COMP 472: Artificial Intelligence
Machine Learning
Unsupervised Learning video #6

- Russell & Norvig: *not much really*

Today

1. Introduction to ML
2. Naïve Bayes Classification
   a. Application to Spam Filtering
3. Decision Trees
4. (Evaluation
5. Unsupervised Learning)
6. Neural Networks
   a. Perceptrons
   b. Multi Layered Neural Networks

supervised

Types of Machine Learning

lots of data is unlabeled

unlabeled

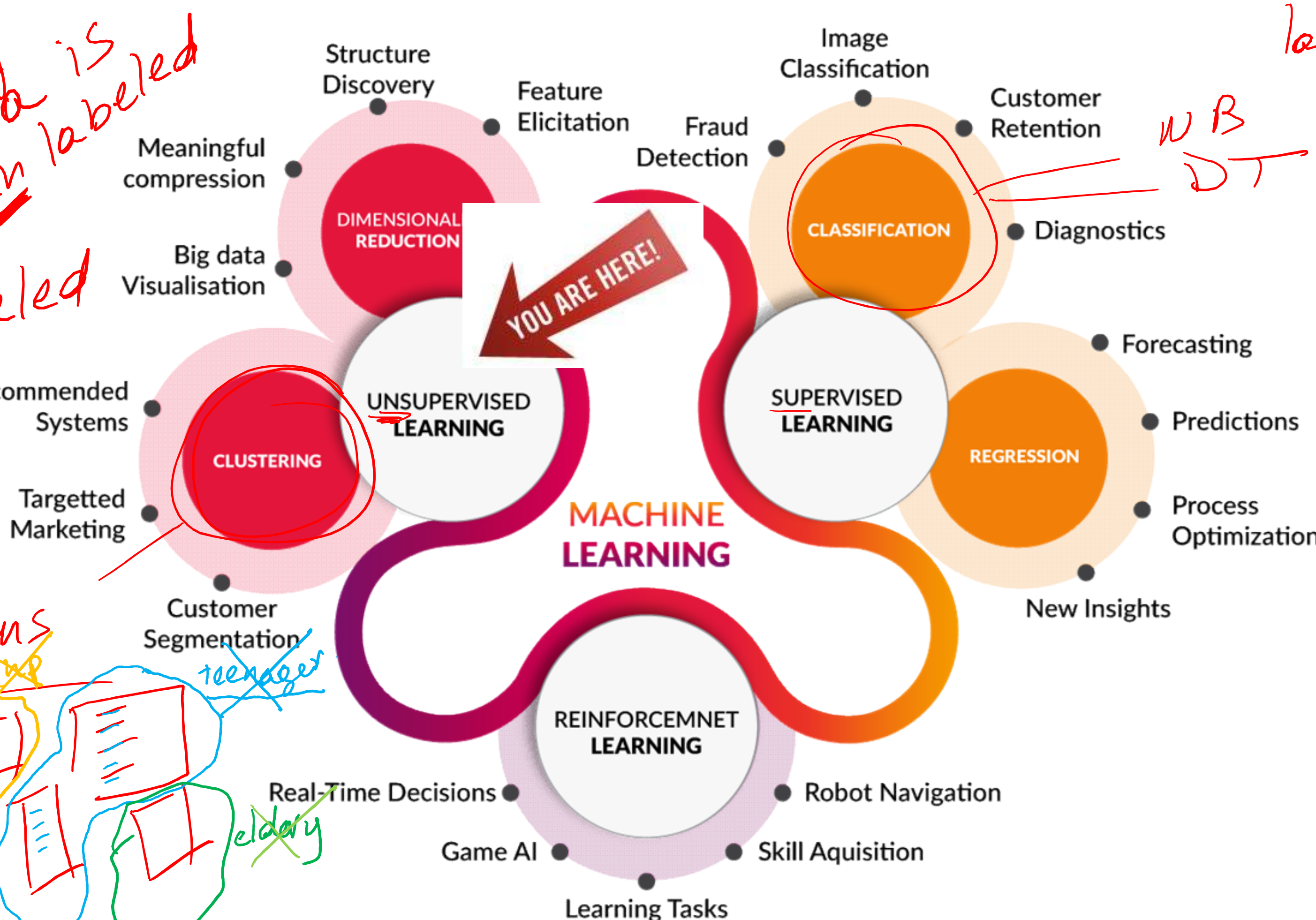
x ? f(x)

k-means

labeled data

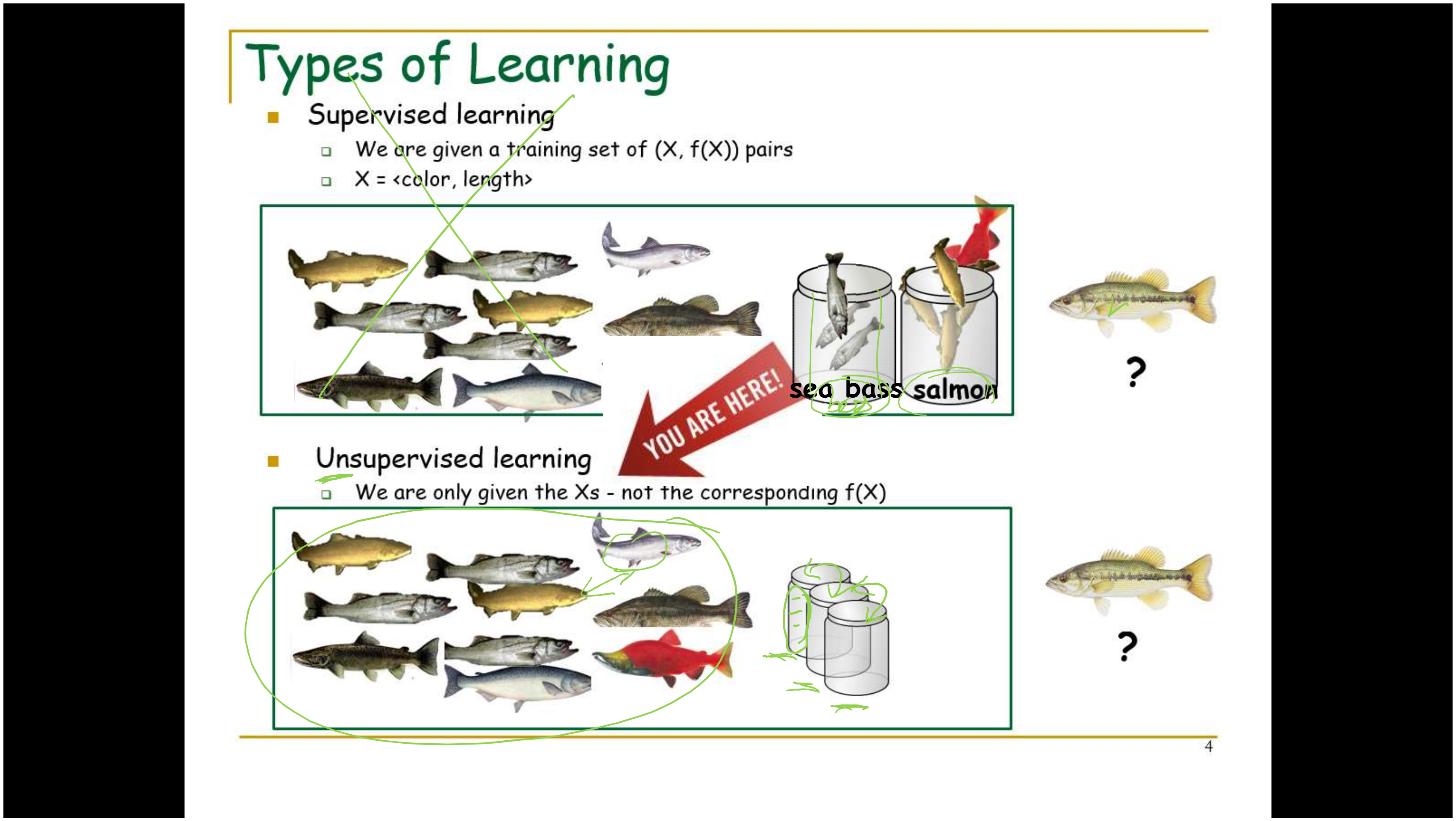
NB DT

x f(x)



http://www.cognub.com/index.php/cognitive-platform/

Remember this slide?



Unsupervised Learning

- Learn without labeled examples
  - i.e. X is given, but not f(X)

small nose	big teeth	small eyes	moustache	f(X) = ? =

X

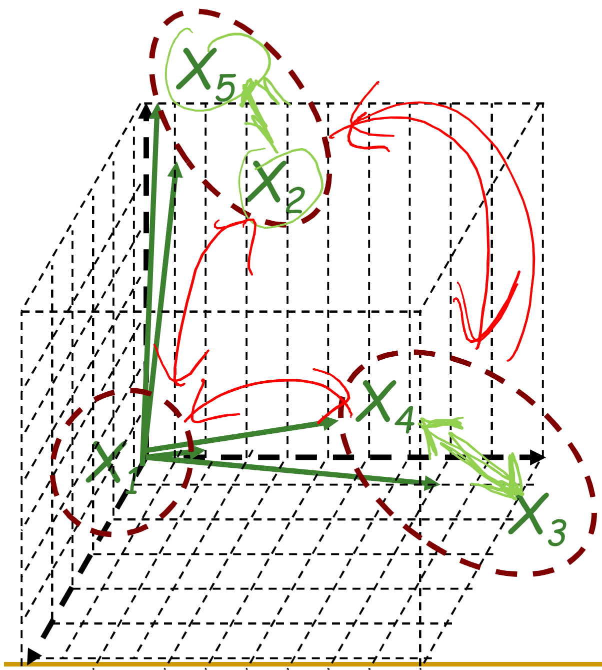
not given

- Without a f(X)
  - you can't really identify/label a test instance
  - but you can:
    - Cluster/group the features of the test data into a number of groups
    - Discriminate between these groups without actually labeling them

Clustering

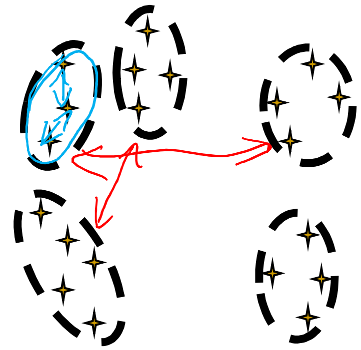
- Represent each instance as a vector $\langle a_1, a_2, a_3, \ldots, a_n \rangle$
- Each vector can be visually represented in a n dimensional space

	a_1	a_2	a_3	Output
X_1	1	0	0	?
X_2	1	6	0	?
X_3	8	0	1	?
X_4	6	1	0	?
X_5	1	7	1	?



k-means Clustering

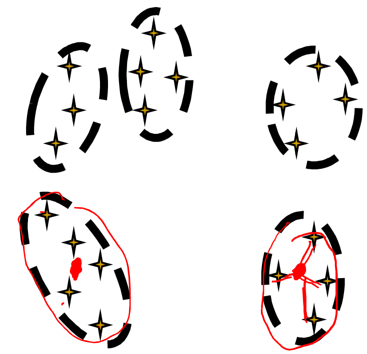
1. Represent each instance as a point on a n dimensional space
2. Partition points into k regions such that:
   - distance between points within a region is minimized
   - distance between points across regions is maximized



- Naturally works well with features with numerical values
  - where distance between points can be measured by the Euclidean distance
- Needs modifications for categorical values
  - which have no order
    - eg. "Honda", "Audi", "BMW", "Ferrari", "Nissan", "Lamborghini"
  - needs domain-specific distance measure

dist(Honda, Nissan) = 1
editdist(Honda, Audi) dist(Honda, Audi) = 3
dist(Ferrari, Lamborghini) = 1
editdist(Honda, nissan)

k-means Clustering



- User selects how many clusters they want (the value of k)

1. Place k points into the space (eg. at random). These points represent initial ~~group~~ cluster centroïds.
2. Assign each data point x_n to the nearest centroïd.
3. When all data points have been assigned, recalculate the positions of the k centroïds as the average of the cluster
4. Repeat Steps 2 and 3 until none of the data instances change ~~group~~ cluster.

Euclidean Distance

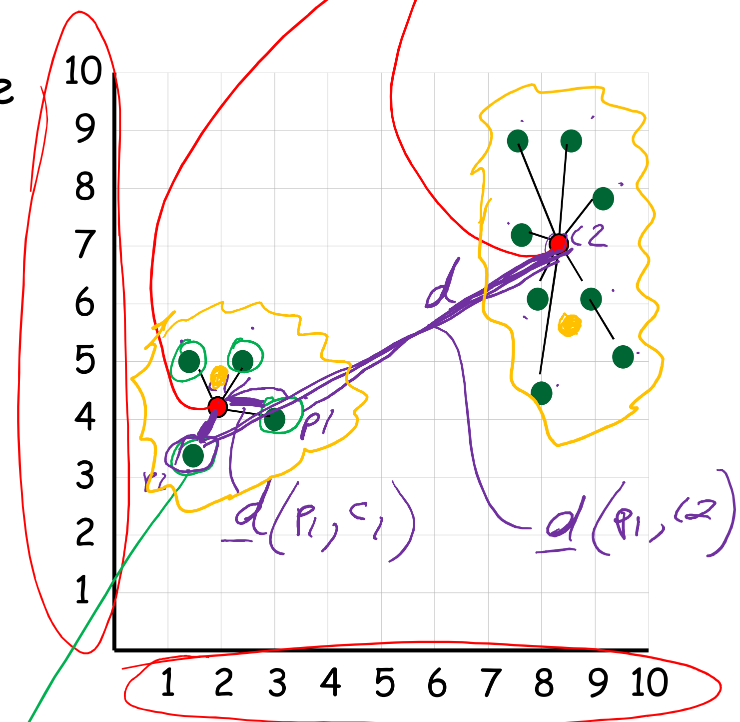
centroid

- To find the nearest centroïd…
  - typical metric is the Euclidean distance
  - Euclidean distance between 2 pts:

$$p = (p_1, p_2, \ldots, p_n) \qquad q = (q_1, q_2, \ldots, q_n)$$

dist(p,q)

$$d = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$



- To compute the next generation of centroïds…
  - take mean of all points in the cluster in each dimension
  - mean of 2 points:

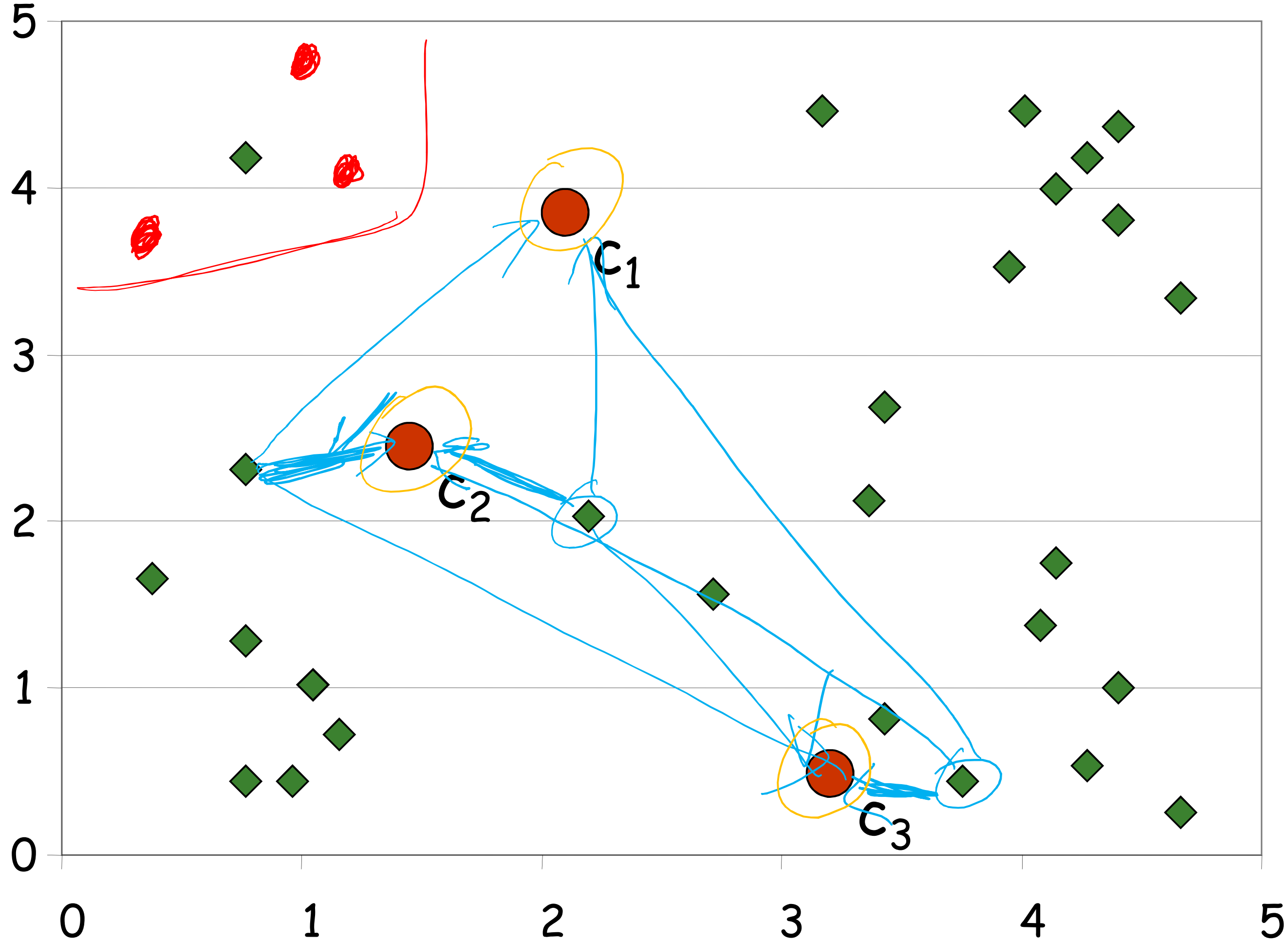
$$p = (p_1, p_2, \ldots, p_n)$$
$$q = (q_1, q_2, \ldots, q_n)$$

$$c = \left(\frac{p_1 + q_1}{2}, \frac{p_2 + q_2}{2}, \ldots, \frac{p_n + q_n}{2}\right)$$

instances in the dataset

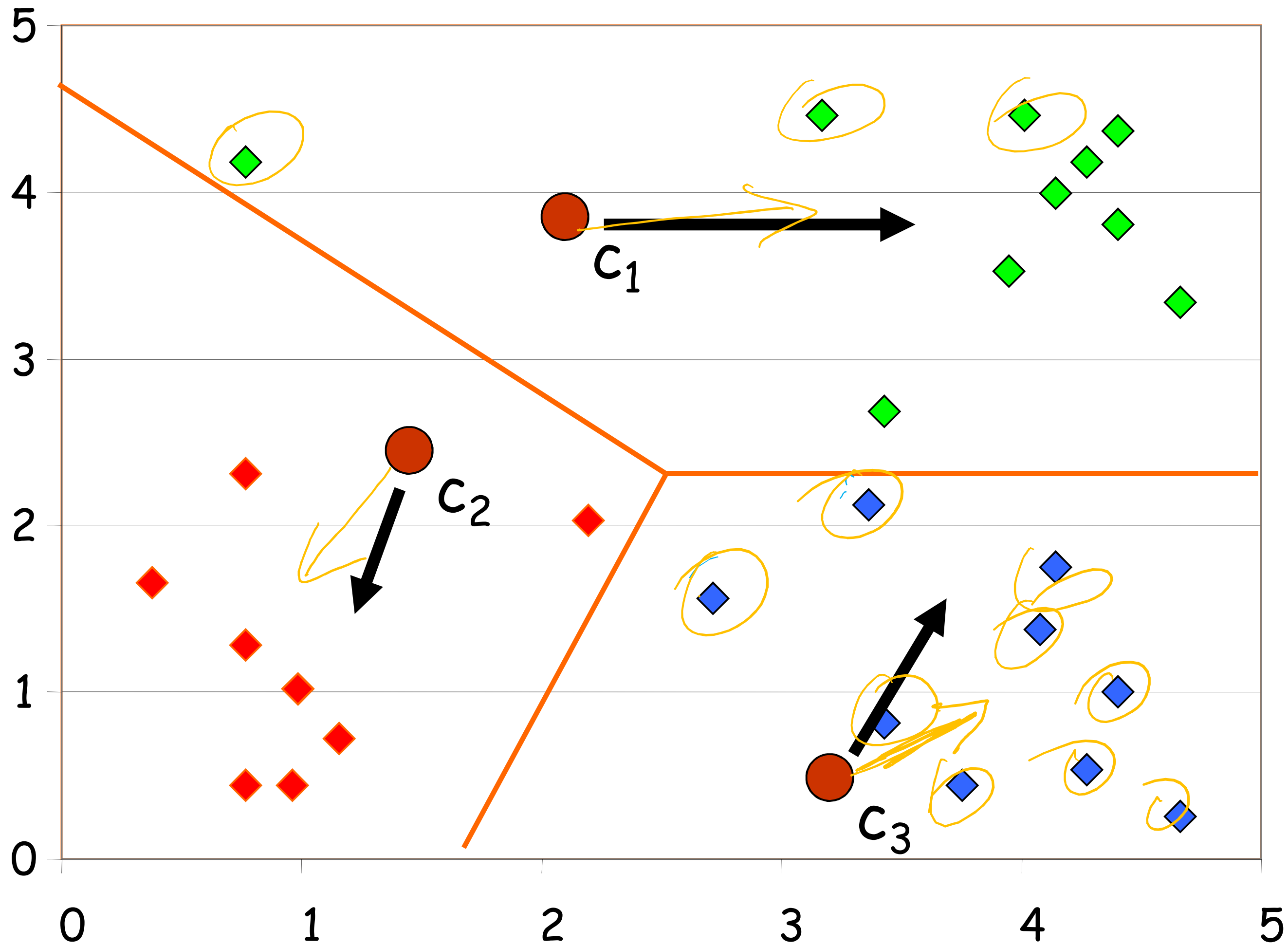
Example (in 2-D... i.e. 2 features)

initial 3 random centroïds



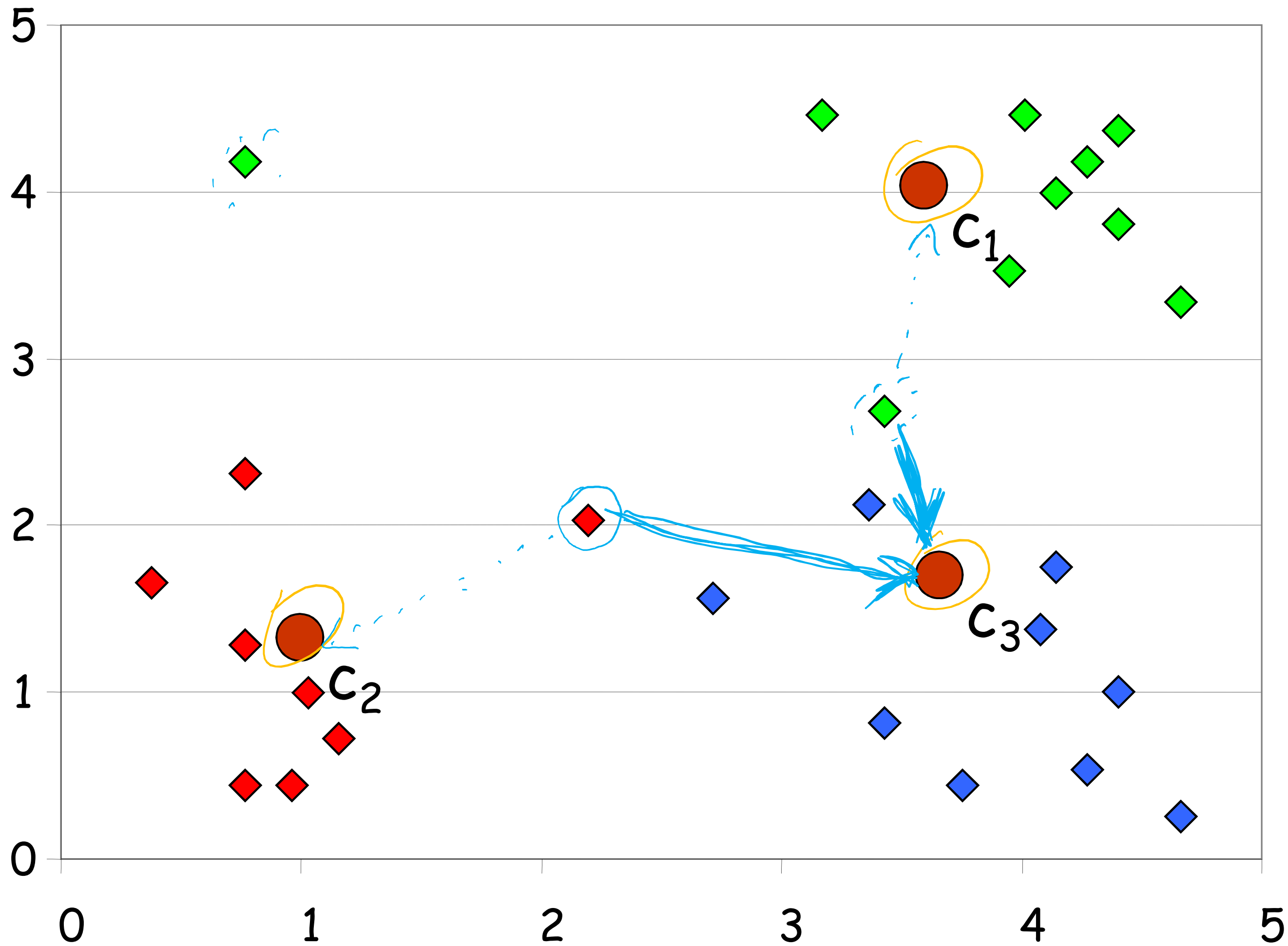
Example

partition data points to closest centroïd



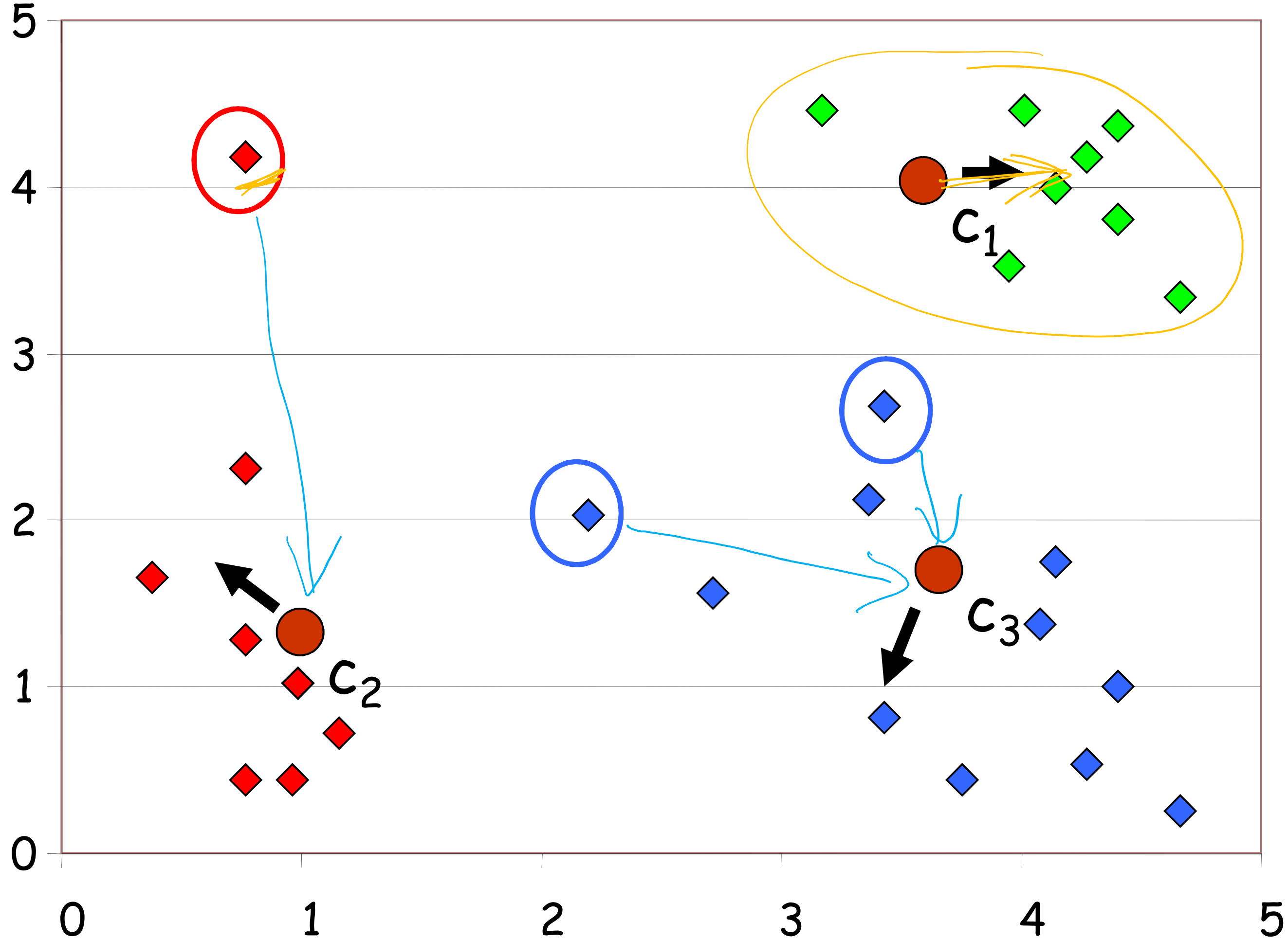
Example

re-compute new centroïds

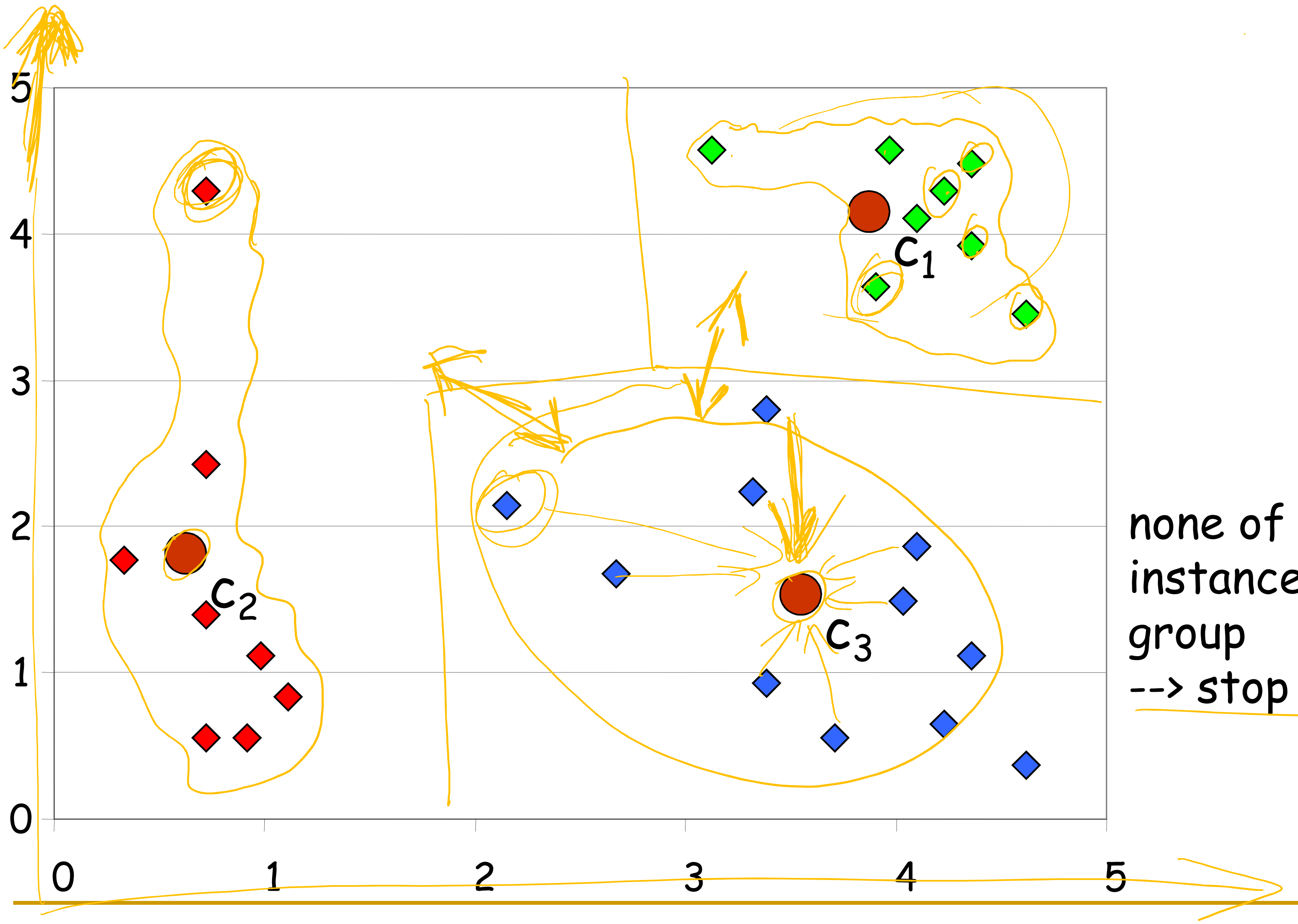


Example

re-assign data points to new closest centroïds



Example



none of the data instances change group
--> stop

Notes on k-means

- negatives:
  - does not guarantee to converge to the global optimum
  - very sensitive to initial choice of centroids
    - many find useless clusters…
  - user must set initial k
    - not easy to do…
- but converges very fast!
- many other clustering algorithms…

Today

1. Introduction to ML
2. Naïve Bayes Classification
   a. Application to Spam Filtering
3. Decision Trees
4. (Evaluation
5. Unsupervised Learning)
6. Neural Networks video #7
   a. Perceptrons
   b. Multi Layered Neural Networks

Up Next

1. Introduction to ML
2. Naïve Bayes Classification
   a. Application to Spam Filtering
3. Decision Trees
4. (Evaluation
5. Unsupervised Learning)
6. Neural Networks
   a. Perceptrons
   b. Multi Layered Neural Networks