UNIVERSITÀ DEGLI STUDI DI MILANO

Unsupervised learning

Enrique Muñoz Ballester

Dipartimento di Informatica
via Bramante 65, 26013 Crema (CR), Italy
enrique.munoz@unimi.it

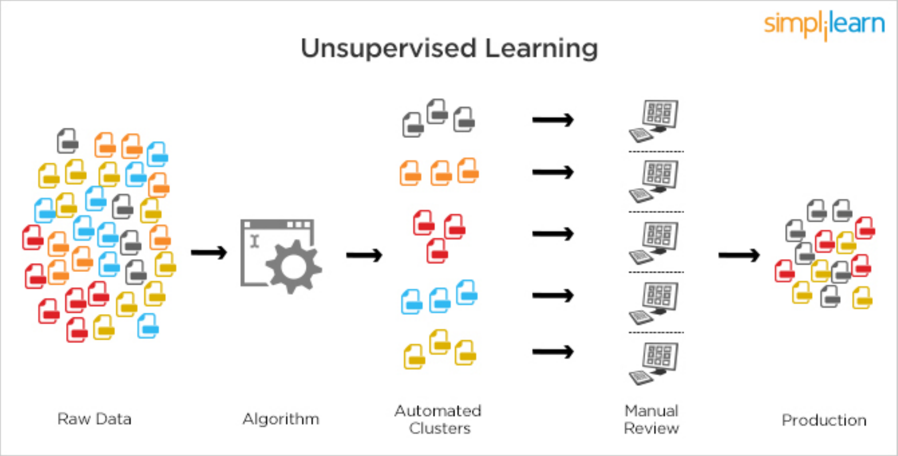
Material

- Download slides data and scripts:

https://homes.di.unimi.it/munoz/teaching.html

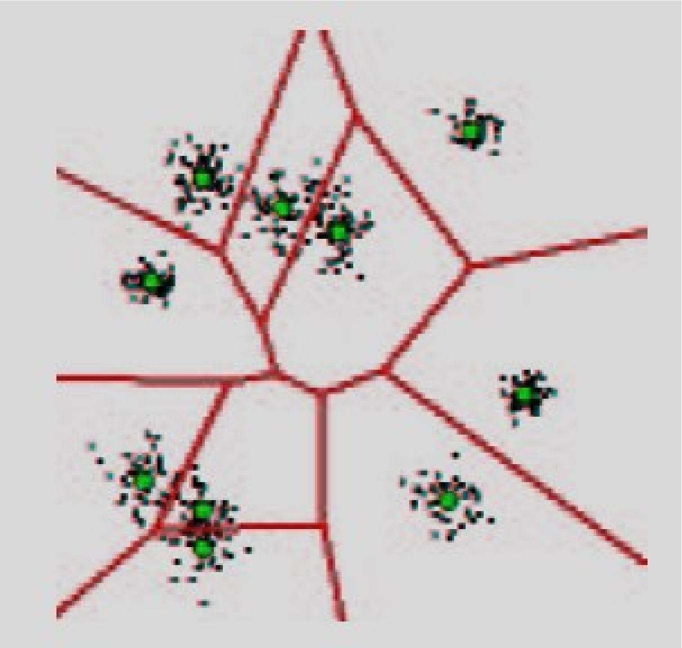
Unsupervised learning

- Supervised learning:
  - Methods such as regression and classification
  - We observe both a set of features $P_1, P_2, \ldots, P_n$ for each object, as well as a response or outcome variable Y
  - The goal is then to predict Y using $P_1, P_2, \ldots, P_n$
- Unsupervised learning
  - We observe only the features $P_1, P_2, \ldots, P_n$
  - We are not interested in prediction, because we do not have an associated response variable Y



The Goals of Unsupervised Learning

- The goal is to discover interesting things about the measurements:
  - Is there an informative way to visualize the data?
  - Can we discover subgroups among the variables or among the observations?
- We will discuss three methods:
  - Self organizing maps
  - K-means
  - Fuzzy C-means



Applications of unsupervised learning

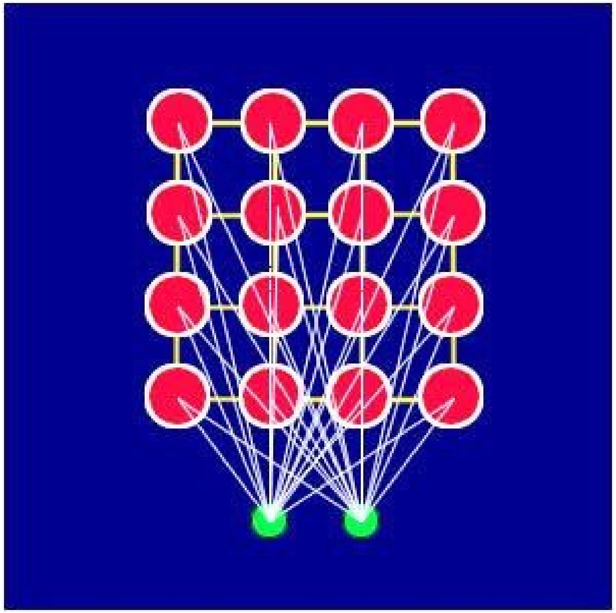
- Market segmentation by grouping people according to their buying patterns
- Bioinformatic analysis by grouping genes with related expression patterns
- Profiling of the behavior of criminals
- Categorization of galaxies
- Categorization of real estates
- Exploration of full-text databases, i.e., document organization and retrieval
- Image segmentation
- …

Self organizing maps

- Also known as SOMs or Kohonen maps
- A type of neural network
- Capable of representing multidimensional data in much lower dimensional spaces, usually one or two dimensions
- Store information so that any topological relationships within the training set are maintained

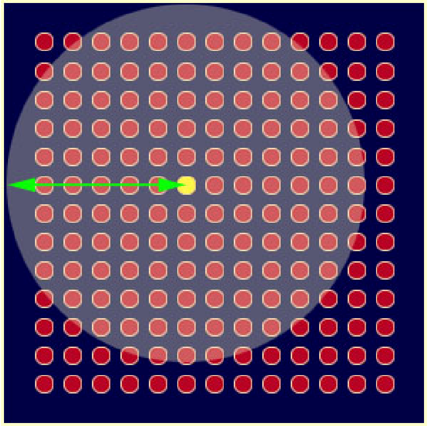
SOM: network architecture

- Lattice of nodes fully connected to the input layer
- Neurons can be arranged in n-dimensional patterns, we will focus on 2-dimensional patterns
- Each node has a specifical topological position (coordinates x and y)
- Each node contains a vector of weights of the same dimension of the input vectors: $w_1, w_2, \ldots, w_n$
- Lines indicate adjacency



SOM: basic learning algorithm

1. Initialize the weights
   - Typically small standardized random values
2. Present a sample to the network
3. Calculate the Best Matching Unit (BMU)
   - $Dist = \sqrt{\sum_{i=0}^{i=n}(p_i - W_i)^2}$
4. Determine the BMU's Local Neighborhood
   - $N_i(d)=\{j, d_{ij} \leq d\}$
5. Adjust the weights of BMU and its neighbors
   - $w_i(q)=w_i(q-1)+\alpha(p(q)-w_i(q-1))$

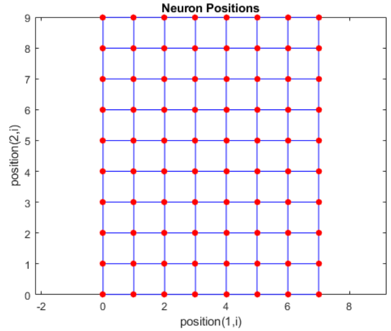


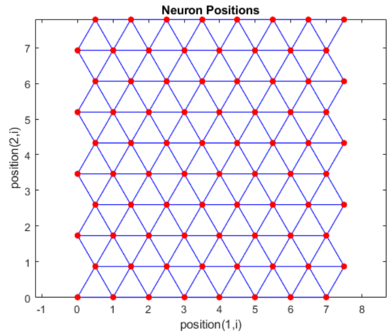
SOM: learning algorithm phases

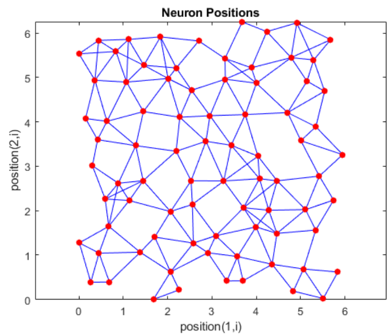
- **Ordering Phase**
  - The neighborhood distance starts at a given initial distance, and decreases to 1
  - The neurons of the network typically order themselves in the input space with the same topology in which they are ordered physically.
- **Tuning Phase**
  - Only the winning neuron learns for each sample
  - Refining cluster centers

SOM: topologies

- Grid
  - Matlab function *gridtop*
- Hexagonal (default)
  - Matlab function *hextop*
- Random
  - Matlab function *randtop*

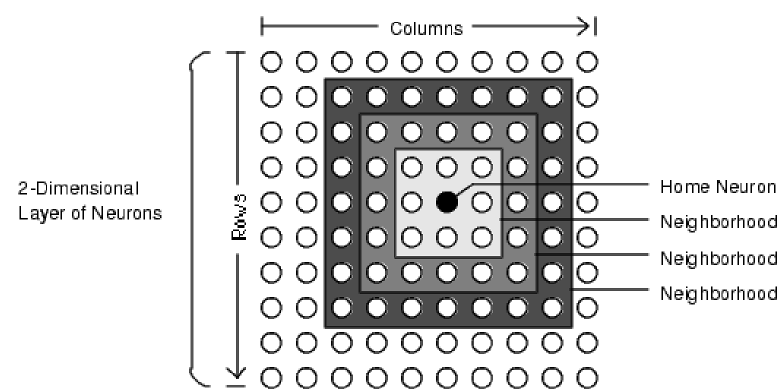






SOM: distances

- Euclidean
  - Matlab function *dist*
- Link (default)
  - Number of links to get to the considered neuron
  - Matlab function *linkdist*
- Manhattan
  - D = sum(abs(x-y))
  - Matlab function *mandist*
- Box
  - Matlab function *boxdist*



SOM: Matlab commands

- Create SOM
  - som=selforgmap(dimensions, ordering_epochs, initial_neighbor_distance, topologyFcn, distanceFcn)
  - Example: som=selforgmap([10 10], 100, 5, 'hextop', 'linkdist')
- Indicate total number of epochs (ordering + tuning):
  - som.trainParam.epochs=200
- Training
  - [som, stats] = train(som, data)
- Plotting network activations
  - plotsomhits(som, data)

Example 1

- Training on simple 3-dimensional data
  - Use a SOM to map a 3-D space (input) into a 2-D space (SOM grid)
  - Data:
    - Included in sphere_data.mat (https://homes.di.unimi.it/munoz/teaching.html)
    - Points given by their Cartesian coordinates (*x; y; z*), that lie on the unit sphere ($x^2 + y^2 + z^2 = 1$)
    - Form two clusters:
      - Cluster 1: samples from 1:100
      - Cluster 2: samples from 101:200

Exercises

1. Study overlap of activations for two clusters
   – Train two models, one using P20 and the other using P30 (included in sphere_data, download from https://homes.di.unimi.it/munoz/teaching.html)
   – Use plotsomhits to study overlap (you can use the code in example1)
   – What amount of overlapping do you see for each different data set?
2. Study activations with models trained using some other dataset (e.g. train using P10, test using P30)
   – What is the response of a network that is trained on a data set with a large standard deviation, when used on a data set with a small standard deviation? Why?
   – What is the response of a network that is trained on a data set with a small standard deviation, when used on a data set with a large standard deviation? Why?

Example 2

- Mapping of RGB color data
  - Create a mapping from 3 to 2 dimensions
  - Analyze how SOMs preserve the “topology” of the data
  - Data:
    - Included in rgb_data.mat (https://homes.di.unimi.it/munoz/teaching.html)
    - Colors are represented using red, green and blue components, ranging from 0 (no color) to 1 (full color)

Exercises

3. Find a SOM that obtains a smoot representation, i.e., with colors that change only gradually between neighboring nodes
   - Try to optimize the results changing the parameters: number of epochs in the ordering phase and tuning phase, and initial neighborhood size
   - What relation between ordering phase and tuning phase seems to be best in order to get a smooth color map? Why do you think that is?

Data preprocessing: Normalization

- To calculate the BMU we use
  $Dist = \sqrt{\sum_{i=0}^{i=n}(V_i - W_i)^2}$
- Consider a dataset of employees
- Distances between employees is dominated by salary

Empid	Salary	Age	Experience
1	25000	24	4
2	40000	27	5
3	55000	32	7
4	27000	25	5
5	53000	30	5

	1	2	3	4	5
1	0.0000000	15000.0003333	30000.0012167	2000.0005000	28000.0006607
2	15000.0003333	0.0000000	15000.0009667	13000.0001538	13000.0003462
3	30000.0012167	15000.0009667	0.0000000	28000.0009464	2000.0020000
4	2000.0005000	13000.0001538	28000.0009464	0.0000000	26000.0004808
5	28000.0006607	13000.0003462	2000.0020000	26000.0004808	0.0000000

Data preprocessing: Normalization

- How to solve this problem?
- Apply normalization:
  - Scale data to fit in a specific range
  - We will focus on Min Max Normalization, which transforms a value A to B which fits in the range[C,D]

$$B = \frac{A - \min(A)}{\max(A) - \min(A)} \times (D - C) + C$$

Empid	Salary	Age	Experi ence
1	-1	-1	-1
2	0	-0,25	-0,33333
3	1	1	1
4	-0,86667	-0,75	-0,33333
5	0,86667	0,5	-0,33333

	1	2	3	4	5
1	0	1,416667	3,464102	0,724377	2,485737
2	1,416667	0	2,083333	1,000555	1,146129
3	3,464102	2,083333	0	2,885259	1,430229
4	0,724377	1,000555	2,885259	0	2,137041
5	2,485737	1,146129	1,430229	2,137041	0

Example 3

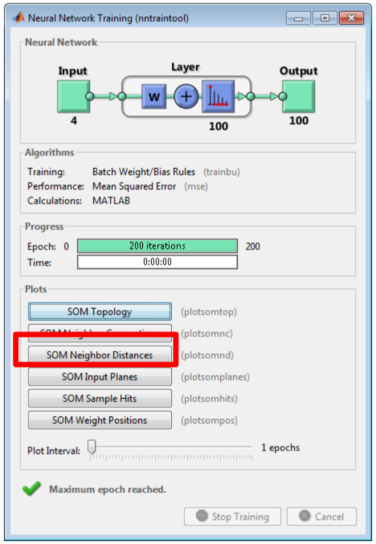
- Analysis of wine data
  - Analyze how SOMs can differentiate classes with unsupervised learning
  - Study overlapping of neuron activations
  - Create a 5x5 node SOM with hextop topology and linkdist distance

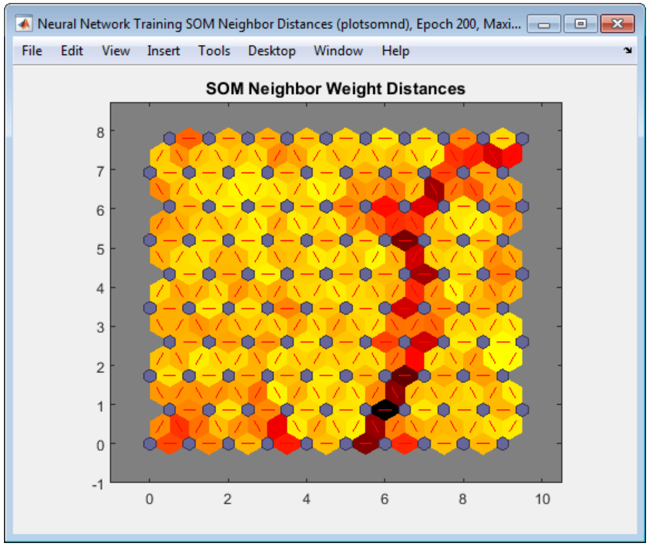
Exercises

4. Try to optimize the parameters of the SOM to reduce overlap
   – *What training parameters did you use? How well does the trained* SOM *separate the classes in your opinion? Is some class easier to separate than the rest?*
5. Repeat the same procedure as with a 10x10-node SOM
   – *What training parameters did you use? What major differences do you see in the results compared to the 5x5-node SOM?*
6. Training with normalized data
   – *Normalize the data using Matlab function mapminmax*
   – *Retrain the 5x5 and the 10x10-node SOMs*
   – *How well do the two SOMs separate the classes in the normalized data?*

Analyzing SOM neighbor weight distances

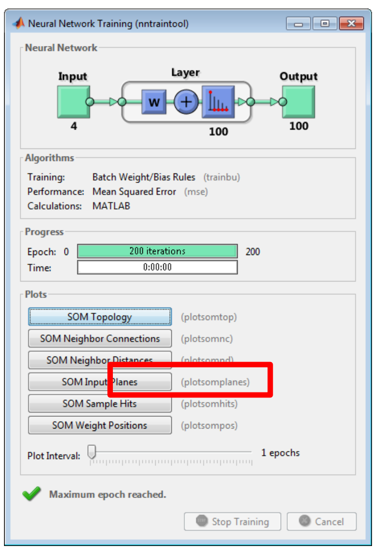
- Provide an insight of distinct groups in the data
- Blue hexagons represent the neurons
- Red lines connect neighboring neurons
- Colors indicate the distances between neurons
  - Darker colors represent larger distances, and lighter colors smaller distances

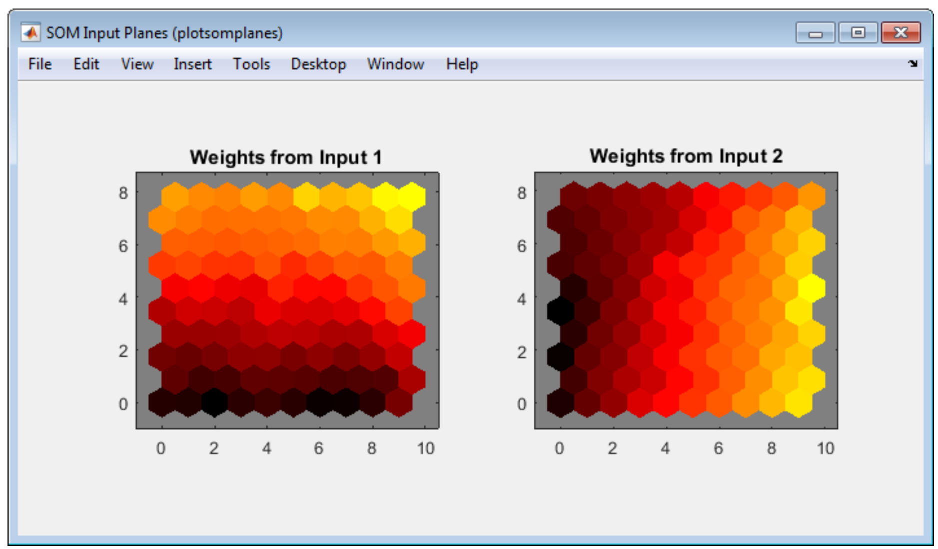




Analyzing SOM weight planes

- Visualization of the weights that connect each input to each of the neurons
- Shows a weight plane for each element of the input vector
- Darker colors represent larger weights, lighter colors smaller weights
- If the connection patterns of some inputs were very similar, the inputs can be considered highly correlated





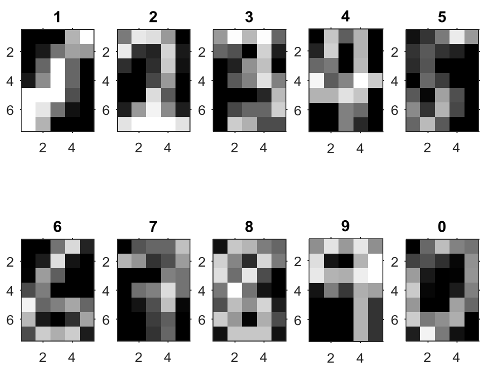
Exercises

7. Analysis of flower data
   - Load iris_dataset (included in Matlab)
   - Create and train a 10x10-node SOM with suitable parameters
   - Study the class separation using sample hits plots (Iris Setosa (samples 1-50), Iris Versicolour (samples 51-100) and Iris Virginica (samples 101-150))
   - Analyze SOM Neighbor Distances, does any evidence indicate that data are not from a single species?
   - Analyze SOM Weight Planes, Does any attribute seem particularly correlated?
8. Analysis unknown data
   - Load unknown_data, it contains a matrix and 5 points (https://homes.di.unimi.it/munoz/teaching.html)
   - Analyze the unknown_data using a SOM, in some suitable way
   - How many well-separated clusters are there in the data set?
   - Which data points are from the same cluster?

Exercises

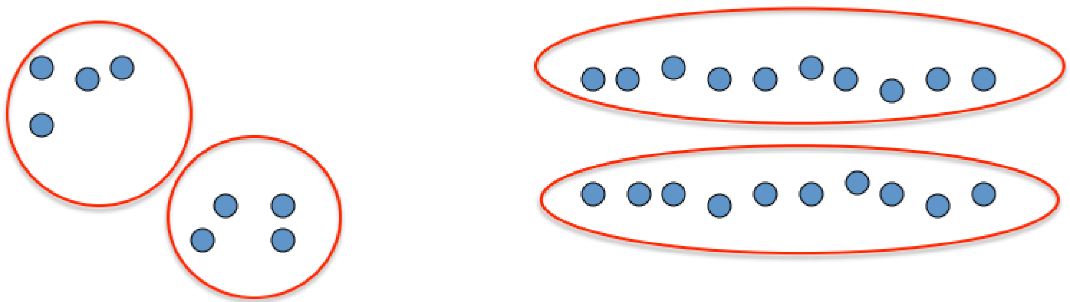
9. Analysis of hand-written digits
   - Analyze the digits data using a SOM, in some suitable way
   - Could you obtain separated clusters for different digits?
   - Load images from digits directory and targets from digit_names.mat (download from https://homes.di.unimi.it/munoz/teaching.html)
   - Useful code to create P and T

```
load('digits_names.mat');
files=dir('digits/*.bmp');
for i=1:numel(files)
   im=imread(['digits/' files(i).name]);
   P(:,i)=double(reshape(im,[1,35]));
   for j=1:numel(names)
      if strcmp(files(i).name,names{j})
        [~,T(i)]=max(targetsByName(:,j));
        break;
      end
   end
end
```



Clustering

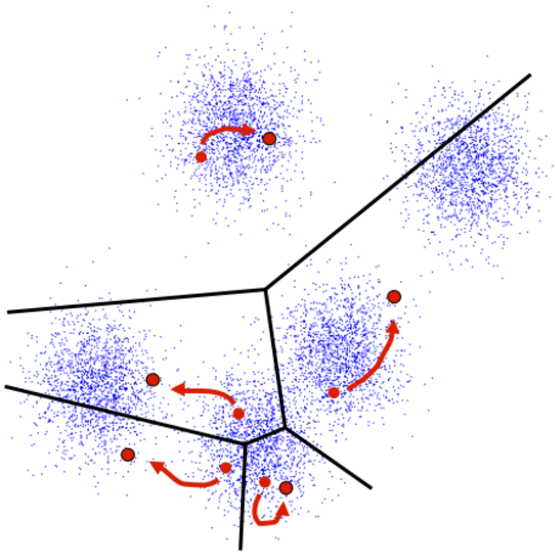
- Basic idea: group together similar instances
- Example: 2D point patterns



- What could "similar" mean?
  – One option: small Euclidean distance (squared)

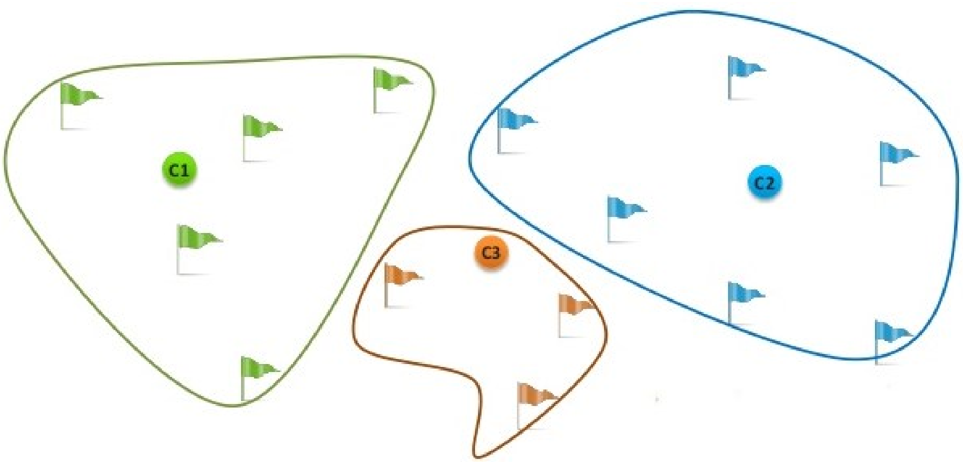
K-means clustering

- Iterative clustering algorithm
  – Initialize: pick *K* random points as cluster centers
  – Alternate:
    1. Assign data points to closest cluster center
    2. Change the cluster center to the average of its assigned points
- Stop when no points' assignments change



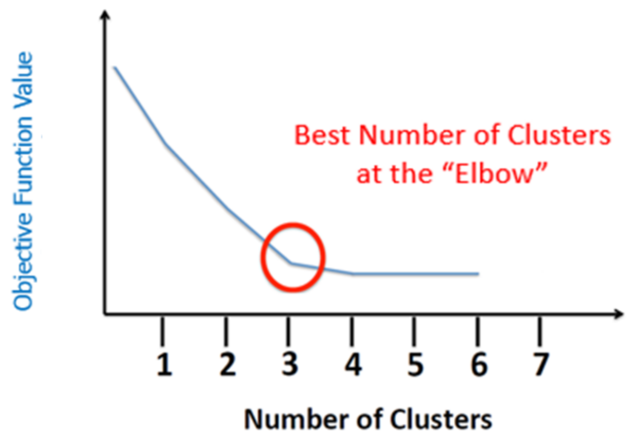
Example 4

- Help deciding where to place hospitals
  - We have a matrix with coordinates of emergency calls
  - Decide the best position for 3 hospitals that minimizes the distance from all the points of a particular cluster



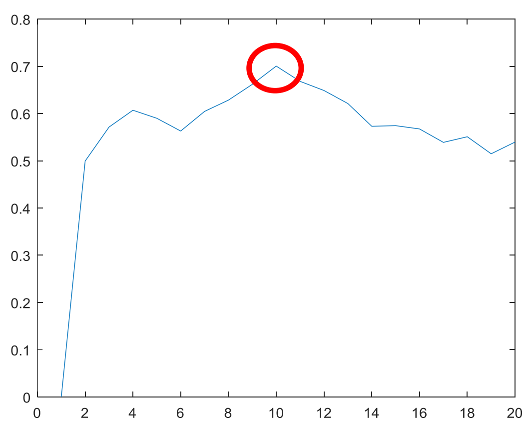
K-means: deciding number of clusters

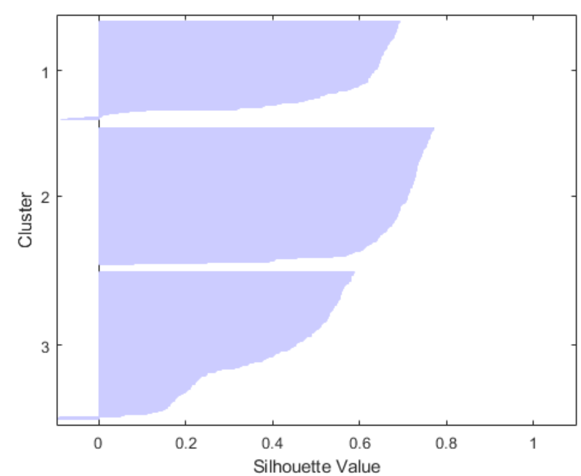
- K-means requires that a number of clusters *k* is decided a-priori
- How to find an optimal *k*
- Possibility 1:
  – Minimize objective function, elbow method



K-means: deciding number of clusters

- Possibility 2:
  – Use a cluster evaluation technique
  – Silhouette: a measure of how close each point in one cluster is to points in the neighboring clusters
  – Minimize mean Silhouette values





Exercises

10. Obtain the optimal number of hospitals in example 4
    - Try different values for *k*
    - Evaluate the solutions obtained using objective function

    [idx, C, **sumd**] = kmeans(points, numberOfClusters);

    - Evaluate the solutions obtained using mean Silhouette values

    [silh,h] = silhouette(points,idx);

    mean(silh)

    - Which value of k obtains better performance with each measure?

Fuzzy C-means

- Similar to K-means
- Uses concepts from the field of fuzzy logic and fuzzy set theory
- Objects are allowed to belong to more than one cluster
- Each object belongs to every cluster with some weight w_{ij}

Fuzzy C-means: algorithm

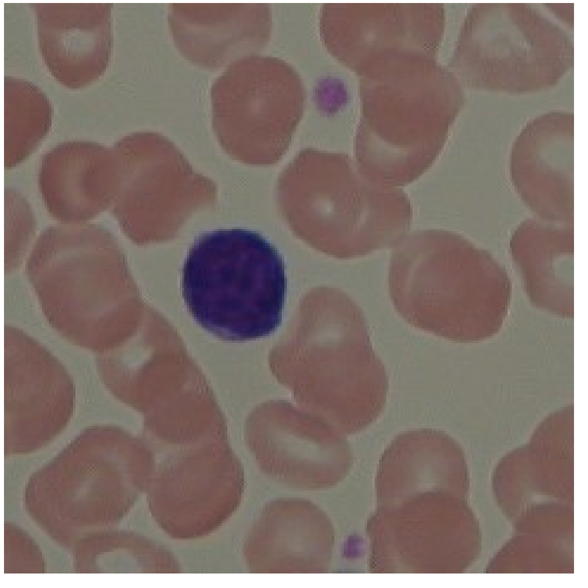
- Iterative algorithm:
  - Initialize: select an initial fuzzy pseudo-partition, i.e., assign values to all w_{ij}
  - Alternate:
    1. Compute the centroid of each cluster using the fuzzy partition
    2. Update the fuzzy partition, i.e, the w_{ij}
  - Stop when the centroids don’t change

Example 5

- Cluster foods according to preference correlations
  - 42 individuals were asked to order 15 breakfast items due to their preference
  - Cluster breakfast data into three clusters, to represent cluster membership in RGB color space
  - Analyze the visual results

Example 6

- Segment an image
  - Use fuzzy C-Means to separate background and objects from an image



Suggested readings

- A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
- A. K. Jain, R.P.W Duin, J. Mao, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.22, no.1, pp.4,37, Jan 2000.
- T. Kohonen, “The Self-Organizing map”, *Proceedings of the IEEE*, 78, pp. 1464-1480, 1990.
- B. Fritzke, “ A growing neural gas network learns topologies,” *Advances in Neural Information Processing Systems 7 (NIPS’94)*, MIT Press, Cambridge, MA, pp. 625-632, 1995.
- https://www.it.uu.se/edu/course/homepage/mil/vt11/labcourse/lab3.pdf