INF552 Machine Learning

Liyue Fan
liyuefan@usc.edu
Integrated Media Systems Center
University of Southern California

Class Communication - Update

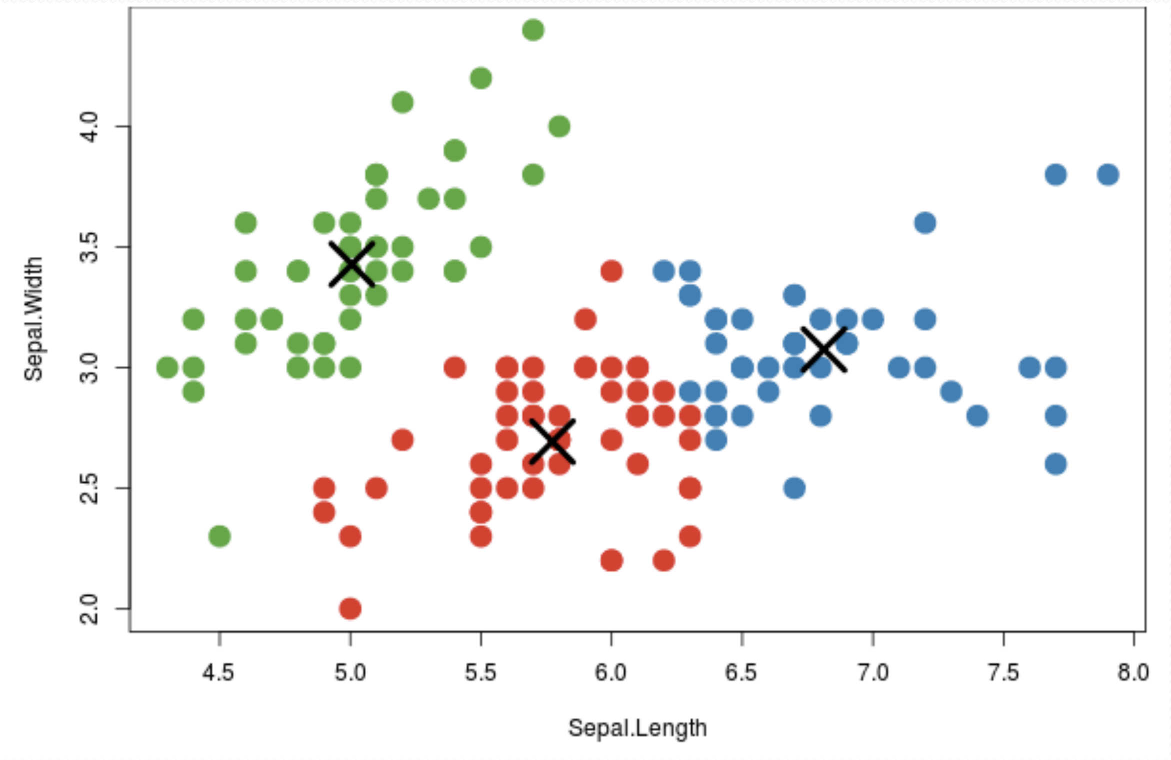
- Discussion board on Blackboard
- Kien's office hours
  - Tuesdays 2-4pm, Thursdays 4-5pm @ SAL Open Lab
  - Email: kien.nguyen@usc.edu
- My office hours
  - Wednesdays 3:30-4:30pm @ PHE 335
- Talk to me before class or during breaks
- Email me if all the above fails
  - usually reply within 48 hours

Clustering

Some slides by E Alpaydin

k-Means Clustering

- Find k best representations of the data set $\mathcal{X}$
  - Iris dataset from UCI



k-Means Clustering

- Find *k* reference vectors (prototypes/codebook vectors/ codewords) which best represent data
- Reference vectors, $\boldsymbol{m}_j$, j =1,...,k
- Use nearest (most similar) reference:

$$\left\|\mathbf{x}^t - \mathbf{m}_i\right\| = \min_j \left\|\mathbf{x}^t - \mathbf{m}_j\right\|$$

- Best reference vectors -> Min. Reconstruction error

$$E\left(\left\{\mathbf{m}_i\right\}_{i=1}^k \middle| \mathcal{X}\right) = \sum_t \sum_i b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|$$

$$b_i^t = \begin{cases} 1 & \text{if } \left\|\mathbf{x}^t - \mathbf{m}_i\right\| = \min_j \left\|\mathbf{x}^t - \mathbf{m}_j\right\| \\ 0 & \text{otherwise} \end{cases}$$

k-Means Clustering

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$

Repeat

For all $\boldsymbol{x}^t \in \mathcal{X}$

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

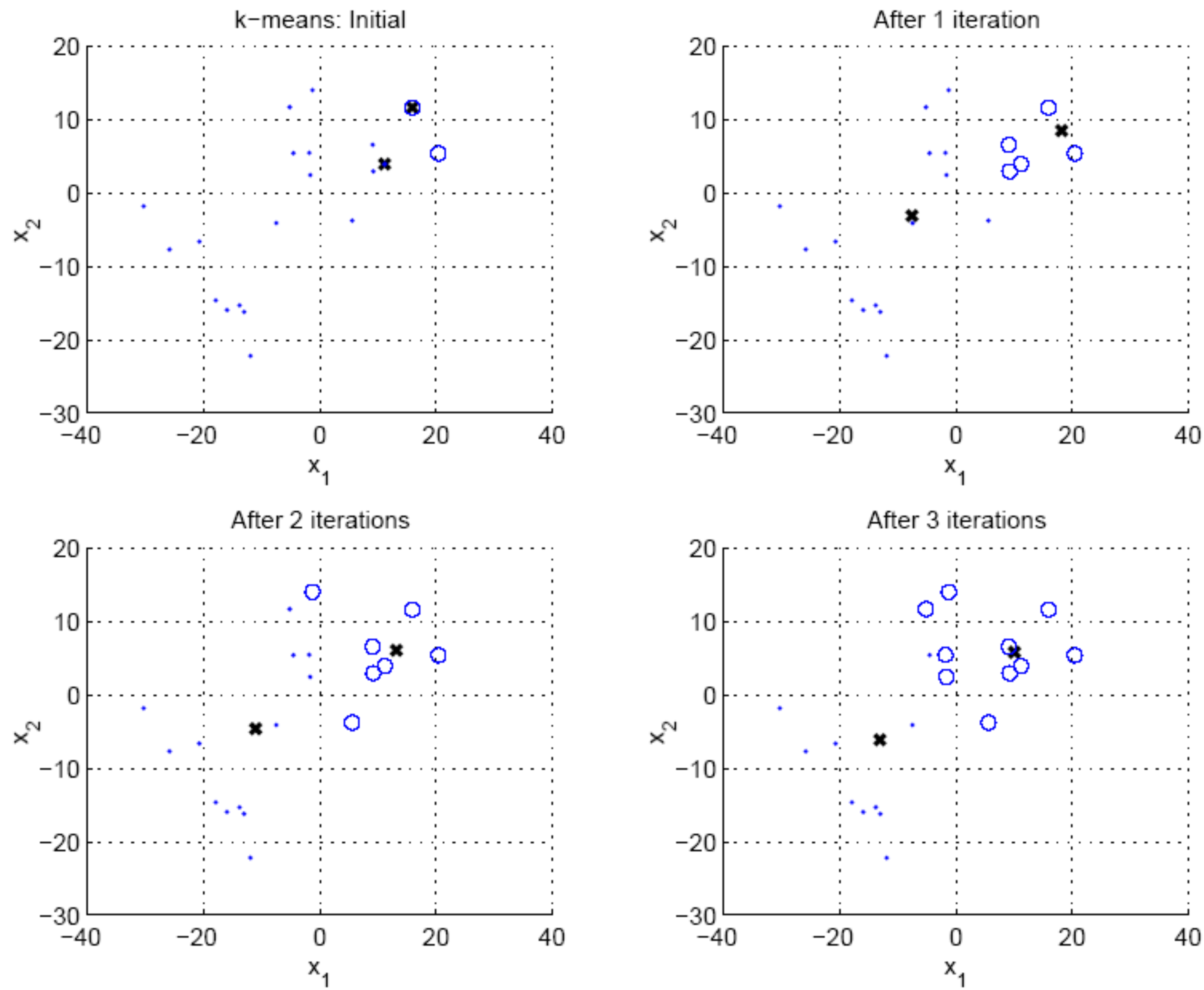
For all $\boldsymbol{m}_i, i = 1, \ldots, k$

$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

Until $\boldsymbol{m}_i$ converge

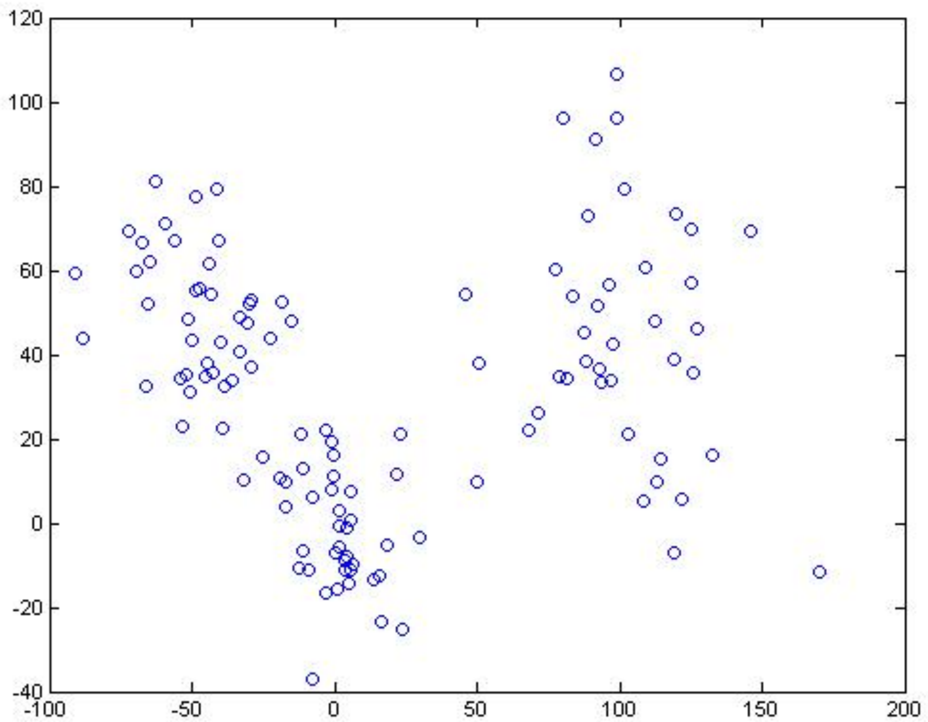
K-Means Clustering

- Choose a number of clusters *k*
- Initialize cluster centers $m_1, \ldots m_k$
  - Either pick *k* data points and set cluster centers to these points
  - Or could randomly assign points to clusters and take means of clusters
- For each data point, compute the cluster center it is closest to (using some distance measure) and assign the data point to this cluster
- Re-compute cluster centers (mean of data points in cluster)
- Stop when there are no new re-assignments



k−means: Initial
After 1 iteration
After 2 iterations
After 3 iterations
x_1
x_2

K-Means Clustering (cont.)



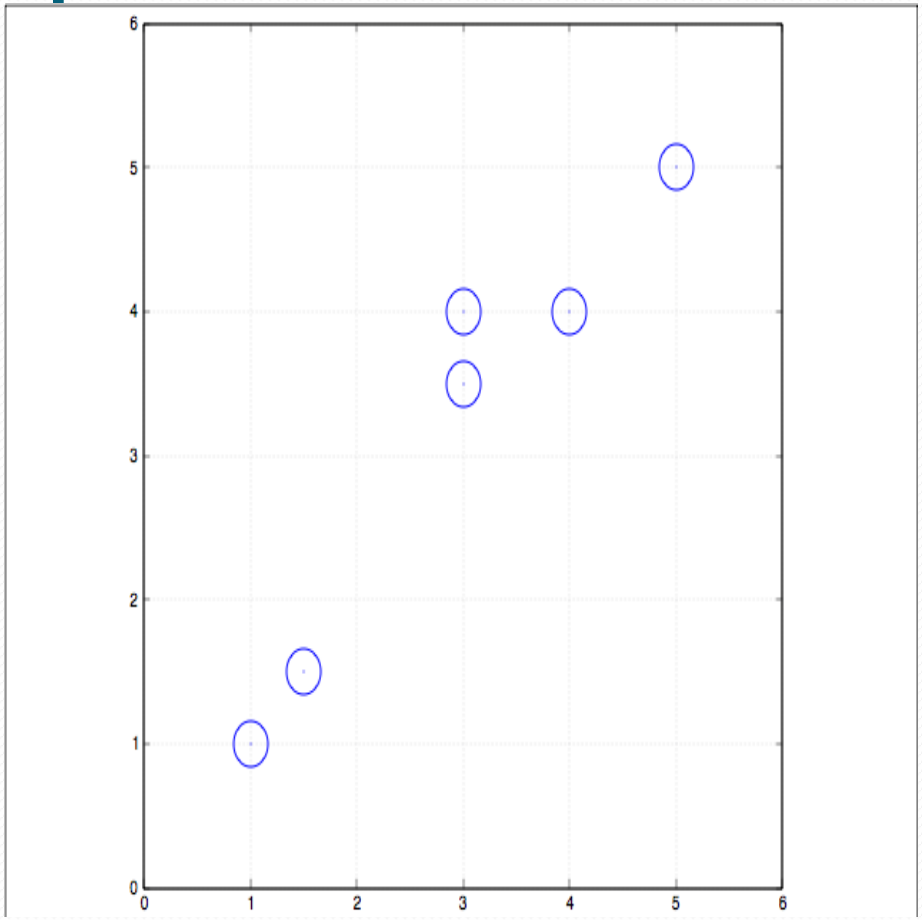
How many clusters do you think there are in this data?

K-Means Clustering (cont.)

$$k = 2$$

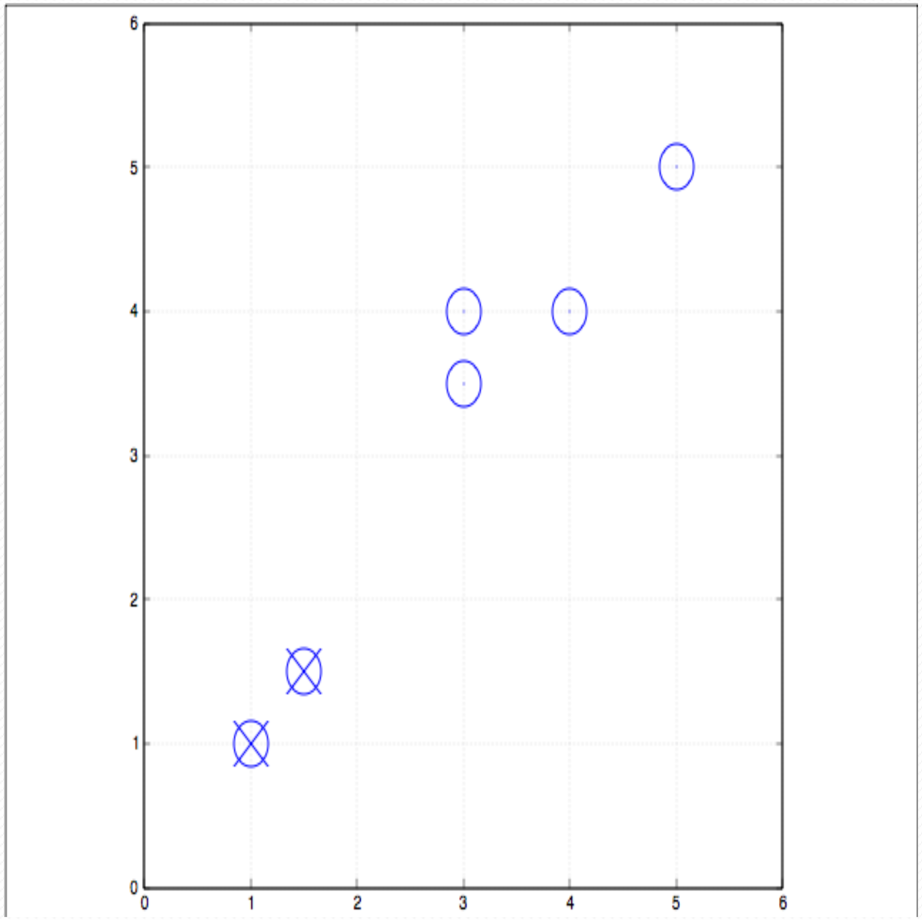
Example: Start points

X1	X2
1	1
1.5	1.5
5	5
3	4
4	4
3	3.5



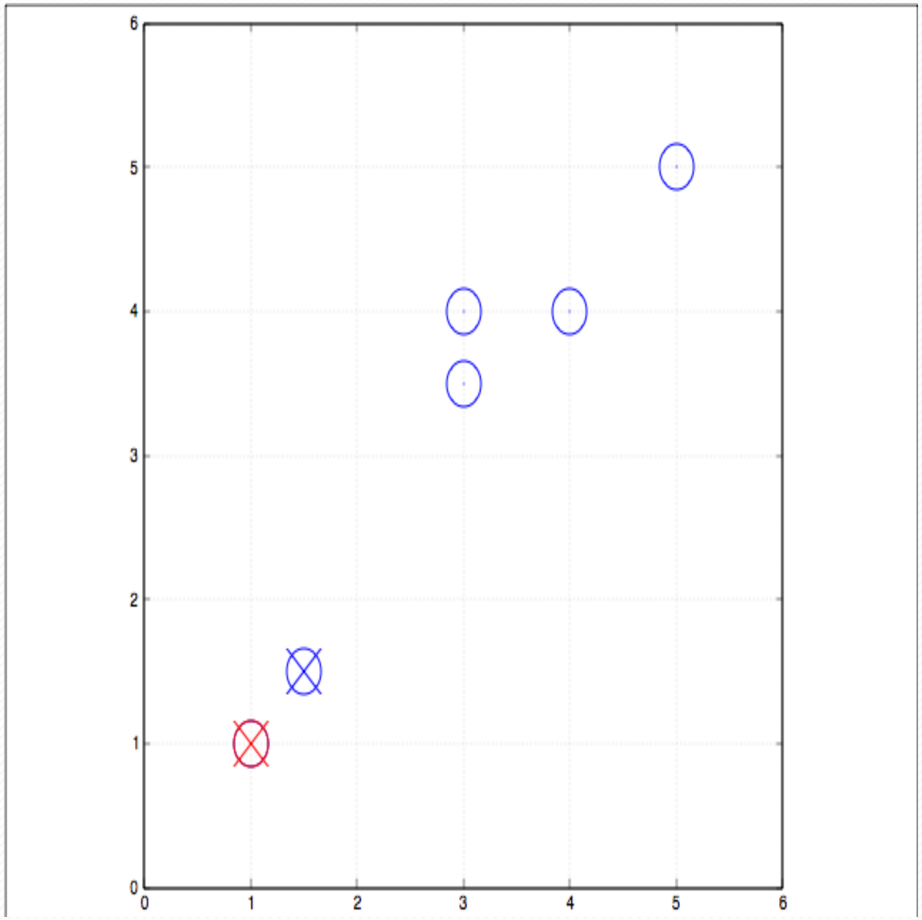
Initialize centroids

X1	X2	Centroid 1 (1, 1)	Centroid 2 (1.5, 1.5)
1	1		
1.5	1.5		
5	5		
3	4		
4	4		
3	3.5		



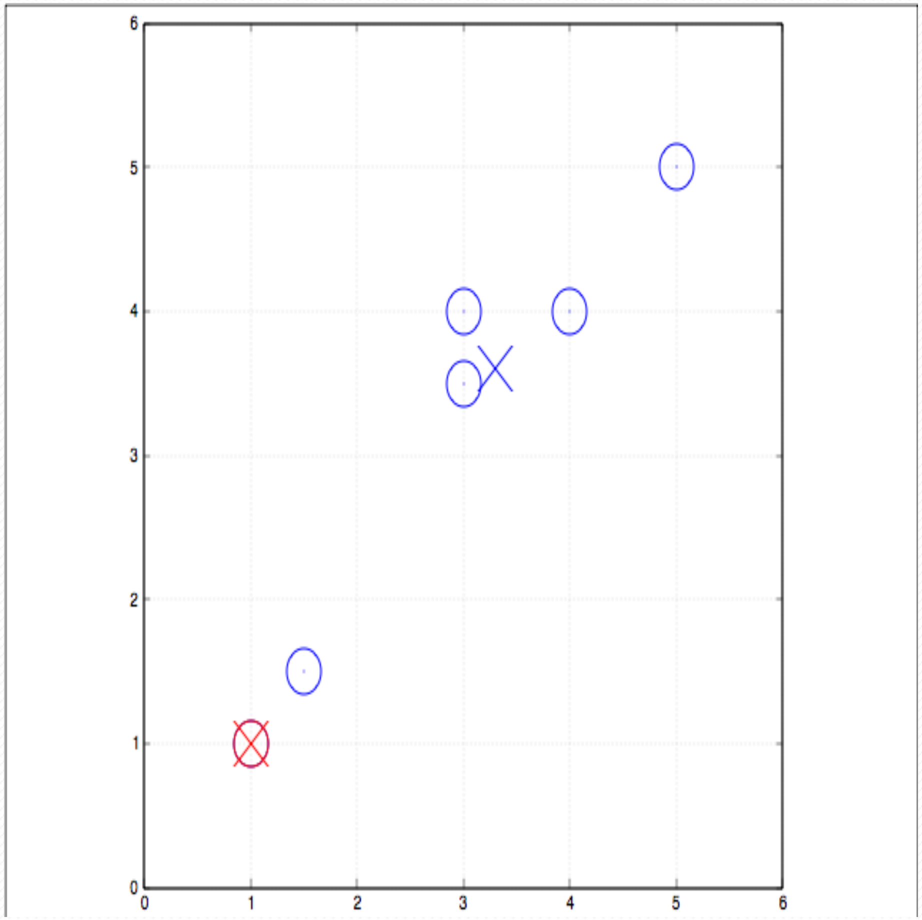
Cluster assignments

X1	X2	Centroid 1 (1, 1)	Centroid 2 (1.5, 1.5)
1	1	0	0.707
1.5	1.5	0.707	0
5	5	5.656	4.949
3	4	3.605	2.915
4	4	4.242	3.535
3	3.5	3.201	2.5



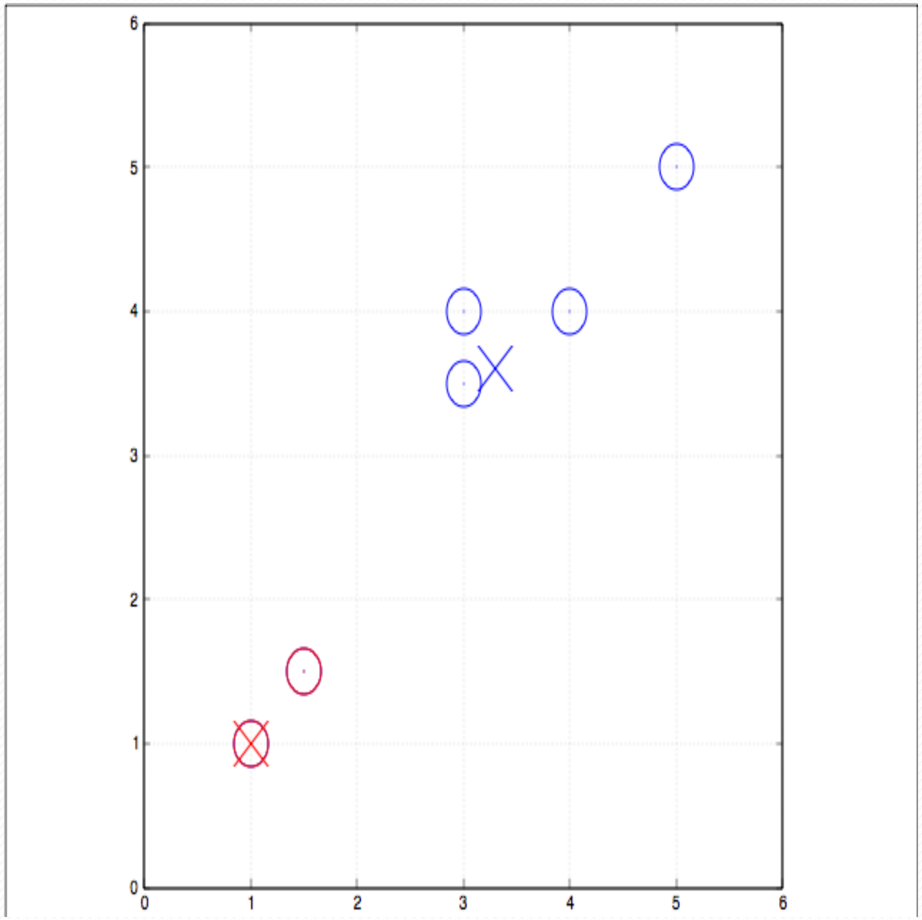
Move centroids

X1	X2	Centroid 1 (1, 1)	Centroid 2 (3.3, 3.6)
1	1		
1.5	1.5		
5	5		
3	4		
4	4		
3	3.5		



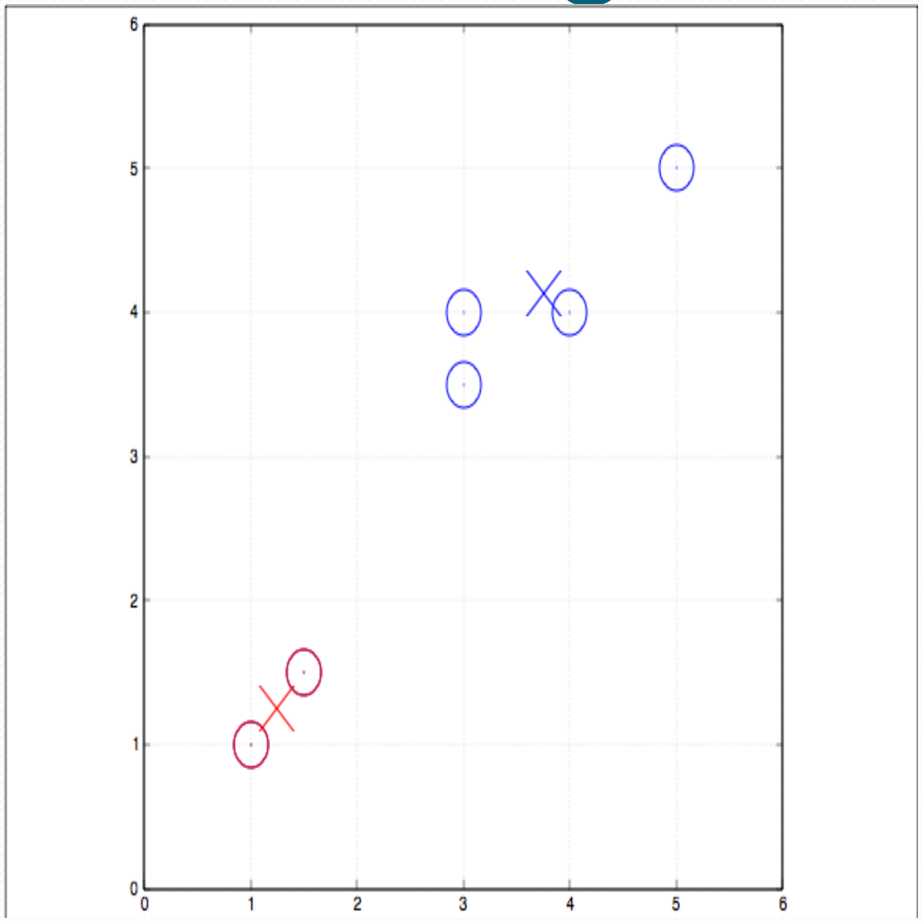
Cluster assignments

X1	X2	Centroid 1 (1, 1)	Centroid 2 (3.3, 3.6)
1	1	0	3.471
1.5	1.5	0.707	2.765
5	5	5.656	2.202
3	4	3.605	0.499
4	4	4.242	0.806
3	3.5	3.201	0.316



Move centroids - Converged

X1	X2	Centroid 1 (1.25, 1.25)	Centroid 2 (3.75, 4.125)
1	1		
1.5	1.5		
5	5		
3	4		
4	4		
3	3.5		



K-Means Clustering Issues

- Random initialization means that you may get different clusters each time
- Data points are assigned to only one cluster (hard assignment)
- Implicit assumptions about the "shapes" of clusters (why?)
- You have to pick the number of clusters, *k*

Choosing *k* - Empirically

- Defined by the application, e.g., image quantization
- Plot data (after PCA) and check for clusters
- Incremental (leader-cluster) algorithm: Add one at a time until "elbow" (reconstruction error/log likelihood/ intergroup distances)
- Manually check for meaning

Determining the "correct" number of clusters

- We'd like to have a measure of cluster quality Q and then try different values of k until we get an optimal value for Q
- But, since clustering is an unsupervised learning method, we can't really expect to find a "correct" measure Q…
- So, once again there are different choices of Q and our decision will depend on what dissimilarity measure we're using and what types of clusters we want

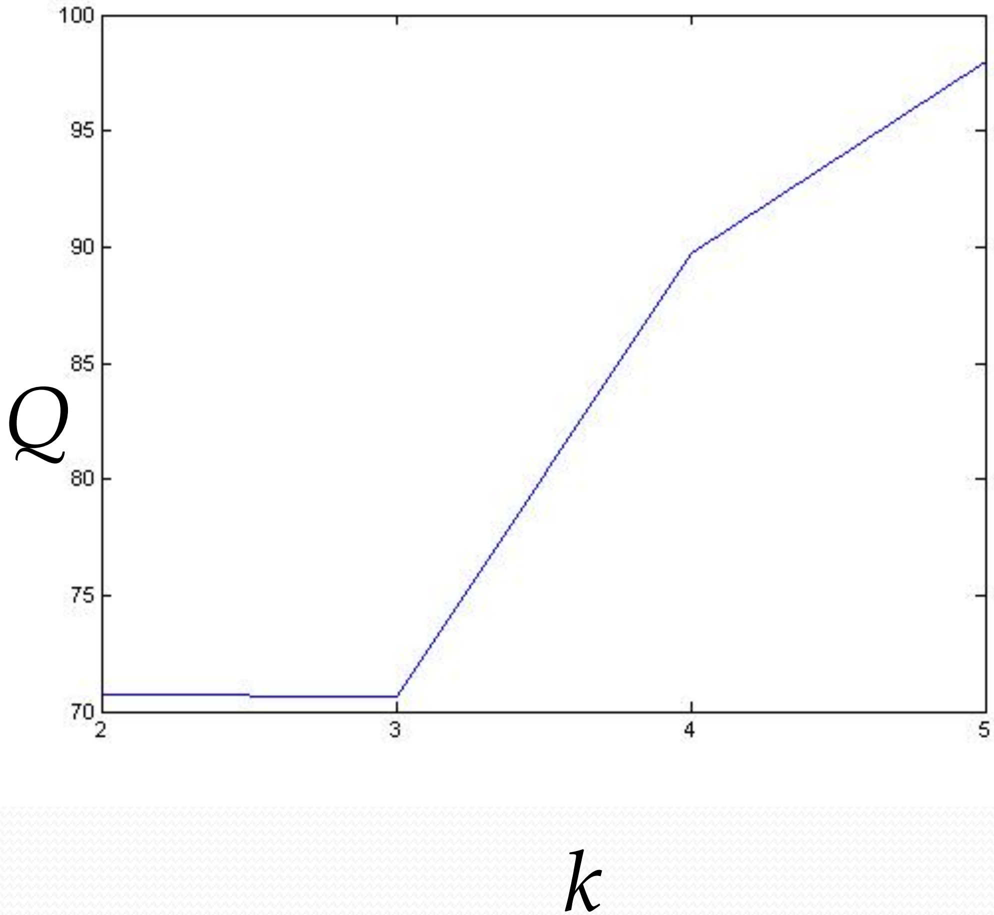
Cluster Quality Measures

- A measure that emphasizes cluster tightness or homogeneity:

$$Q = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} d(\boldsymbol{x}, \mu_i)$$

- $|C_i|$ is the number of data points in cluster i
- Q will be small if (on average) the data points in each cluster are close

Cluster Quality (cont.)



This is a plot of the Q measure for k-means clustering on the data shown earlier.

How many clusters do you think there actually are?

Cluster Quality (cont.)

- The *Q* measure given before takes into account homogeneity within clusters, but not separation between clusters
- Other measures try to combine these two characteristics (i.e., the Davies-Bouldin measure see https://en.wikipedia.org/wiki/Davies%E2%80%93Bouldin_index)
- An alternate approach is to look at cluster stability:
  - Add random noise to the data many times and count how many pairs of data points no longer cluster together
  - How much noise to add? Should reflect estimated variance in the data

After Clustering

- Dimensionality reduction methods find correlations between features and group features
- Clustering methods find similarities between instances and group instances
- Allows knowledge extraction through

    number of clusters,

    prior probabilities,

    cluster parameters, i.e., center, range of features.

Example: CRM, customer segmentation

Hierarchical Clustering

- Cluster based on similarities/distances
- Distance measure between instances $\boldsymbol{x}^r$ and $\boldsymbol{x}^s$

Minkowski (L_p) (Euclidean for p = 2)

$$d_m\left(\mathbf{x}^r,\mathbf{x}^s\right)=\left[\sum_{j=1}^{d}\left(x_j^r-x_j^s\right)^p\right]^{1/p}$$

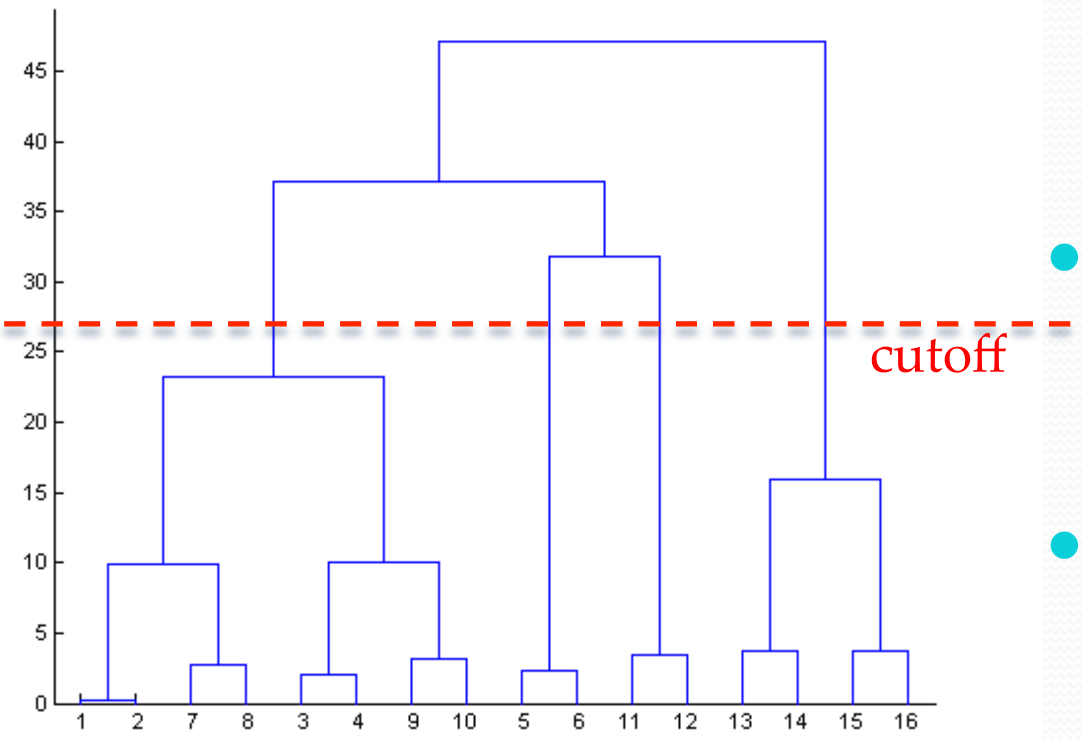
City-block distance

$$d_{cb}\left(\mathbf{x}^r,\mathbf{x}^s\right)=\sum_{j=1}^{d}\left|x_j^r-x_j^s\right|$$

Hierarchical Agglomerative Clustering

- We start with every data point in a separate cluster
- We keep merging the most similar two clusters until we have one big cluster left
- This is called a *bottom-up* or *agglomerative* method

Hierarchical Clustering (cont.)



- This produces a binary tree or ***dendrogram***
- The final cluster is the root and each data item is a leaf
- The height of the bars indicate how close the items are

Linkage Options

- Distance between two groups G_i and G_j:
  - Single-link:

$$d(G_i, G_j) = \min_{\mathbf{x}^r \in G_i, \mathbf{x}^s \in G_j} d(\mathbf{x}^r, \mathbf{x}^s)$$

  - Complete-link:

$$d(G_i, G_j) = \max_{\mathbf{x}^r \in G_i, \mathbf{x}^s \in G_j} d(\mathbf{x}^r, \mathbf{x}^s)$$

  - Average-link, centroid

Linkage in Hierarchical Clustering

- We already know about distance measures between data items, but what about between a data item and a cluster or between two clusters?
- We just treat a data point as a cluster with a single item, so our only problem is to define a ***linkage*** method between clusters
- As usual, there are lots of choices…

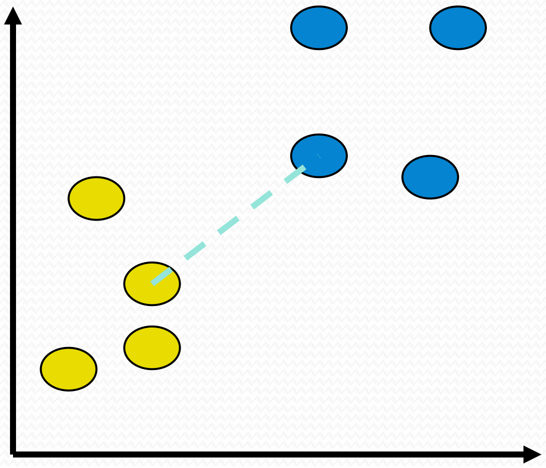
Average Linkage

- Defined as the average of all pairwise distances between points in the two clusters

- "Centroid linkage" is defined as follows:
  - Each cluster c_i is associated with a mean vector μ_i which is the mean of all the data items in the cluster
  - The distance between two clusters c_i and c_j is then just $d(\mu_i, \mu_j)$

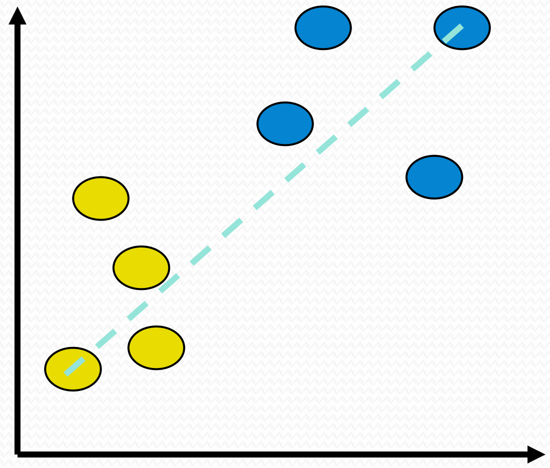
Single Linkage

- The minimum of all pairwise distances between points in the two clusters
- Tends to produce long, "loose" clusters



Complete Linkage

- The maximum of all pairwise distances between points in the two clusters
- Tends to produce very tight clusters



Working Example

- Data points: A-F

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

- Pairwise Distance (Adjacency) Matrix:

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Working Example

- Merging D and F, how to update the distance between clusters with Single Linkage?

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Working Example

- Merging A and B, update again:

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Working Example

- Merging {D, F} and E, update again:

Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

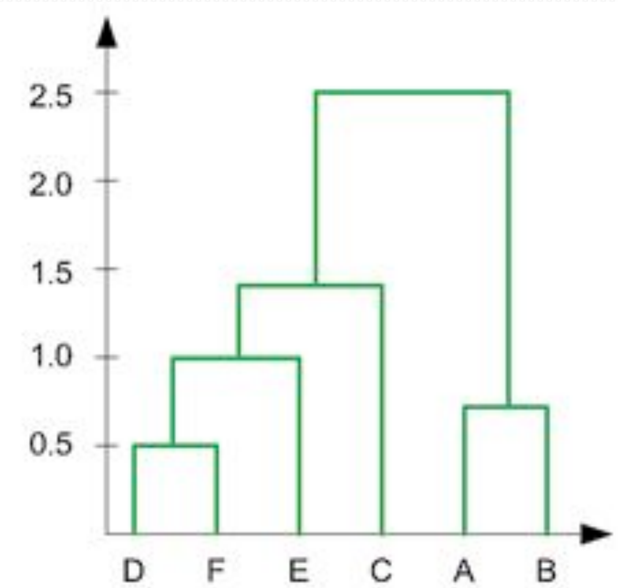
- Merging {{D,F},E} with C, update again:

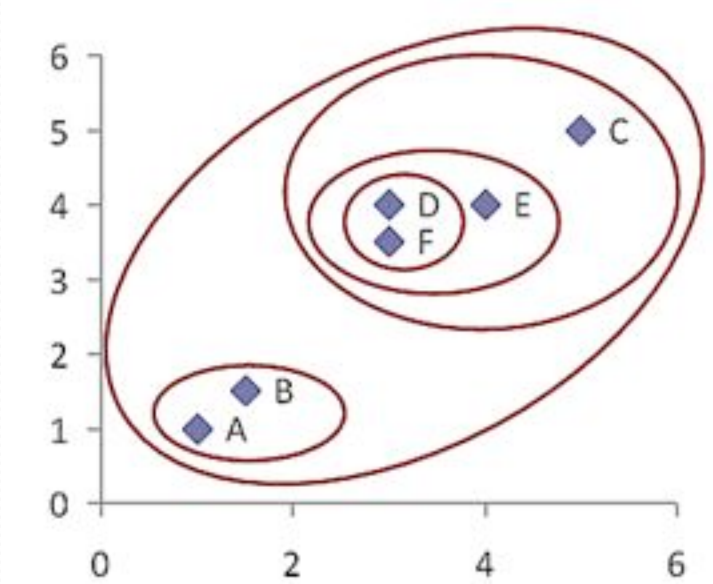
Min Distance (Single Linkage)

Dist	(A,B)	(D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

Working Example

- Merging {A,B} with {{{D,F},E},C} -> one cluster





Hierarchical Clustering Issues

- Distinct clusters are not produced – sometimes this can be good, if the data has a hierarchical structure w/o clear boundaries
- There are methods for producing distinct clusters, but these usually involve specifying somewhat arbitrary cutoff values
- What if data doesn't have a hierarchical structure? Is HC appropriate?

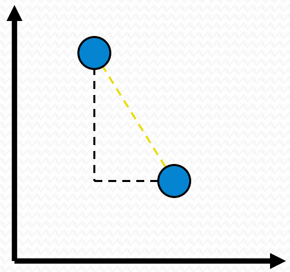
How do we define “similarity”?

- Recall that the goal is to group together “similar” data – but what does this mean?
- No single answer – it depends on what we want to find or emphasize in the data; this is one reason why clustering is an “art”
- The similarity measure is often more important than the clustering algorithm used – don’t overlook this choice!

(Dis)similarity measures

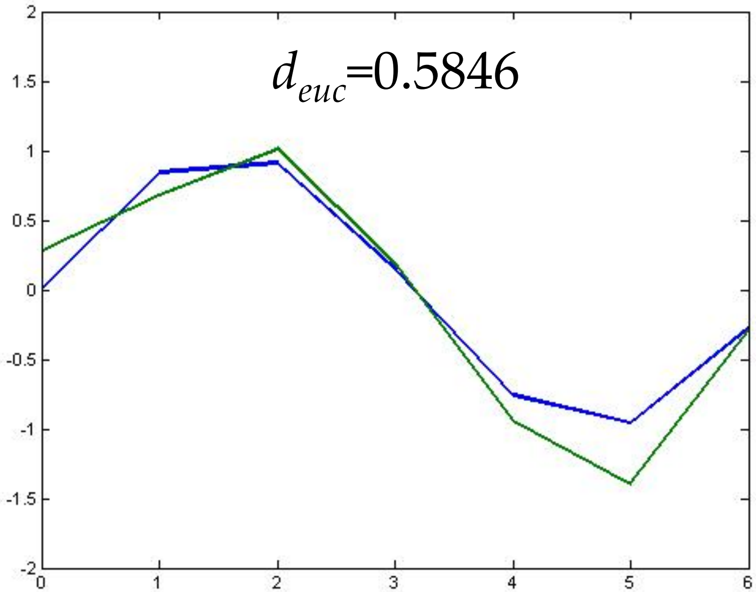
- Instead of talking about similarity measures, we often equivalently refer to dissimilarity measures
- A dissimilarity measure as a function f(**x**,**y**) such that f(**x**,**y**) > f(**w**,**z**) if and only if **x** is less similar to **y** than **w** is to **z**
- This is always a *pair-wise* measure

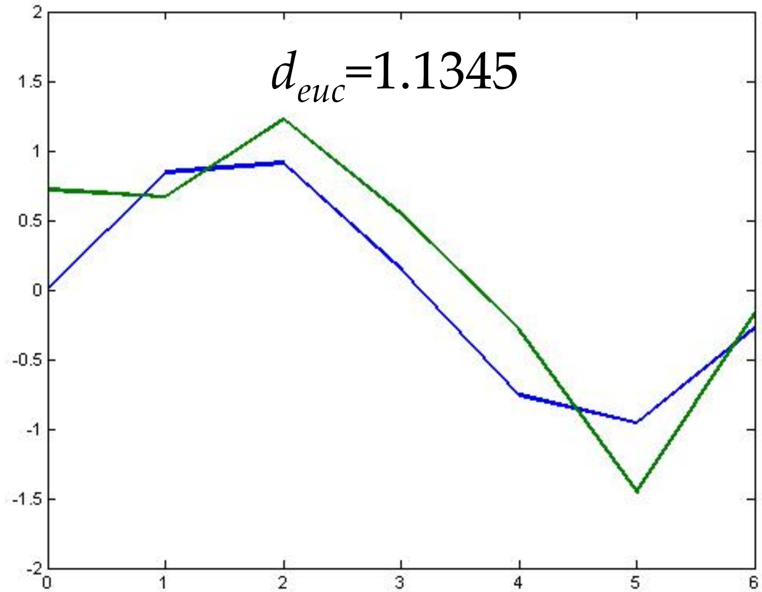
Euclidean distance

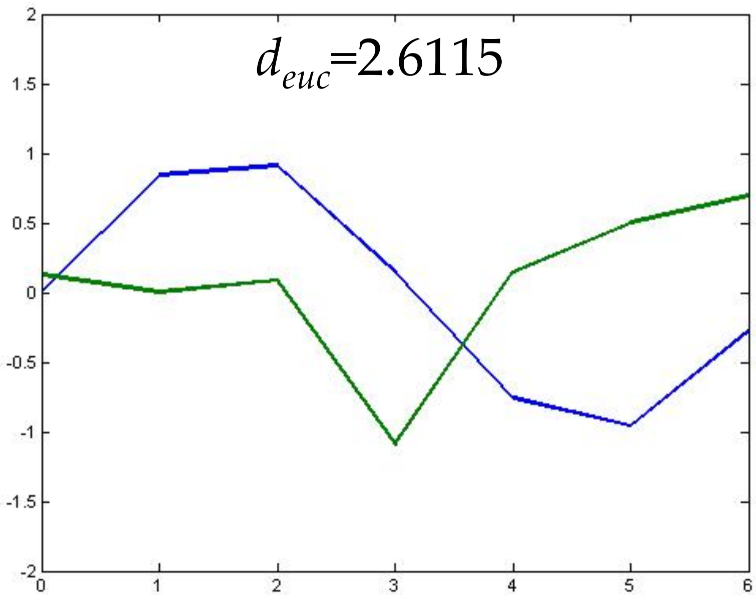


$$d_{euc}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

- Here *n* is the number of dimensions in the data vector. For instance:
  - Number of time-points (when clustering time series, trajectories, etc.)

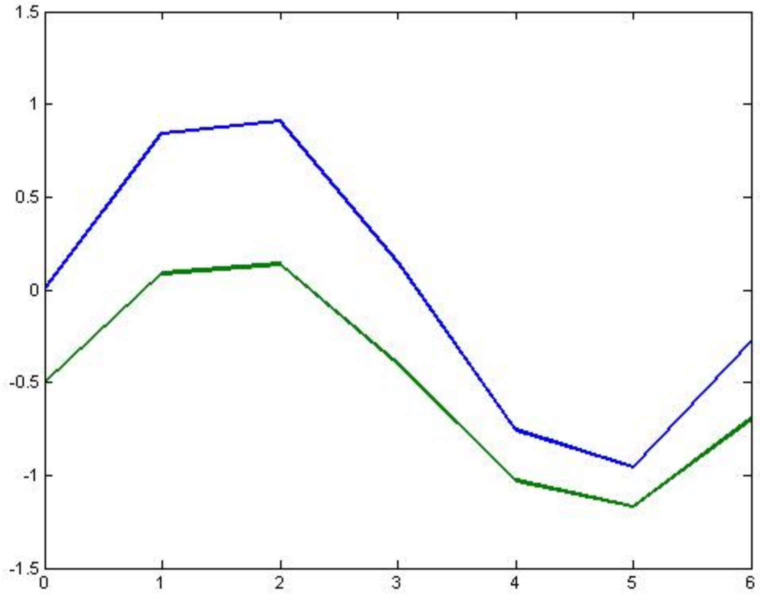




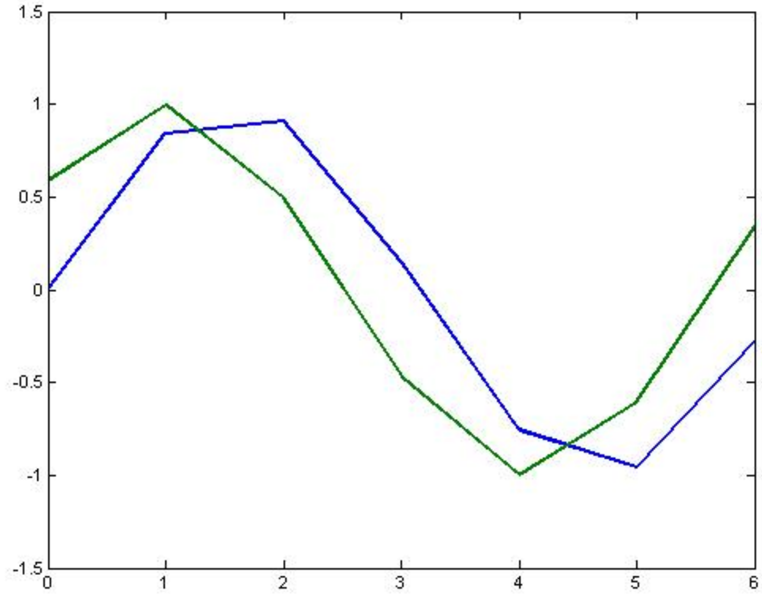


These examples of Euclidean distance match our intuition of dissimilarity pretty well…

d_{euc}=1.41



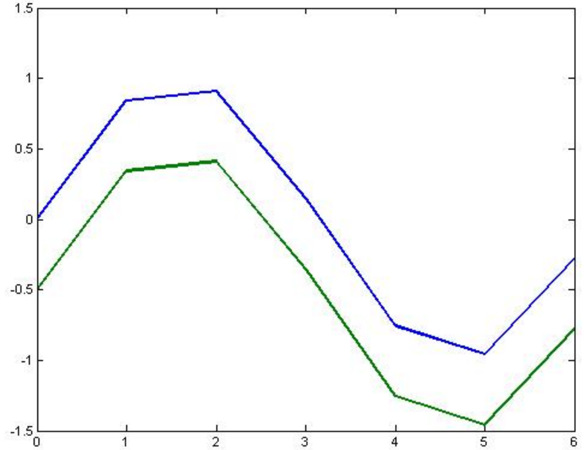
d_{euc}=1.22

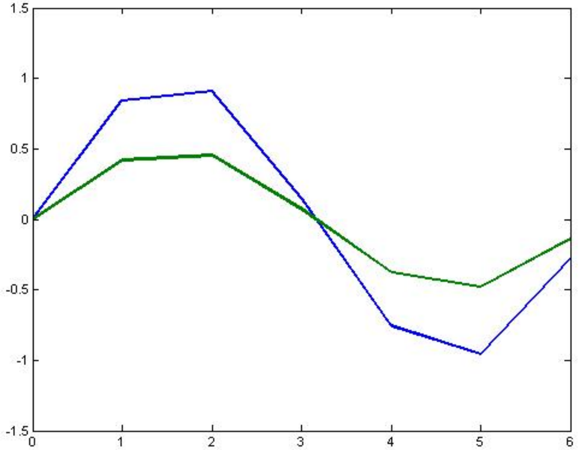


…But what about these?

Correlation

- We might care more about the overall shape of time series rather than the actual magnitudes
- That is, we might want to consider time series similar when they are "up" and "down" together
- When might we want this kind of measure? What experimental issues might make this appropriate?





Pearson Linear Correlation

$$\rho(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$\bar{x} = \frac{1}{n}\sum_{i}^{n} x_i, \bar{y} = \frac{1}{n}\sum_{i}^{n} y_i$$

- We're shifting the time series down (subtracting the means) and scaling by the standard deviations (i.e., making the data have mean = 0 and std = 1)

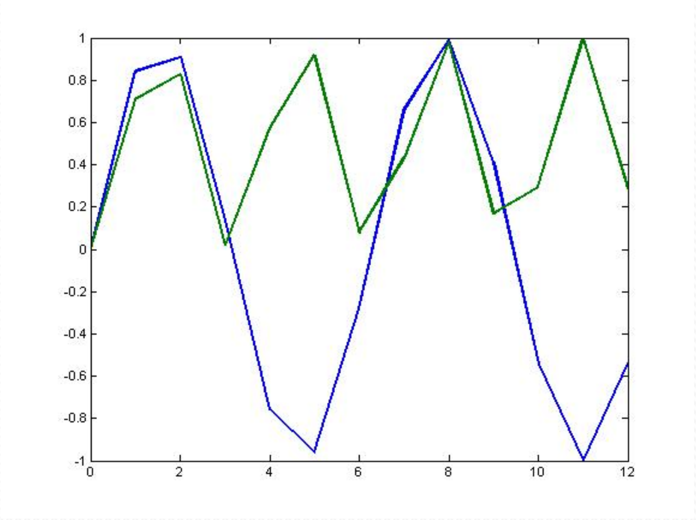
Pearson Linear Correlation

- Pearson linear correlation (PLC) is a measure that is invariant to scaling and shifting (vertically) of the values
- Always between –1 and +1 (perfectly anti-correlated and perfectly correlated)
- This is a similarity measure, but we can easily make it into a dissimilarity measure:

$$d_p = \frac{1 - \rho(\mathbf{x}, \mathbf{y})}{2}$$

PLC (cont.)

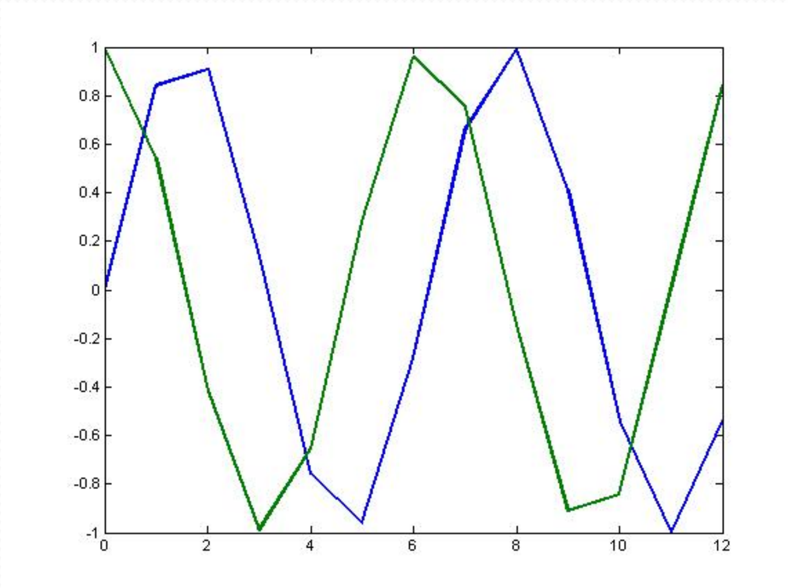
- PLC only measures the degree of a *linear* relationship between two data sets/sequences!
- If you want to measure other relationships, there are many other possible measures (for more examples)

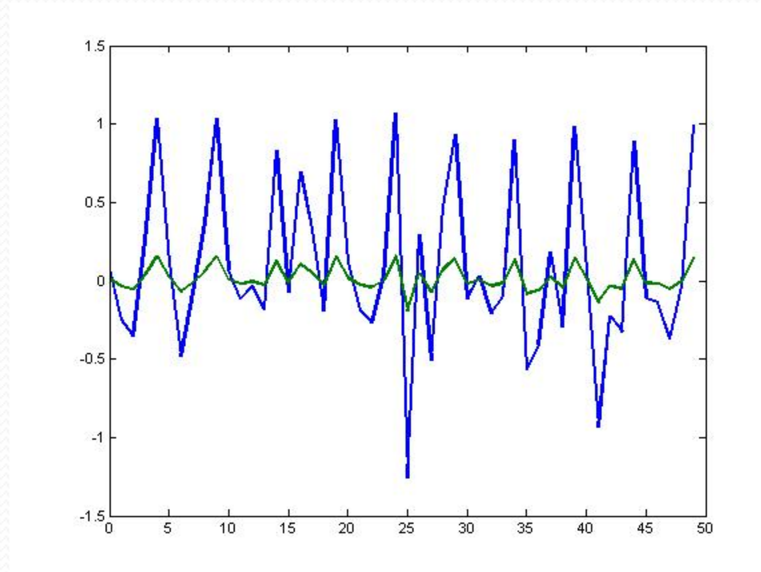


$\rho = 0.0249$, so $d_p = 0.4876$

The green curve is the square of the blue curve – this relationship is not captured with PLC

More correlation examples





What do you think the correlation is here?

How about here?

We'll come back to dissimilarity metrics later!

Presentations

- Read the articles before you start preparing!
  - Notes on Presenting a Paper http://web.stanford.edu/~jacksonm/present.pdf
  - Tips for Successful Academic Paper Presentations http://graddiv.ucsc.edu/about/blogs/grad-deans-blog/11-2013.1.html
- Timing: 20 mins, approximately 20 slides
- Practice

next week

- papers posted on blackboard

- readings are challenging
  - keep in mind that a paper might bring potential project ideas
  - discuss its weaknesses as if you were to improve the paper for your project