchical Clustering

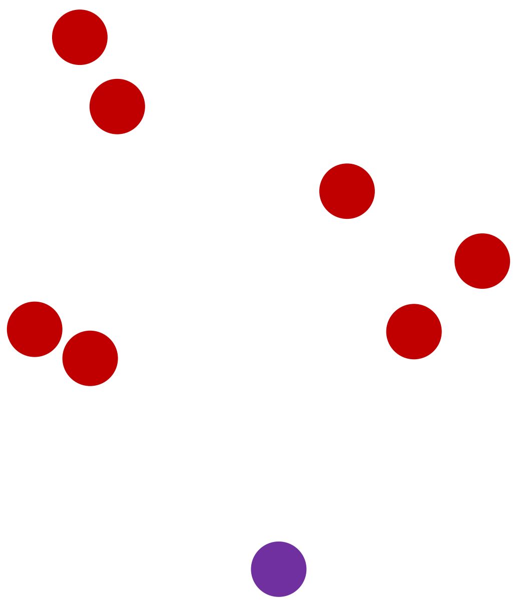


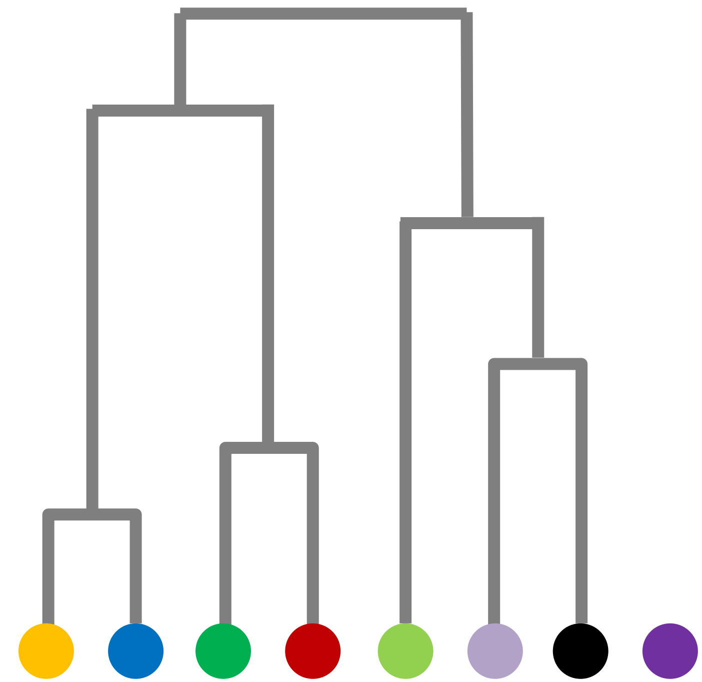
Hierarchical Clustering



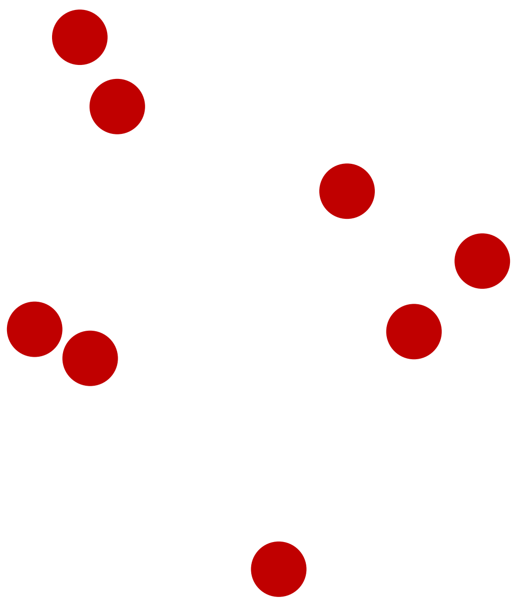


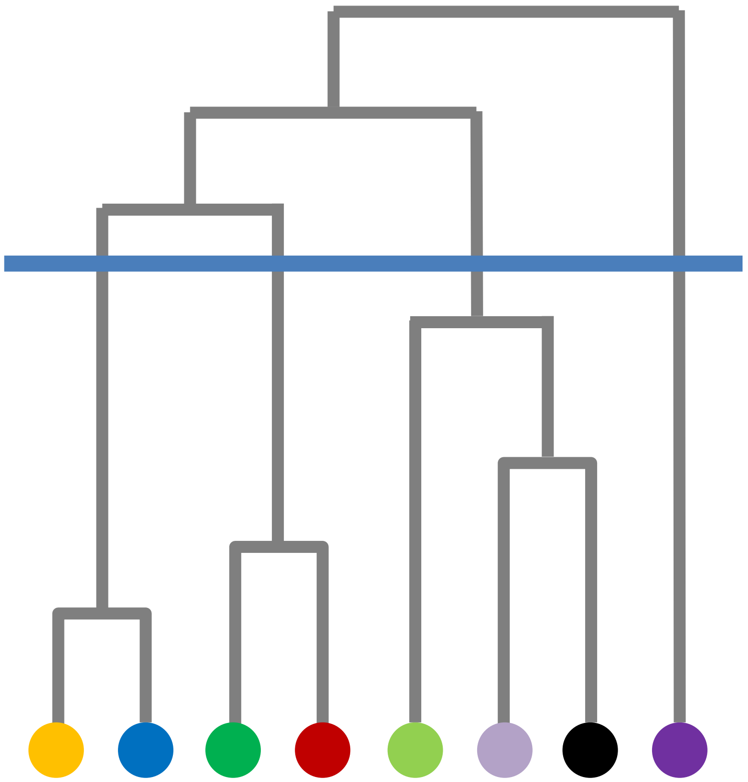
Hierarchical Clustering



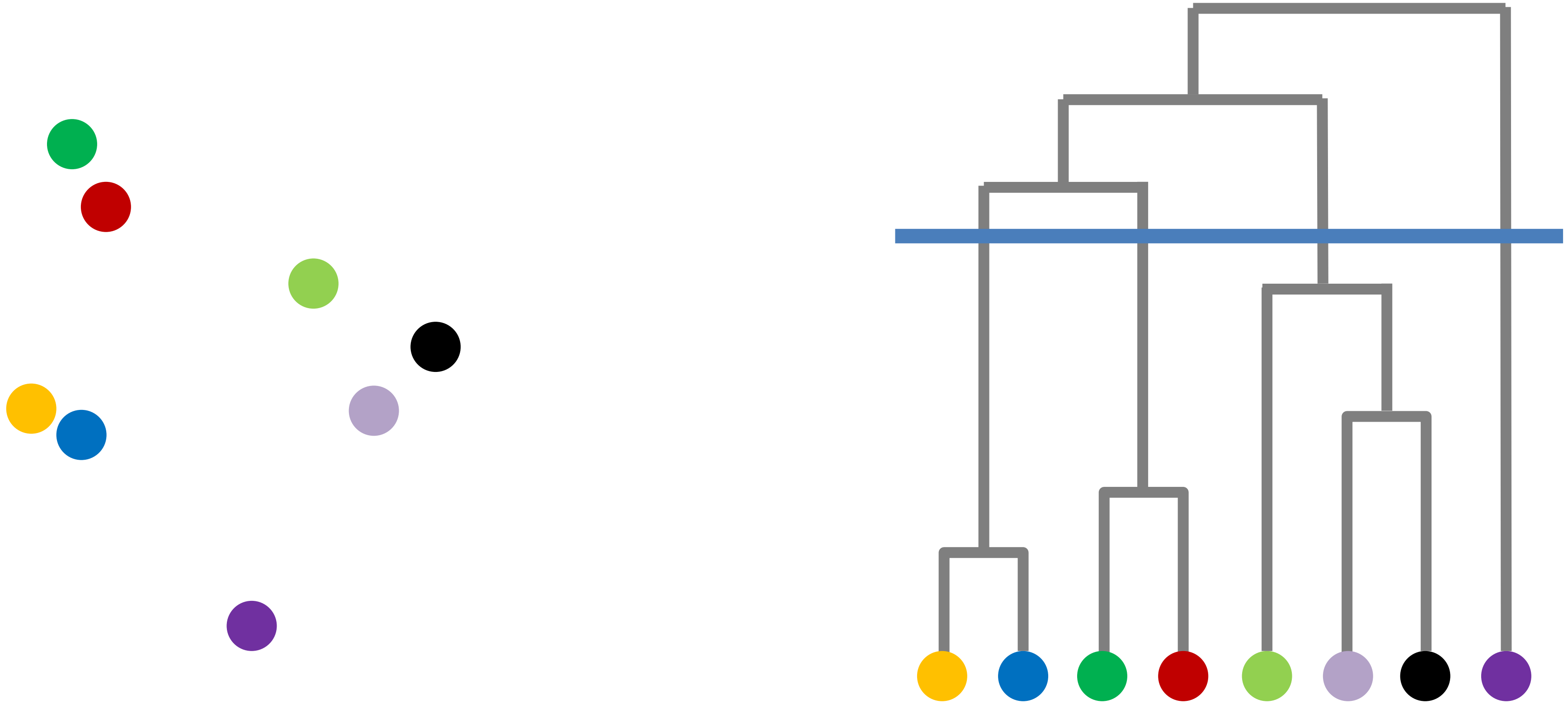


Hierarchical Clustering





Hierarchical Clustering

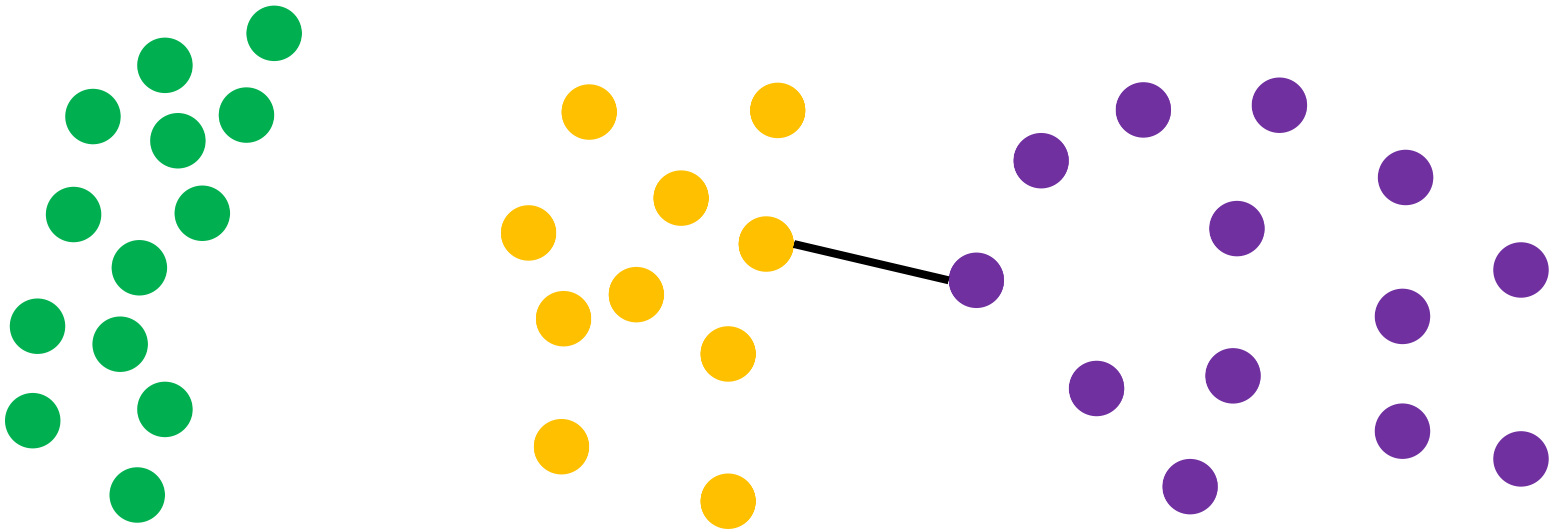


Hierarchical Clustering

- Produces complete structure
- No predefined number of clusters

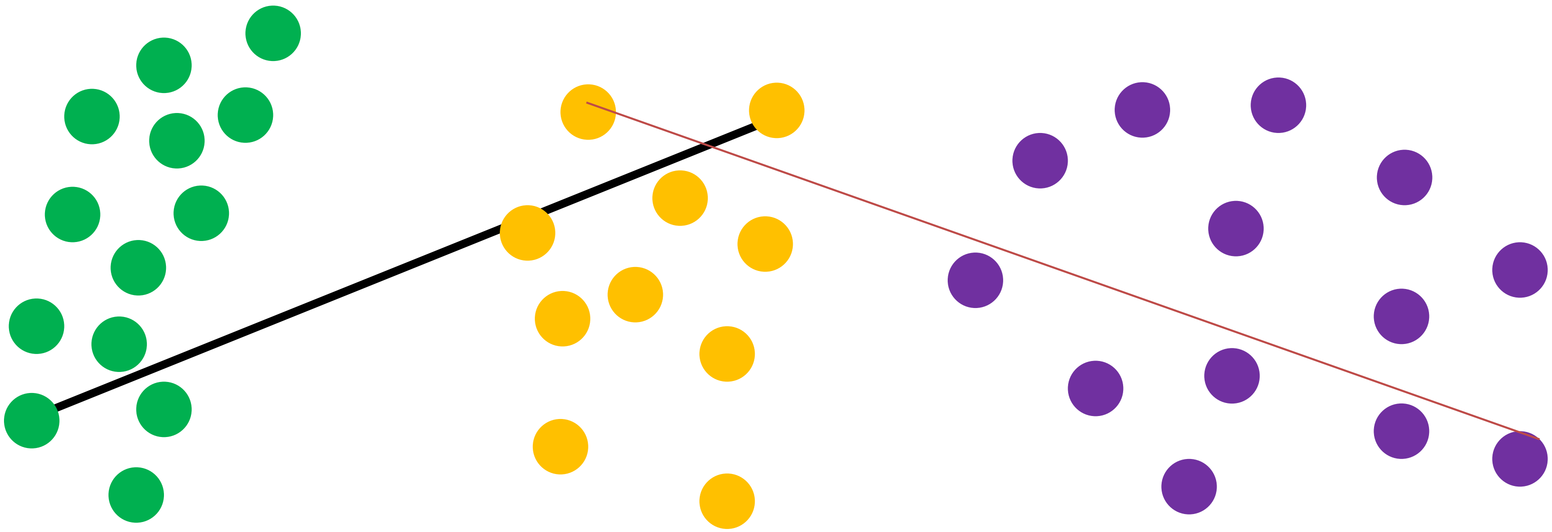
- Similarity between clusters:
  - single-linkage: $\min\{d(x,y) : x \in \mathcal{A}, y \in \mathcal{B}\}$
  - complete-linkage: $\max\{d(x,y) : x \in \mathcal{A}, y \in \mathcal{B}\}$
  - average linkage: $\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x,y)$

Single Linkage



$$\min\{d(x,y) : x \in \mathcal{A}, y \in \mathcal{B}\}$$

Complete Linkage

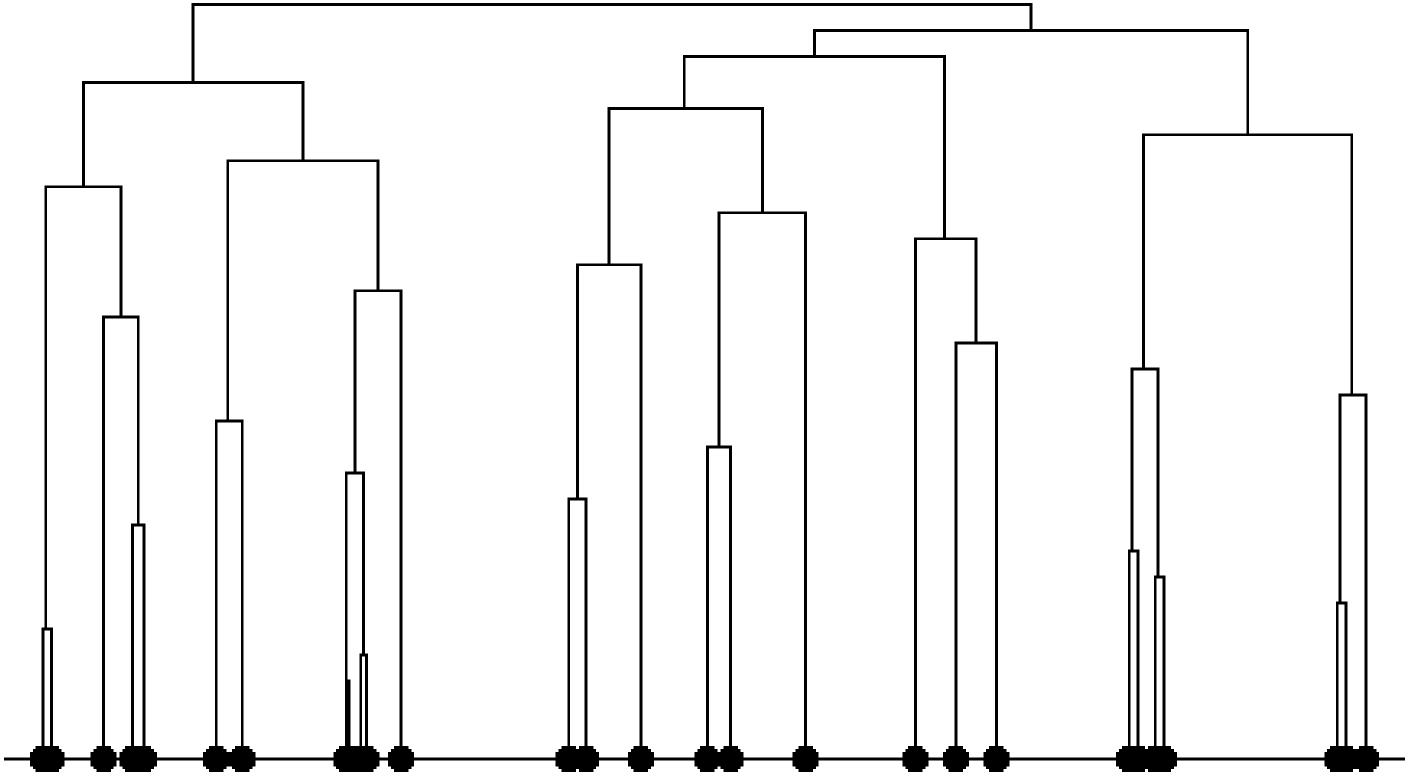


$$\max\{d(x,y) : x \in \mathcal{A}, y \in \mathcal{B}\}$$

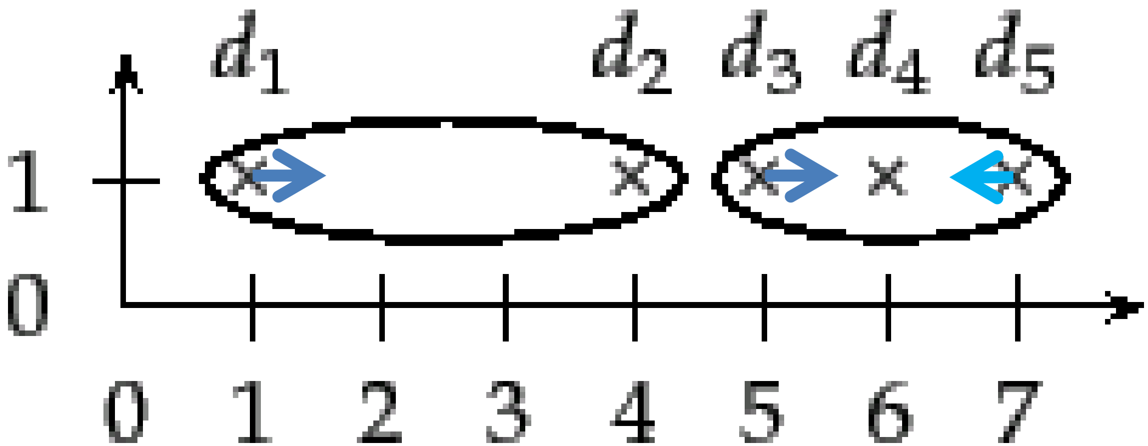
Linkage Matters

- Single linkage: tendency to form long chains
- Complete linkage: Sensitive to outliers
- Average-link: Trying to compromise between the two

Chaining Phenomenon



Outlier Sensitivity



+ 2*epsilon

- 1*epsilon

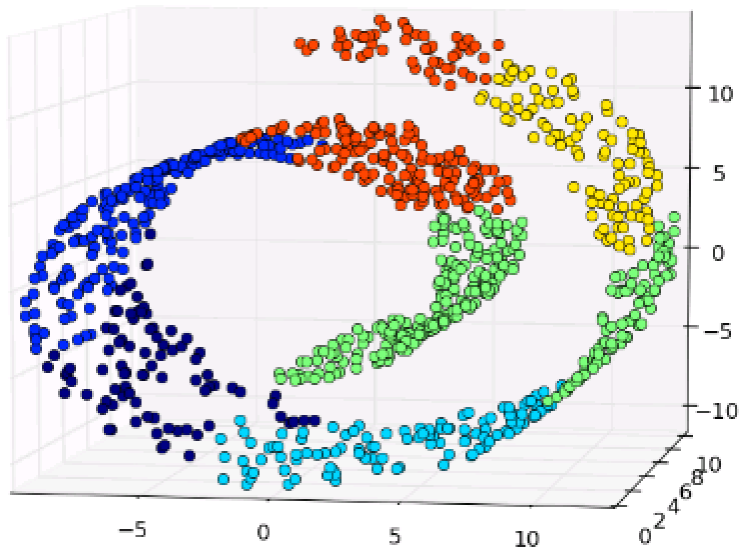
http://nlp.stanford.edu/IR-book/html/htmledition/img1569.png

Efficient Hierarchical Graph-Based Video Segmentation

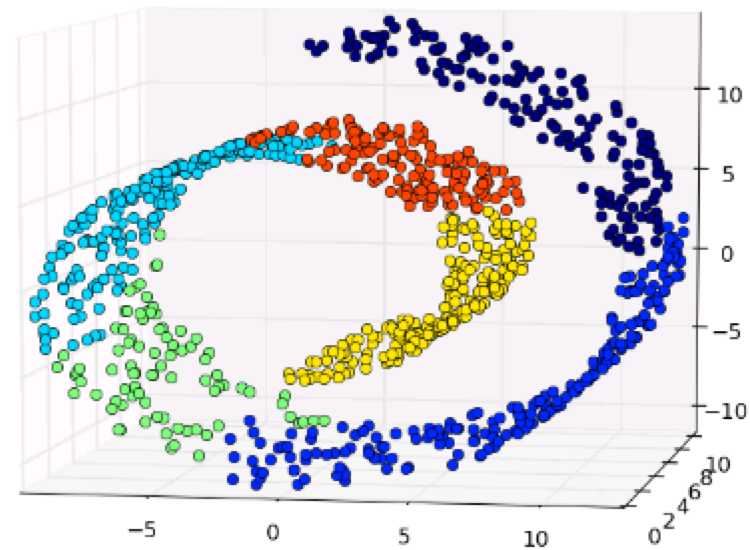
Matthias Grundmann[1,2], Vivek Kwatra[2], Mei Han[2] and Irfan Essa[1]
[1]Georgia Tech [2]Google Research

IEEE CVPR, San Francisco, USA, June 2010

Swiss Role Problem



without connectivity
constraints



with connectivity
constraints

only adjacent clusters can be merged together

http://scikit-learn.sourceforge.net/modules/clustering.html

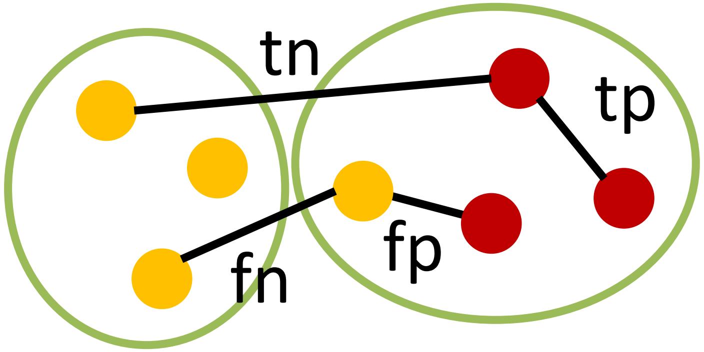
Evaluation Criteria

- Based on expert knowledge
- Debatable for real data
- Hidden Unknown structures could be present
- Do we even want to just reproduce known structure?

Rand Index

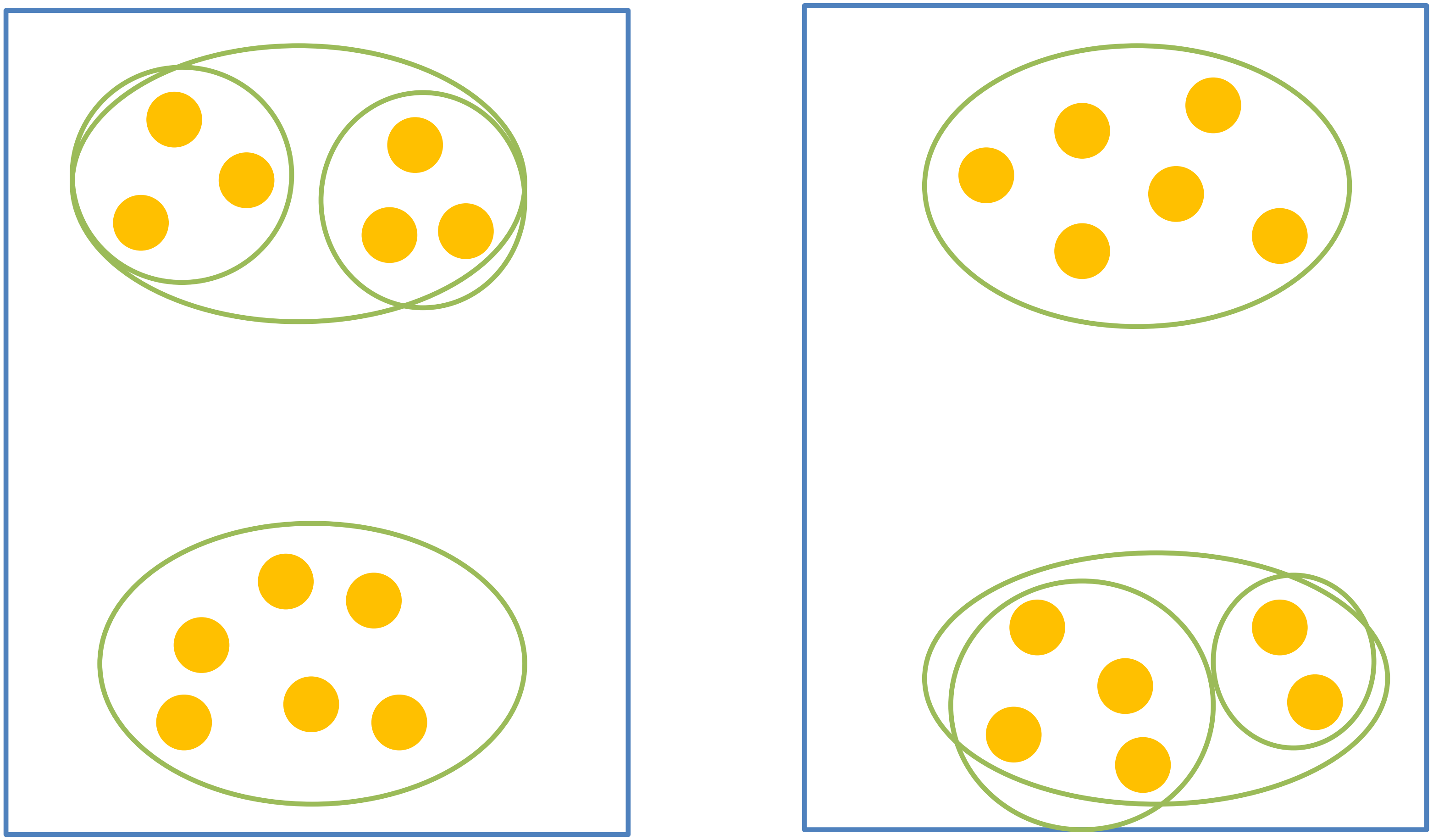
- Percentage of correct classifications
- Compare pairs of elements:

$$R = \frac{tp+tn}{tp+tn+fp+fn}$$



- Fp and fn are equally weighted

Stability

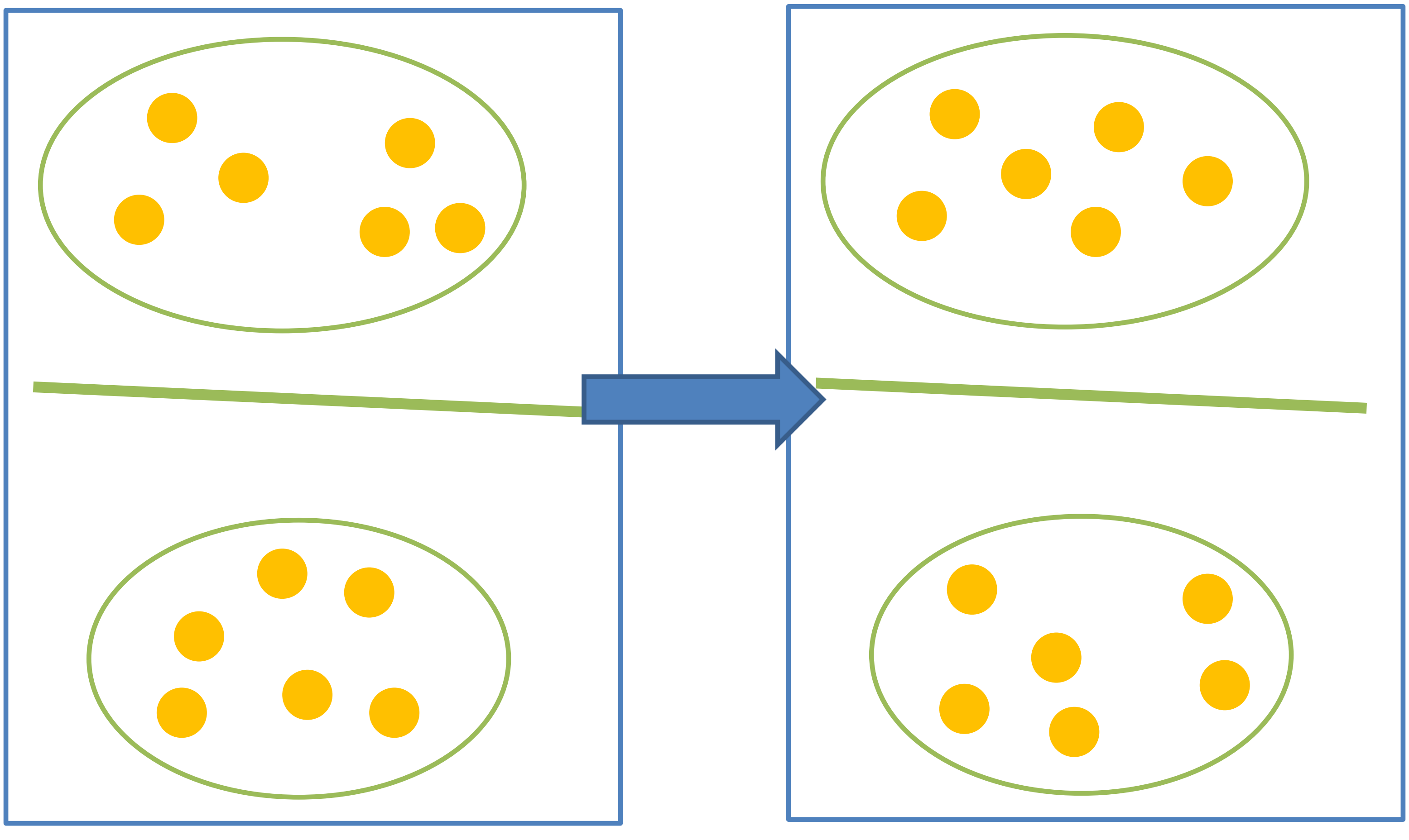


Stability

- What is the right number of clusters?
- What makes a good clustering solution?

- Clustering should generalize!

Stability



Summary

- We have covered a lot today

- Clustering
  - K-means
  - Mean-shift
  - Hierarchical clustering

- Evaluation criteria
  - Rand index
  - Stability