Principles of Computer Science II

Working with Data Sets

Ioannis Chatzigiannakis

Sapienza University of Rome

Lecture 24

Analysis of Data

- Viewing and analyzing vast amounts of biological data in its unstructured entirety can be perplexing.
- It is easier to interpret data if it is organized into clusters that combine similar (i.e., related) data points.

Analyzing data from DNA microarray experiments (expression analysis – i.e., determining which genes are switched "on" or "off" under certain conditions of interest).

Building and understanding phylogenetic (evolutionary) trees based on genomic or other data.

Microarray Analysis

- What do newly sequenced genes do?
- Simply comparing new gene sequences to known DNA sequences often does not reveal the function of a new gene.
- For 40% of sequenced genes, functionality cannot be ascertained by comparing to sequences of other known genes.
- It is easier to interpret data if it is organized into clusters that combine similar (i.e., related) data points.

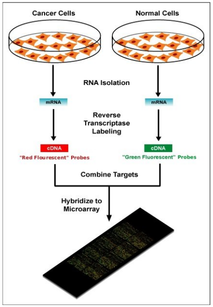
Microarrays and expression analysis

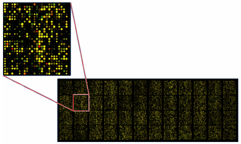
- Microarrays measure activity (expression level) of genes under varying conditions and/or points in time.
- Expression level is estimated by measuring amount of mRNA for that particular gene:
  - A gene is active if it is being transcribed.
  - More mRNA usually indicates more gene activity.

A Microarray Experiment

- Produce cDNA from mRNA (cDNA is more stable)
- Label cDNA with a fluorescent dye or biotin for detection
- Different color labels are available to compare many samples at once
- Wash cDNA over the microarray containing thousands of high density probes that hybridize to complementary strands in the sample and immobilize them on the surface.
- For biotin-labeled samples, stain with the biotin-specific fluorescently labeled antibody
- Read the microarray, using a laser or a high-resolution CCD
- Illumination reveals transcribed/co-expressed genes

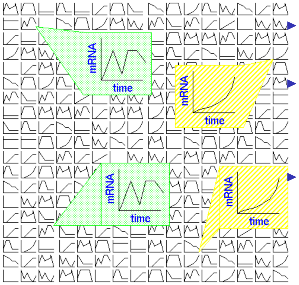
A Microarray Experiment





- Green: expressed only in control
- Red: expressed only in an experimental cell
- Yellow: equally expressed in both samples
- Black: NOT expressed in either control or sample

A Microarray Experiment



- Boxes: Gene's expression over time
- Track sample over period of time: see how gene expression changes.
- Track two different samples under same conditions: see differences in gene expression.

Microarray Data Transformation

- Microarray data are usually transformed into a (relative, normalized) intensity matrix
- Can also be represented as a bit matrix (log_2 of relative intensity)
- The intensity matrix allows biologists to infer correlations between different genes (even if they are dissimilar) and to understand how genes functions might be related
- Care must be taken to normalize the data appropriately, e.g. different time points can come from different arrays.

Microarray Data Intensity Matrix

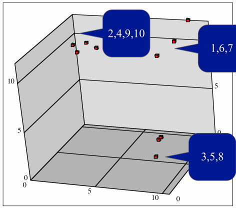
- Which genes are similar?
- What defines co-expression?
- How to measure the distance/similarity?

Gene	Time 1	Time 2	Time 3
1	10	8	10
2	10	0	9
3	4	8.5	3
4	9.5	0.5	8.5
5	4.5	8.5	3
6	10.5	9	12
7	5	8.5	11
8	2.7	8.7	2
9	9.7	2	9
10	10.2	1	9.2

Euclidean Distance in D-dimensions

$$D(x,y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$$

Finding Similar Genes



	1	2	3	4	5	6	7	8	9	10
1		8.1	9.2	7.7	8.9	2.3	5.1	10.9	6.1	7.0
2	8.1		12.0	0.9	11.8	9.5	10.1	13.3	2.0	1.0
3	9.2	12.0		11.2	0.5	11.1	8.1	1.7	10.5	11.5
4	7.7	0.9	11.2		10.9	9.2	9.5	12.5	1.6	1.1
5	8.9	11.8	0.5	10.9		10.8	8.0	2.1	10.3	11.3
6	2.3	9.5	11.1	9.2	10.8		5.6	12.7	7.7	8.5
7	5.1	10.1	8.1	9.5	8.0	5.6		9.3	8.3	9.3
8	10.9	13.3	1.7	12.5	2.1	12.7	9.3		12.0	12.9
9	6.1	2.0	10.5	1.6	10.3	7.7	8.3	12.0		1.1
10	7.0	1.0	11.5	1.1	11.3	8.5	9.3	12.9	1.1	

PAIRWISE DISTANCES

	1	6	7	2	4	9	10	3	5	8
1	0.0	2.3	5.1	8.1	7.7	6.1	7.0	9.2	8.9	10.9
6	2.3	0.0	5.6	9.5	9.2	7.7	8.5	11.1	10.8	12.7
7	5.1	5.6	0.0	10.1	9.5	8.3	9.3	8.1	8.0	9.3
2	8.1	9.5	10.1	0.0	0.9	2.0	1.0	12.0	11.8	13.3
4	7.7	9.2	9.5	0.9	0.0	1.6	1.1	11.2	10.9	12.5
9	6.1	7.7	8.3	2.0	1.6	0.0	1.1	10.5	10.3	12.0
10	7.0	8.5	9.3	1.0	1.1	1.1	0.0	11.5	11.3	12.9
3	9.2	11.1	8.1	12.0	11.2	10.5	11.5	0.0	0.5	1.7
6	8.9	10.8	8.0	11.8	10.9	10.3	11.3	0.5	0.0	2.1
8	10.9	12.7	9.3	13.3	12.5	12.0	12.9	1.7	2.1	0.0

REARRANGED DISTANCES

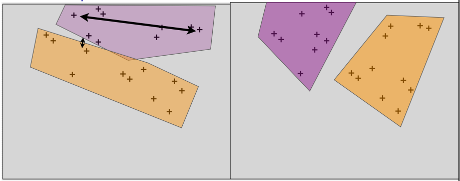
The Clustering Problem

- Motivation: Find patterns in a sea of data
- Input
  - A (large) number of datapoints: N
  - A measure of distance between any two data points d_{ij}
- Output
  - Groupings (clustering) of the elements into K (the number can be user-specified or automatically determined) 'similarity' classes
  - Sometimes there is also an objective measure that the obtained clustering seeks to minimize.

Clustering Principles

- Homogeneity – elements of the same cluster are maximally close to each other.
- Separation – elements in separate clusters are maximally far apart from each other.
- One is actually implied by the other (in many cases).
- Generally it is a hard problem.
  - Clustering in 2 dimensions looks easy
  - Clustering small amounts of data looks easy
  - High-dimensional spaces look different – Almost all pairs of points are at about the same distance

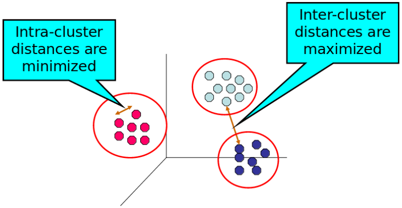
Some Examples



- Both principles are violated
- Points in the same cluster are far apart
- Points in different cluster are close

- More reasonable assignment.
- We need to use an objective function to optimize cluster assignment.

Intra/Inter Cluster Distances


Intra-cluster distances are minimized
Inter-cluster distances are maximized

- Suitably select distance metric.
- Maximize Inter-cluster distances.
- Minimize Intra-cluster distances.

Distance Measures

- Each clustering problem is based on some kind of "distance" between points.
- Two major classes of distance measure:
  1. Euclidean
  2. Non-Euclidean
- A Euclideanspace has some number of real-valued dimensions.
  - There is a notion of "average" of two points.
  - A Euclidean distance is based on the locations of points in such a space.
- A Non-Euclidean distance is based on properties of points, but not their "location" in a space.

Axioms of a Distance Measure

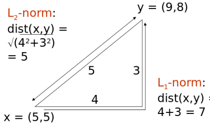
d is a distance measure if it is a function from pairs of points to real numbers such that:

1. $d(x,y) > 0$
2. $d(x,y) = 0$ *iff* $x = y$
3. $d(x,y) = d(y,x)$
4. $d(x,y) < d(x,z) + d(z,y)$ (triangle inequality)

Some Euclidean Distances

L_2 norm: $d(x,y)$ = square root of the sum of the squares of the differences between x and y in each dimension.
The most common notion of "distance".

L_1 norm: sum of the differences in each dimension.
Manhattan distance = distance if you had to travel along coordinates only.



Some Non-Euclidean Distances

Jaccard distance for sets = 1 minus ratio of sizes of intersection and union.

Cosine distance = angle between vectors from the origin to the points in question.

Edit distance = number of inserts and deletes to change one string into another.

Jaccard Distance for Sets

Example: $p_1 = 10111$; $p_2 = 10011$.
Size of intersection = 3; size of union = 4, Jaccard similarity (not distance) = $\frac{3}{4}$.
$d(x,y) = 1$–(Jaccard similarity) = $\frac{1}{4}$.

Why JD is a distance measure?

1. $d(x,x) = 0$ because $x \cap x = x \cup x$
2. $d(x,y) = d(y,x)$ because union and intersection are symmetric
3. $d(x,y) \geq 0$ because $|x \cap y| \leq |x \cup y|$
4. $d(x,y) < d(x,z) + d(z,y)$ more difficult...
$\left(1 - \frac{|x \cap z|}{|x \cup z|}\right) + \left(1 - \frac{|y \cap z|}{|y \cup z|}\right) \geq 1 - \frac{|x \cap y|}{|x \cup y|}$

Edit Distance

The edit distance of two strings is the number of inserts and deletes of characters needed to turn one into the other. Equivalently:

$$d(x,y) = |x| + |y| - 2|LCS(x,y)|$$

LCS = longest common subsequence = any longest string obtained both by deleting from x and deleting from y.

Example

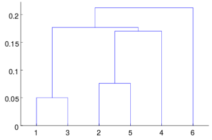
- x = abcde ; y = bcduve.
- Turn x into y by deleting a, then inserting u and v after d. Edit distance = 3.
- Or, LCS(x,y) = bcde.
- Note: $|x| + |y| - 2|LCS(x,y)| = 5 + 6 - 2 \times 4 = 3 =$ edit dist

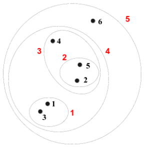
Why Edit Distance is a Distance Measure?

1. $d(x,x) = 0$ because 0 edits suffice.
2. $d(x,y) = d(y,x)$ because insert/delete are inverses of each other
3. $d(x,y) \geq 0$ no notion of negative edits
4. $d(x,y) < d(x,z) + d(z,y)$ Triangle inequality: changing x to z and then to y is one way to change x to y.

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram – A tree like diagram that records the sequences of merges or splits





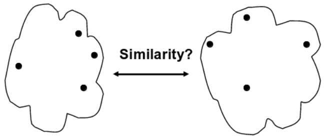
Agglomerative Hierarchical Clustering

- Initially, each point is a cluster
- Repeatedly combine the two "nearest" clusters into one

```
Compute the proximity matrix
Let each data point be a cluster
Repeat
        Merge the two closest clusters
        Update the proximity matrix
Until only a single cluster remains
```

- Key operation is the computation of the proximity of two clusters
- Different approaches to defining the distance between clusters distinguish the different algorithms

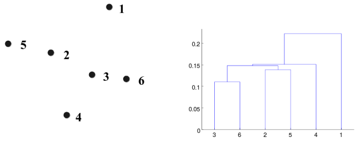
How to define Inter-cluster similarity?



- Minimum – based on the two most similar (closest) points in the different clusters
- Maximum – based on the two least similar (most distant) points in the different clusters
- Group Average

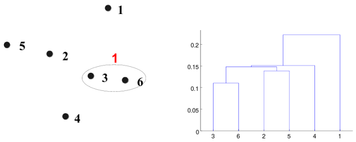
Minimum – Example

Minimum – based on the two most similar (closest) points in the different clusters



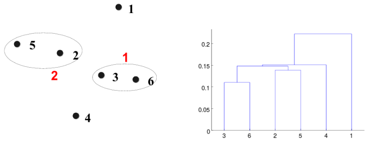
Minimum – Example

Minimum – based on the two most similar (closest) points in the different clusters



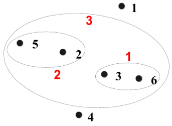
Minimum – Example

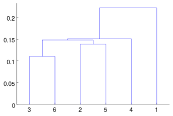
Minimum – based on the two most similar (closest) points in the different clusters



Minimum – Example

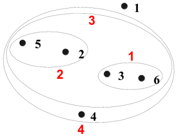
Minimum – based on the two most similar (closest) points in the different clusters

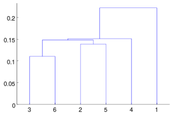




Minimum – Example

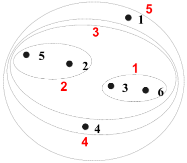
Minimum – based on the two most similar (closest) points in the different clusters

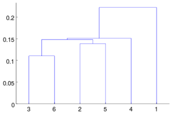




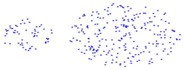
Minimum – Example

Minimum – based on the two most similar (closest) points in the different clusters

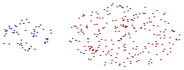




Minimum – Strength

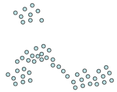


Original Points

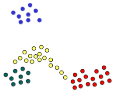


Two Clusters

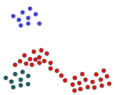
Minimum – Limitations



Original Points



Four clusters



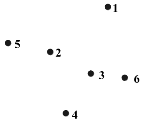
Three clusters:

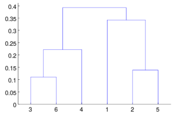
The yellow points got wrongly merged with the red ones, as opposed to the green one.

Sensitive to noise and outliers

Maximum – Example

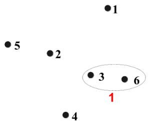
Maximum – based on the two least similar (most distant) points in the different clusters

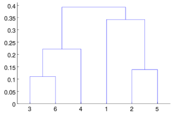




Maximum – Example

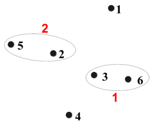
Maximum – based on the two least similar (most distant) points in the different clusters

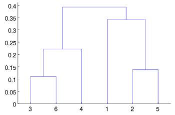




Maximum – Example

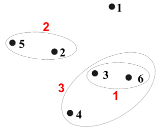
Maximum – based on the two least similar (most distant) points in the different clusters

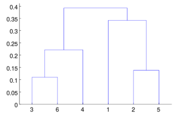




Maximum – Example

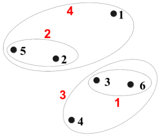
Maximum – based on the two least similar (most distant) points in the different clusters

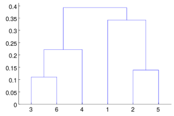




Maximum – Example

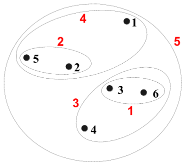
Maximum – based on the two least similar (most distant) points in the different clusters

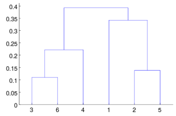




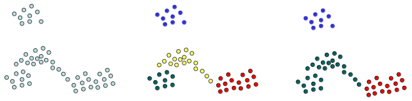
Maximum – Example

Maximum – based on the two least similar (most distant) points in the different clusters





Maximum – Strength



Original Points

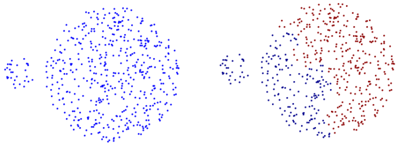
Four clusters

Three clusters:

The yellow points get now merged with the green one.

Less susceptible respect to noise and outliers

Maximum – Limitations



K-means Algorithm

- Developed and published in Applied Statistics by Hartigan and Wong, 1979.
- Many variations have been proposed since then.
- Standard/core function of R, Python, Matlab, ...
- Assumes Euclidean space/distance

The aim of the K-means algorithm is to divide M points in N dimensions into k clusters so that the within-cluster sum of squares is minimized.

$$\min._{C_1,\ldots,C_K} \sum_{k=1}^{k} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

Cluster Initialization

- Start by picking k, the number of clusters
- Initialize clusters by picking one point per cluster

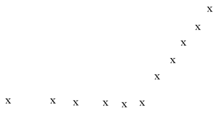
Example: Pick one point at random, then $k-1$ other points, each as far away as possible from the previous points

Populating Clusters

1. For each point, place it in the cluster whose current centroid it is nearest
2. After all points are assigned, update the locations of centroids of the k clusters
3. Reassign all points to their closest centroid
   - Sometimes moves points between clusters
4. Repeat 2 and 3 until convergence

Convergence: Points do not move between clusters and centroids stabilize

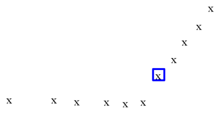
A Simple Example



x … data point
□ … centroid

Clusters after round 1

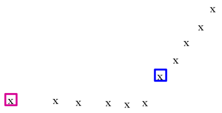
A Simple Example



x … data point
□ … centroid

Clusters after round 1

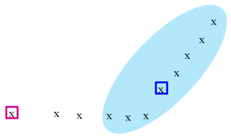
A Simple Example



x … data point
□ … centroid

Clusters after round 1

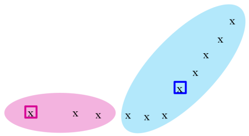
A Simple Example



x … data point
□ … centroid

Clusters after round 1

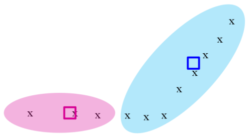
A Simple Example



x … data point
□ … centroid

Clusters after round 1

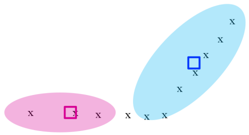
A Simple Example



x … data point
□ … centroid

Clusters after round 2

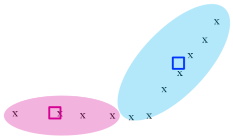
A Simple Example



x … data point
□ … centroid

Clusters after round 2

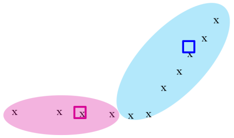
A Simple Example



x … data point
□ … centroid

Clusters after round 2

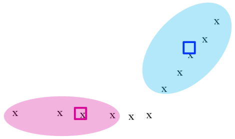
A Simple Example



x … data point
🟩 … centroid

Clusters at the end

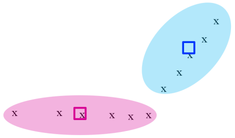
A Simple Example



x … data point
🟩 … centroid

Clusters at the end

A Simple Example

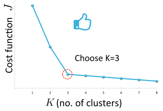


x … data point
🟩 … centroid

Clusters at the end

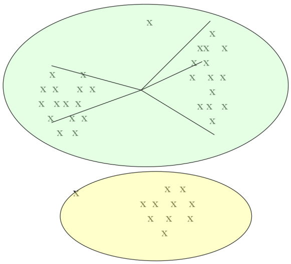
How to select k?

- We use the elbow method to determine the optimum number of clusters.
- Try different k, looking at the change in the average distance to centroid as k increases.
- Average falls rapidly until right k, then changes little.



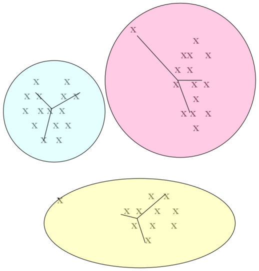
Selection of k – an example

Too few;
many long
distances
to centroid.



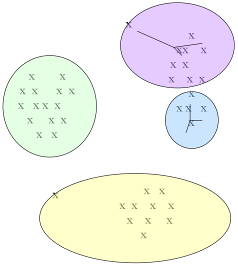
Selection of k – an example

Just right;
distances
rather short.



Selection of k – an example

Too many;
little improvement
in average
distance.



Loading the Iris dataset

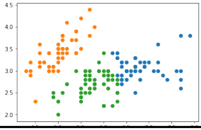
```
import pandas as pd

data = pd.read_csv('iris.csv',
                   names=['slength', 'swidth',
                          'plength', 'pwidth', 'name'])
```

One-dimensional clustering

```
values = data[['slength']]

from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3, init='random')

kmeans.fit(values)

centroids = model.cluster_centers_

c = kmeans.predict(values)
```

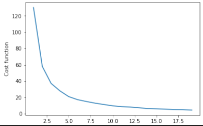
Two-dimensional clustering

```
kmeans = KMeans(n_clusters=3, init='random')
values = data[['slength', 'swidth']]
kmeans.fit(values)
labels = kmeans.predict(values)
values["clusters"] = labels

import matplotlib.pyplot as plt
for k in range(0,3):
    plt.scatter(values[values.clusters==k][['slength']],
                values[values.clusters==k][['swidth']])
plt.show()
```



Examining the number of clusters

```
sd = {}
for k in range(1,20):
    modelk = KMeans(n_clusters=k)
    modelk.fit(values)
    sd[k] = modelk.inertia_

plt.figure()
plt.plot(list(sd.keys()), list(sd.values()))
plt.xlabel("Number of clusters")
plt.ylabel("Cost function")
plt.show()
```



5th Assignment

- https://www.rosalind.info/
  - Complete the following challenges:
    prot, splc, tran, hamm, tree, pdst, sseq, lcsq, orf, perm, grph, inod, edit, edta, glob, mult
  - http://rosalind.info/problems/{challenge}
- Create a GitHub repository and upload the code for each exercise.
- Email `ichatz@diag.uniroma1.it`
  Subject: [PCS2] Homework 5
  A .zip or a .tar.gz file with your python solutions, for all challenges.
  Also send your account user account link:
  http://rosalind.info/users/{username}
- **Deadline: 8/January/2021, 23:59**