MSPR 9: Clustering I

Dr. Hendrik Purwins

AAU CPH

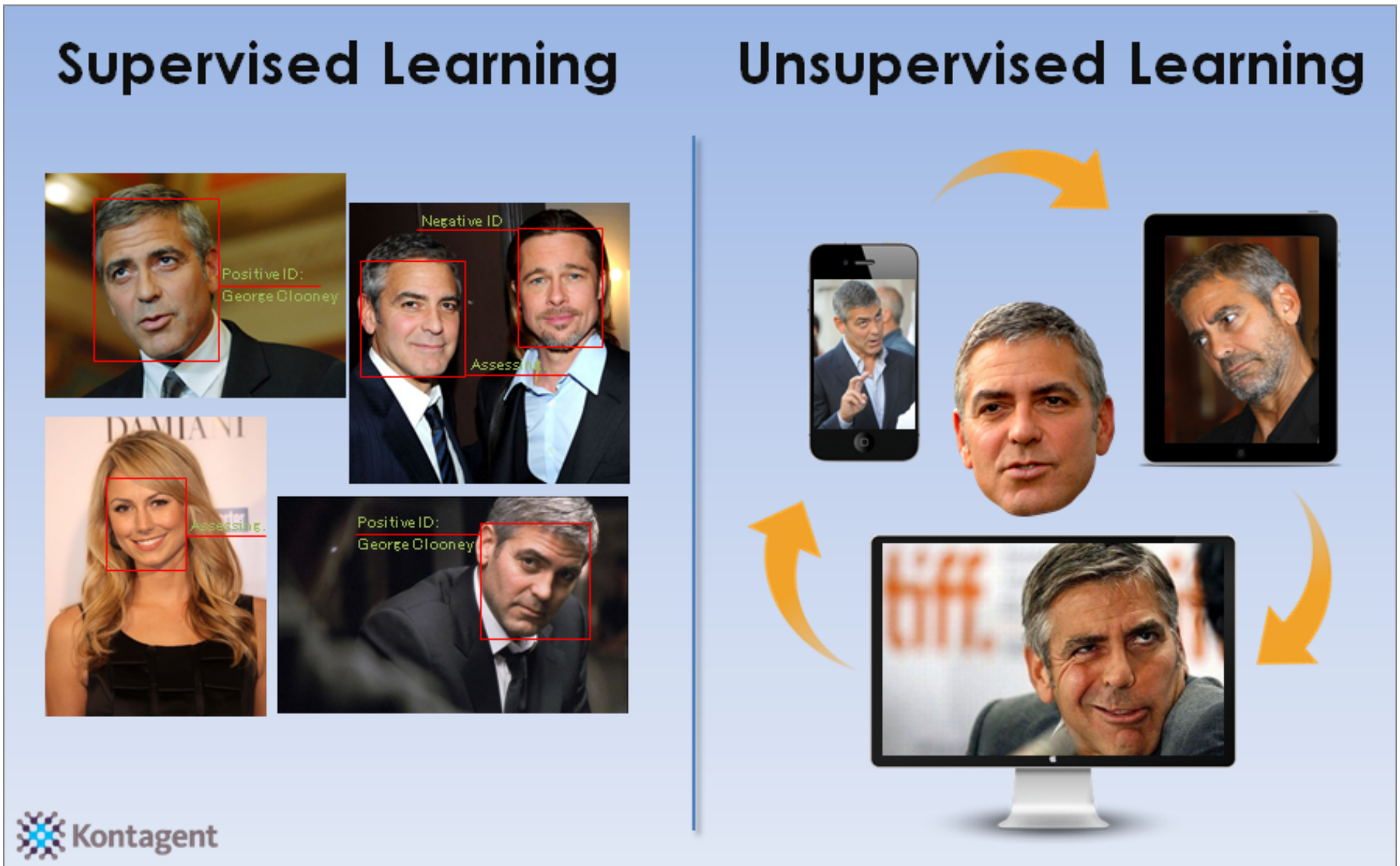
October 26, 2015

Outline

Supervised-Unsupervised Learning

- Supervised learning:
  - Data given with labels
  - Algorithms:
    - Classifiers (Mahalanobis classifier, k- nearest neighbours classifier, linear/quadratic discriminant analysis, support vector machine
    - Regression (linear regression, partial least squares regression, support vector regression)
    - Reinforcement learning (e.g. after a game you get success or not)
- Unsupervised learning:
  - Principal component analysis, non-linear dimension reduction, blind signal separation (cocktail party problem)
  - Clustering (hierarchical clustering, k-means, Gaussian mixture models)

Supervised-Unsupervised Learning: Illustration



Clustering

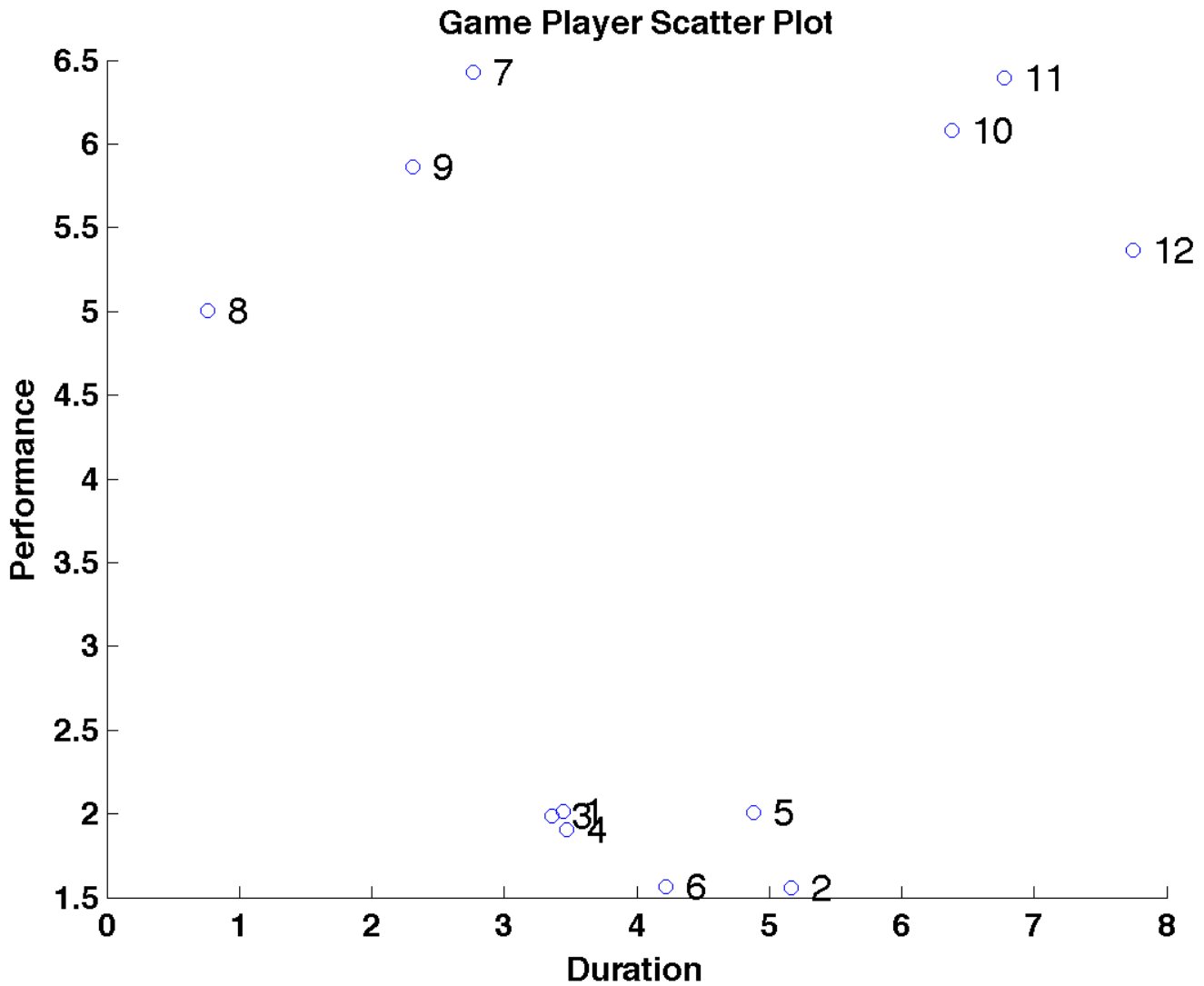
Example Data: Gamer Behaviour (Artificial Data) I

- Load `game_data_small`
- Data is a matrix `X` of 12 rows, a 2-d data points in each column
- Plot data using `scatter`(1st column of X, 2nd column of X)
- `scatter(X(:,1),X(:,2))`
- Use `text` (check help) to put a number next to a point!

Example Data: Gamer Behaviour (Artificial Data) II

```
for i=1:size(X,1),
      scatter(X(i,1), X(i,2));
      hold on;
      text(X(i,1), X(i,2),sprintf(' %d',i));
end
```

Example Data: Gamer Behaviour (Artificial Data) II



Clustering: Application Examples

- Biology: Transcriptomics: Group genes with related expression patterns
- Image segmentation: Divide a digital image into distinct regions for object recognition
- Computer games: Identify player types to adjust level
- Marketing
  - Partition general population for market segmentation, product positioning, new product development
  - Grouping of shopping items.
- WWW
  - Social network analysis: identify communities
  - Improving web search (google) to intelligently group websites together.
- Social science
  - Educationion: identify groups of students with similar properties
  - Typologies: identify typologies of opinions, habits, demographics, useful in politics and marketing

Clustering

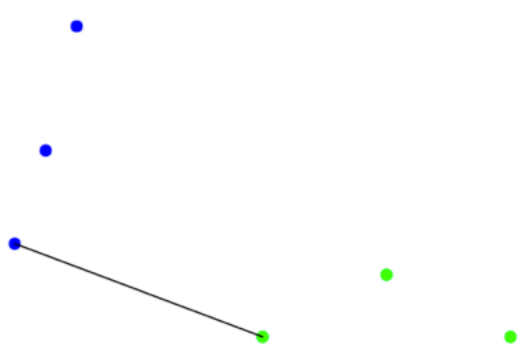
- We want to know if the high dimensional data are structured in a meaningful way
- Maybe some points (high-dim features) group together forming clusters
- They maybe interpreted in a certain way (e.g. as sub-population, gene group, phoneme)
- We can do that without any additional knowledge about the vectors (unsupervised learning)

Agglomerative Clustering with Single Linkage

Bottom-up building of clustering tree.

1. Initialization: Every object is one cluster
2. A new cluster is generated by merging the closest pair of clusters.
   - For a cluster containing more than 1 object the distance between two clusters A (blue) and B (green) is calculated as

$$\min_{a \in A, b \in B} d(a, b)$$

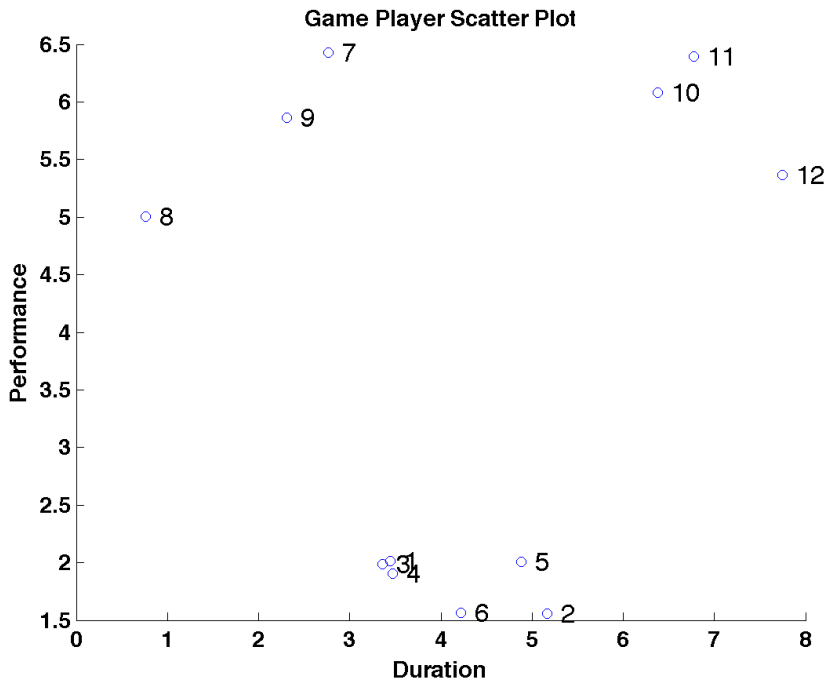


3. Repeat 2. until cluster is reached that contains all objects.

Agglomerative Clustering in Matlab I

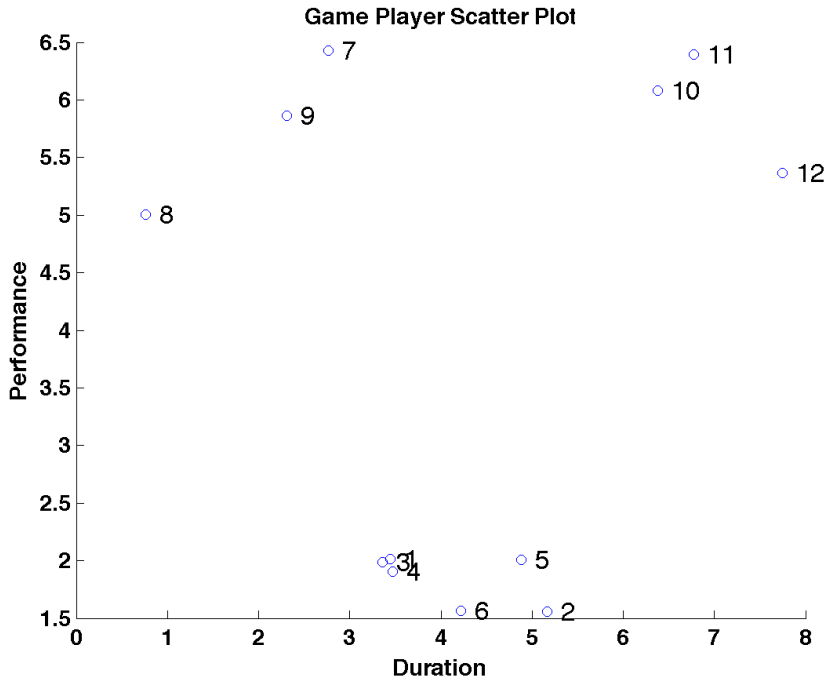
- Use `linkage` to cluster `X`
- `l = linkage(X)` `l =`

```
1.0000  3.0000  0.0858
4.0000 13.0000  0.1097
10.0000 11.0000  0.5052
2.0000  5.0000  0.5333
7.0000  9.0000  0.7201
6.0000 16.0000  0.7952
14.0000 18.0000  0.8273
12.0000 15.0000  1.4144
8.0000 17.0000  1.7759
20.0000 21.0000  3.6295
19.0000 22.0000  3.9848
```

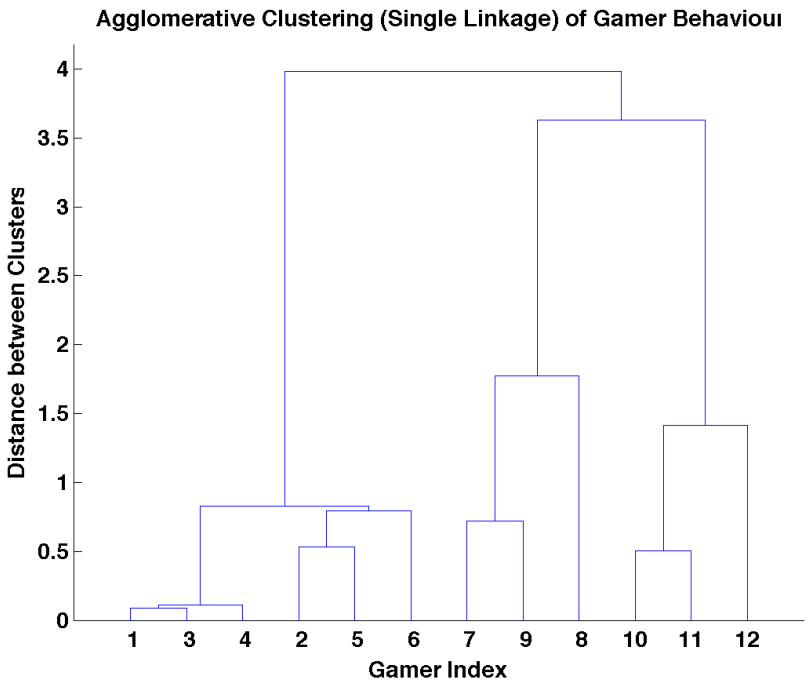


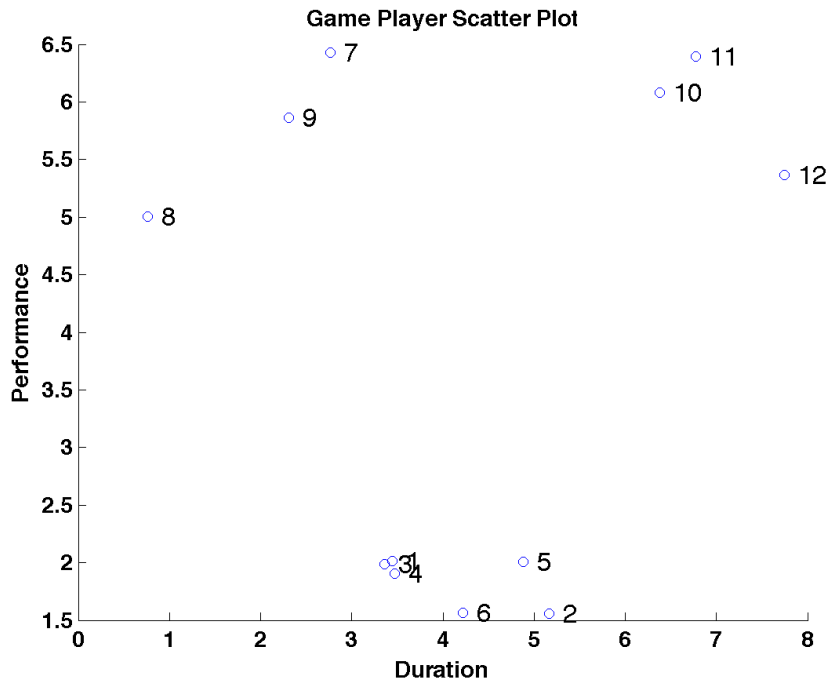
Agglomerative Clustering in Matlab II

- `l = linkage(X)`
- How to visualize *l*?
- Do a dendrogram in Matlab!
- `[H T order]=dendrogram(l)` (order gives you the indices of the clustered vectors in the order of the dendrogram leaves in the display)



Agglomerative Clustering in Matlab III

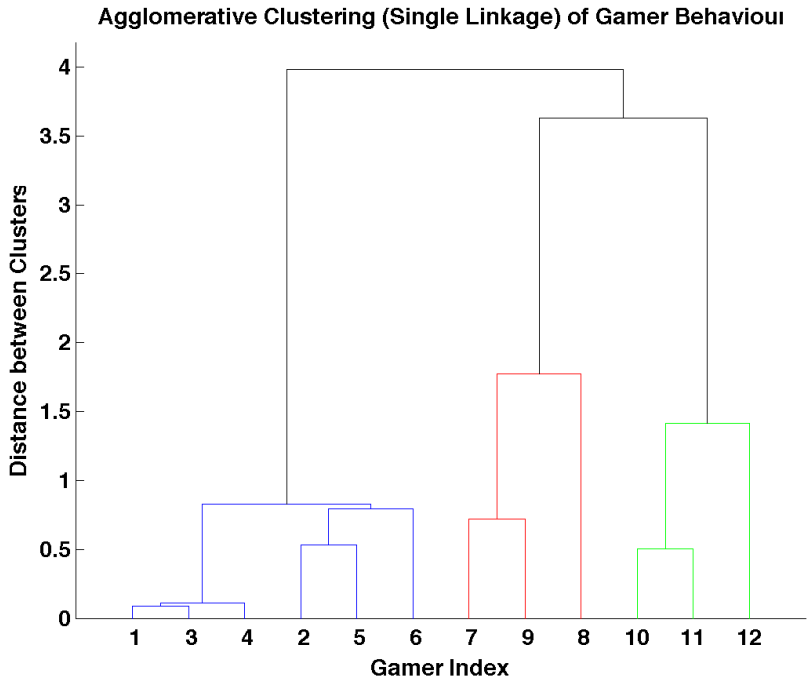




Agglomerative Clustering in Matlab IV

- Where to cut the clustering tree? 2.5
- Try `dendrogram` with cutoff threshold 2.5

```
dendrogram(l,...
'colorthreshold',2.5)
```

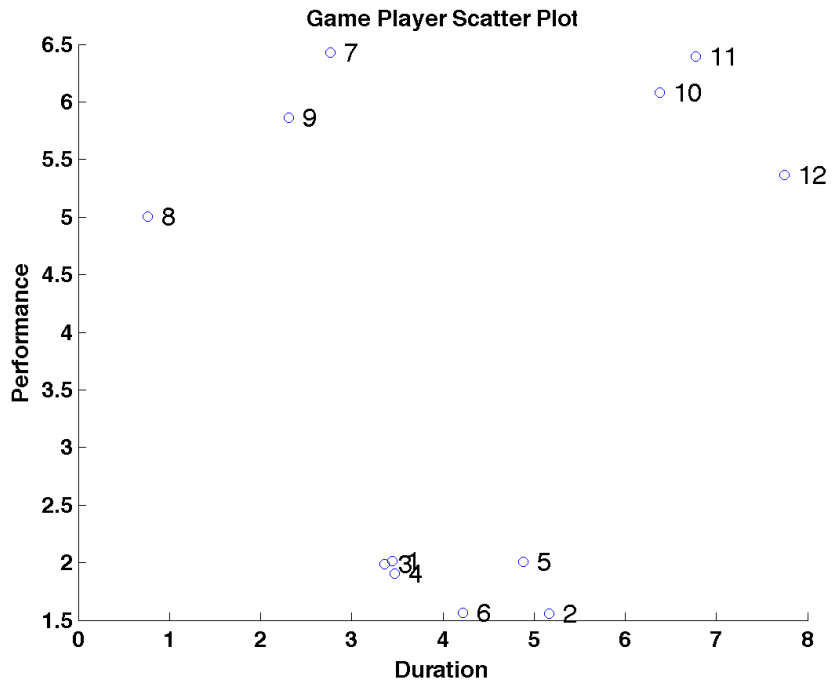


Agglomerative Clustering in Matlab V

```
dendrogram(l,'colorthreshold',2.5)
```

- Colour the points in the scatter plot according to which cluster they belong to.
- Use `cluster` with the right cut-off threshold and the criterion 'distance'
-
```
r=
cluster(l,'cutoff',2.5,...
'criterion','distance')
```
-
```
>> r'
ans =
2 2 2 2 2 2 3 3 3 1 1 1
```

Use `scatter (X(:,1),X(:,2),[],r);`



Class Assignment

Perform agglomerative clustering on the iris data. From the 150×4 *feature matrix only take 10 items of each iris species, that is rows 1-10 (setosa), 51-60 (versicolor), 101-110 (virginica). Decide at which clustering distance to cut the clustering tree.*

K-means Clustering

- Organization of data as groups (clusters) of individual feature vectors
- Expectation maximization (EM) algorithm

Expectation Maximization (EM) Algorithm I

1. The feature vectors $\mathbf{x}_n \in \mathbb{R}^J$ are randomly separated into k clusters $C_i (1 \leq i \leq k)$
2. The mean $\bar{\mathbf{x}}_i$ of each cluster C_i is calculated from the I_i vectors of that cluster (*E-step*):

$$\bar{\mathbf{x}}_i = \frac{1}{I_i} \sum_{\mathbf{x} \in C_i} \mathbf{x}.$$

3. Each point $\mathbf{x}$ is re-assigned to the mean $\bar{\mathbf{x}}_m$ to which it has least Euclidean distance (*M-step*):

$$m = \arg \min_{1 \leq i \leq k} (\|\bar{\mathbf{x}}_i - \mathbf{x}\|)$$

4. Repeat 2. and 3. until convergence

Expectation Maximization (EM) Algorithm II

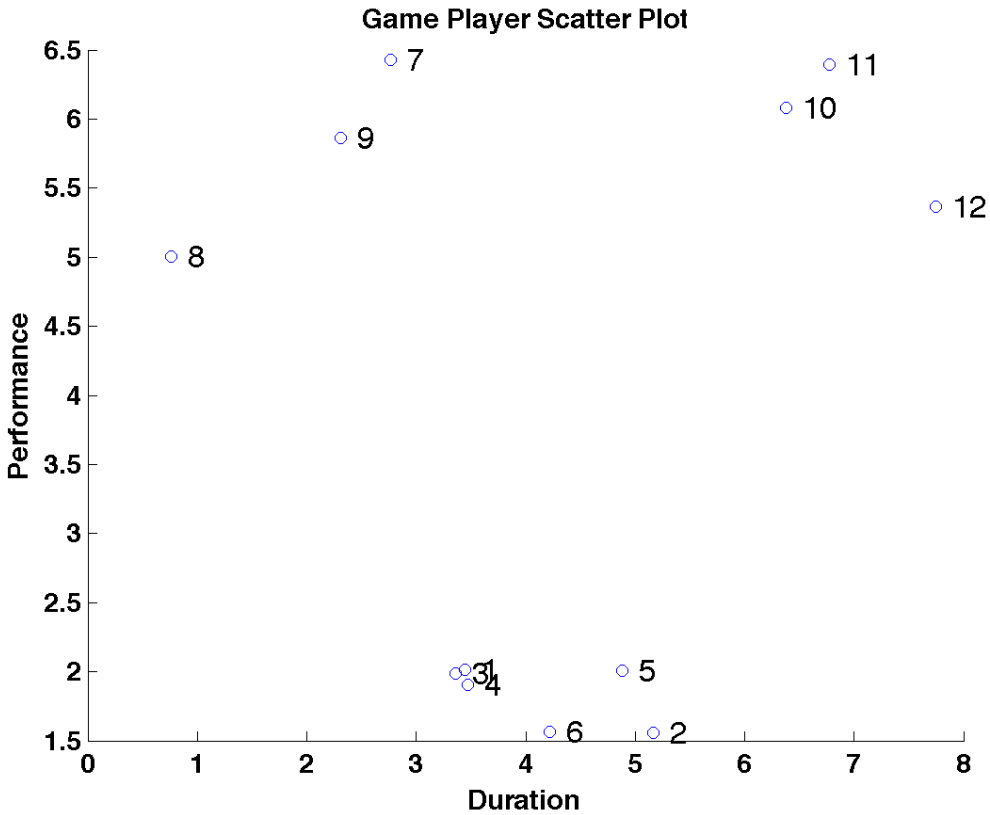
E-step :

$$\bar{\mathbf{x}}_k = \frac{1}{I_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}.$$

M-step :

$$m = \arg \min_{1 \leq k \leq K} (\|\bar{\mathbf{x}}_k - \mathbf{x}_n\|)$$

1. Repeat E- and M-step until convergence



K-means Clustering

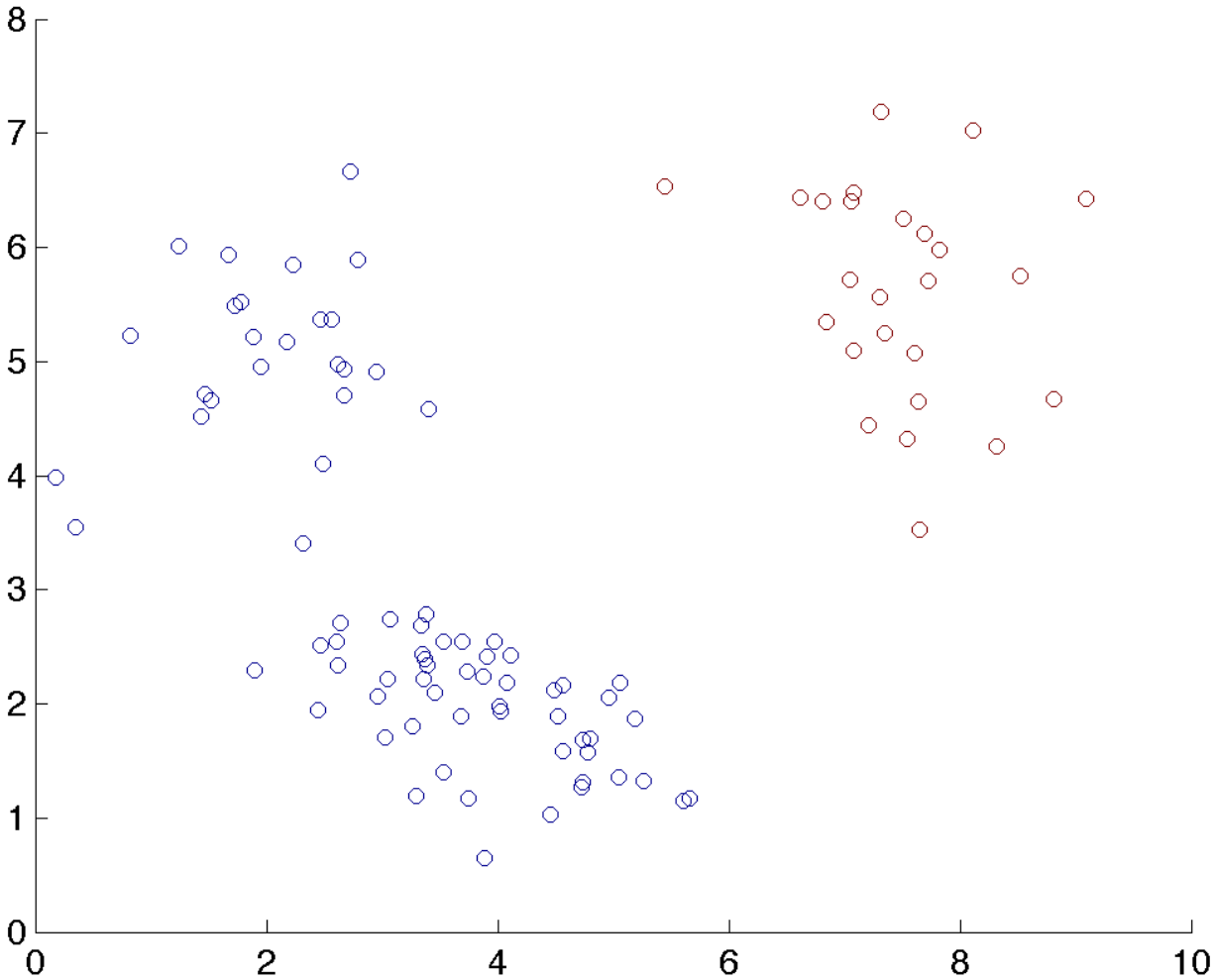
- Convergence guaranteed
- In different runs algorithm can run into different cluster configuration
- However if we do not know anything about the data it is difficult to interpret what the clusters mean.

K-means in Matlab

- Load larger dataset `game_data`
- Use `kmeans` for $k = 2, 3, 4$
- `prmfcc=prdataset(mfcc); idx=kmeans(prmfcc,2)` (there is two functions called kmeans.m, make sure you have the data as a prdataset.)
- Do the scatter plot with coloured cluster memberships
- `scatter(X(:,1),X(:,2),[],idx);`

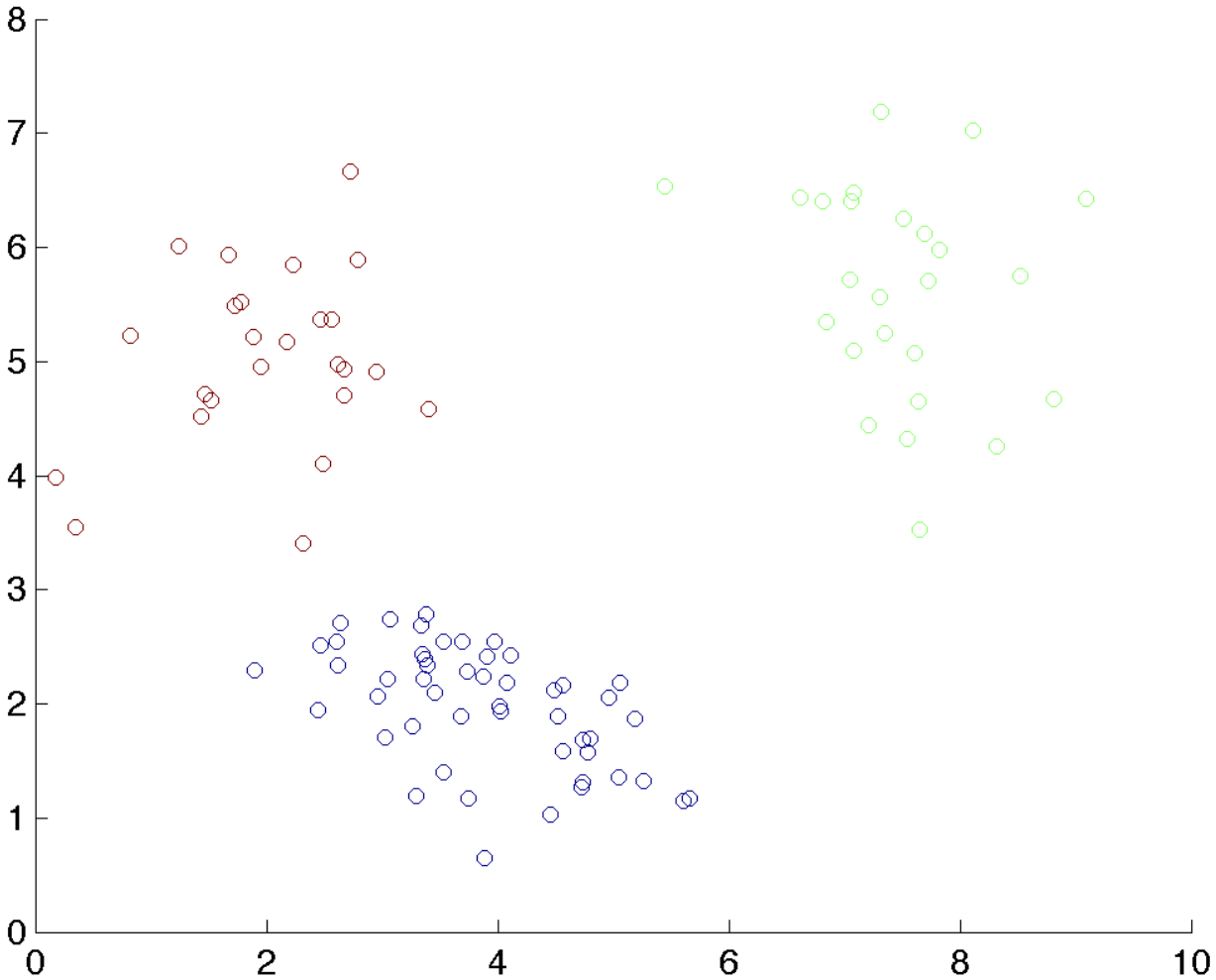
K-means in Matlab

How is this clustering?



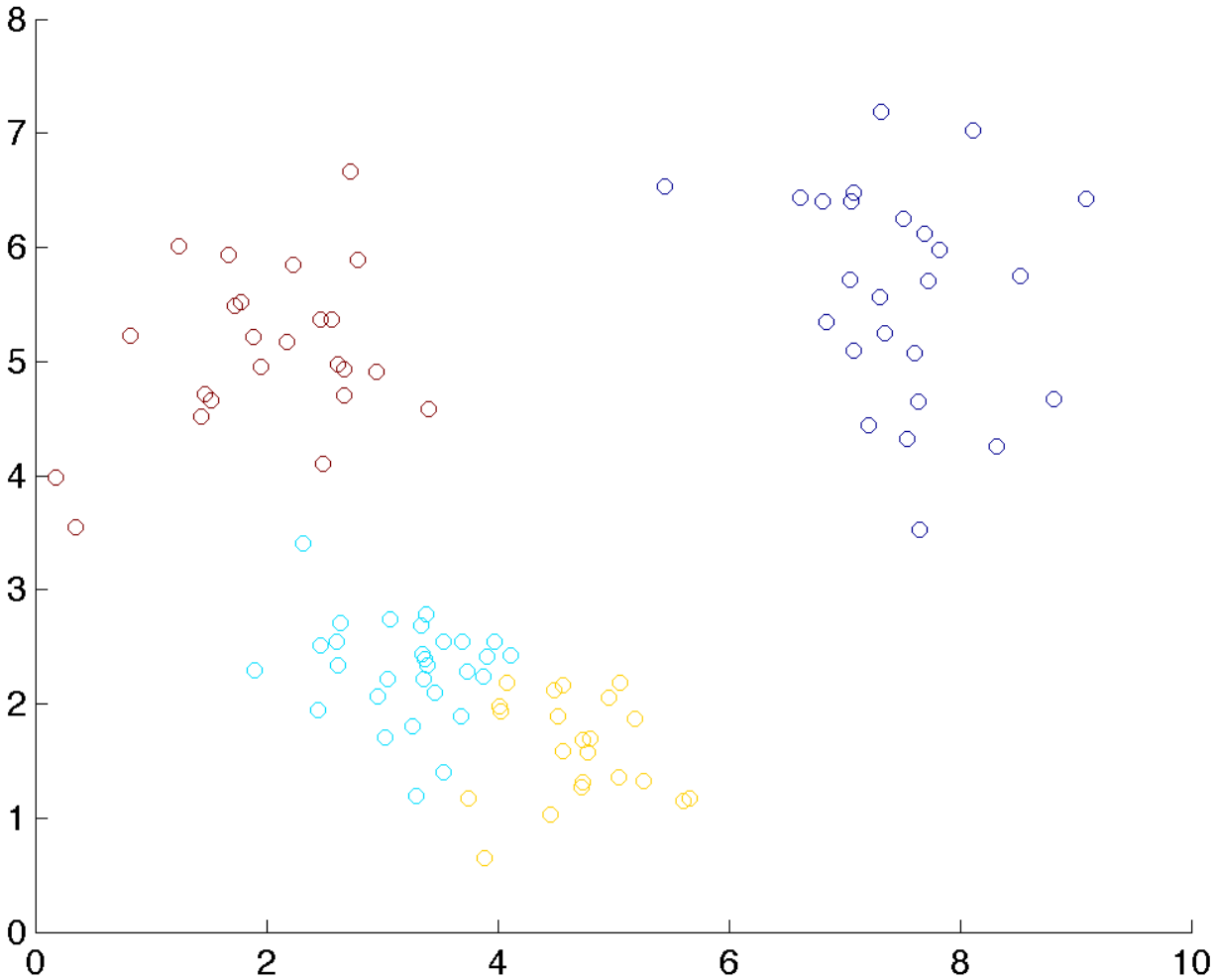
K-means in Matlab

How is this clustering?



K-means in Matlab

How is this clustering?



Can we Determine the Number of Clusters Automatically?

- I: number of data, k: number of clusters, I_i number of data in cluster C_i with cluster mean $\bar{\mathbf{x}}_i$ and overall mean $\bar{\bar{\mathbf{x}}}$
- Overall within-cluster variation SS_W:

$$SS_W = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} I_i \|\mathbf{x} - \bar{\mathbf{x}}_i\|^2$$

- Overall between-cluster variation SS_B:

$$SS_B = \sum_{i=1}^{k} I_i \|\bar{\mathbf{x}}_i - \bar{\bar{\mathbf{x}}}\|^2$$

- *Variance ratio criterion*

$$VRC_k = \frac{(I-k)SS_B}{(k-1)SS_W}$$

Variance-Ratio Criterion in Matlab

- X data matrix (I rows=data points, J features)
- Check cluster numbers $1, \ldots, K$
- Variance-Ratio Criterion also called Calinski-Harabasz Criterion

Applied to game data:

```
load game_data eva =
evalclusters(X,'kmeans','CalinskiHarabasz','KList',[1:K])
```

Output:

```
eva=
CalinskiHarabaszEvaluation with properties:
NumObservations:  100
InspectedK: [1 2 3 4 5 6]
CriterionValues:  [NaN 138.4757 284.5157 268.4199 249.8122
225.1251]
OptimalK: 3
```

3 player types!

Class Assignment

According to the Variance Ratio criterion how many clusters are there in the iris data? Take the full feature matrix (150×4) *test for k-means with cluster* k *from* $1 \ldots 10$