CE213 Artificial Intelligence – Lecture 16

Clustering

Unsupervised learning – Learning from unlabeled data

It is different from decision tree induction and error back-propagation learning in neural networks due to the unavailability of class labels or desired output in the training sample data.

A Brief Review of Machine Learning Concepts and Methods Taught in the Past Weeks

KEY COMPONENTS OF MACHINE LEARNING:

a task and an associated performance measure,

a learning environment or sample dataset,

a model (structure + parameters),

a learning algorithm.

Decision Tree Induction: supervised structural learning

Decision tree; Information gain based best attribute selection.

Learning in Artificial Neural Networks: supervised learning (mostly parametric)

MP neuron model, neuron model with logistic function,

multilayer feedforward neural network;

Hebb rule, Perceptron rule, Delta rule, generalised Delta rule,

error back-propagation learning algorithm.

What is Clustering?

THE TASK

Given a set of ***unlabelled*** training samples:

- Find a good way of partitioning the training samples into classes/groups.
- Construct a representation model that enables the class of any new sample to be determined.

Although the two subtasks are logically distinct, they are usually performed together.

Terminology

Statisticians call this ***clustering***.

Neural network researchers usually call it ***unsupervised learning***,

but unsupervised learning is more than clustering (e.g., data compression by machine learning, learning in generative adversarial network (GAN)).

The Basic Problem in Clustering – Performance Measure

Classification learning programs are successful if the predictions they make are correct, i.e., if they agree with the externally defined class labels.

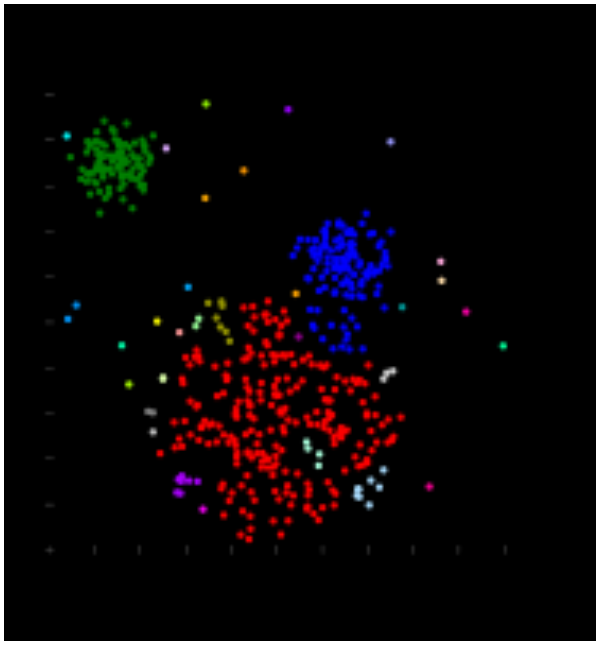
In clustering, *there is no* ***externally defined notion of correctness***.

However, there are many ways in which a training dataset could be partitioned.

Some of these are better than others.

e.g., how to partition the data shown in the figure on the right?

In your opinion, how many clusters are in this dataset?



Partitioning Criteria

Common sense suggests that members of a class should resemble each other more than they resemble members of other classes (or clusters).

Therefore, a good partition should:

- maximise similarity within classes (**Criterion 1**)
- minimise similarity between classes (**Criterion 2**)

Note that this implies the existence of a similarity metric.

Are these two criteria enough to identify good partitions?

No.

Consider the partitioning in which every item is assigned to its own class.

Such a partition would be of no use, but meet the above criteria.

This suggests a further criterion:

- minimise the number of classes created (**Criterion 3**)

Clearly there will be a trade off between this and the other two criteria.

Partitioning Criteria (2)

How do we find the right balance?

Consider why we form classes,

i.e., what do we gain by assigning two samples to the same class?

One important reason for grouping individuals into classes is that being told the class of an item conveys a lot of information about it.

An example:

Suppose I tell you that Fido is a dog:

Immediately you are reasonably confident of the following:

Fido has four legs

Fido barks

Fido has sharp teeth

Fido probably chases cats

etc.

Class membership may contain a lot of information.

Partitioning Criteria (3)

Motivated by the above example, from a different perspective, we could define a good partition as one that

> **maximises the ability to predict unknown attribute values of an item from its class membership.**

This may be difficult to implement. We still need to consider individual criteria in certain balanced manner.

There may be no perfect solution to this balance problem. Let's see how specific clustering algorithms take these partitioning criteria into account in a balanced manner.

Approaches to Clustering

Numerous methods have been devised for clustering.

We will look at the following two techniques:

Agglomerative Hierarchical Clustering

K-Means Clustering

Other techniques, including competitive learning, such as self-organising map, are beyond the scope of CE213.

Agglomerative Hierarchical Clustering

The Basic Idea (pseudo code):

```
Assign each sample to its own cluster.
WHILE there are at least two clusters
    Find the most similar pair of clusters
    (if there are more than one pairs with the highest similarity, choose one of them randomly)
    Merge them into a new larger cluster
```

Results are usually presented as a tree called a **dendrogram** (an example in the next slide).

This approach

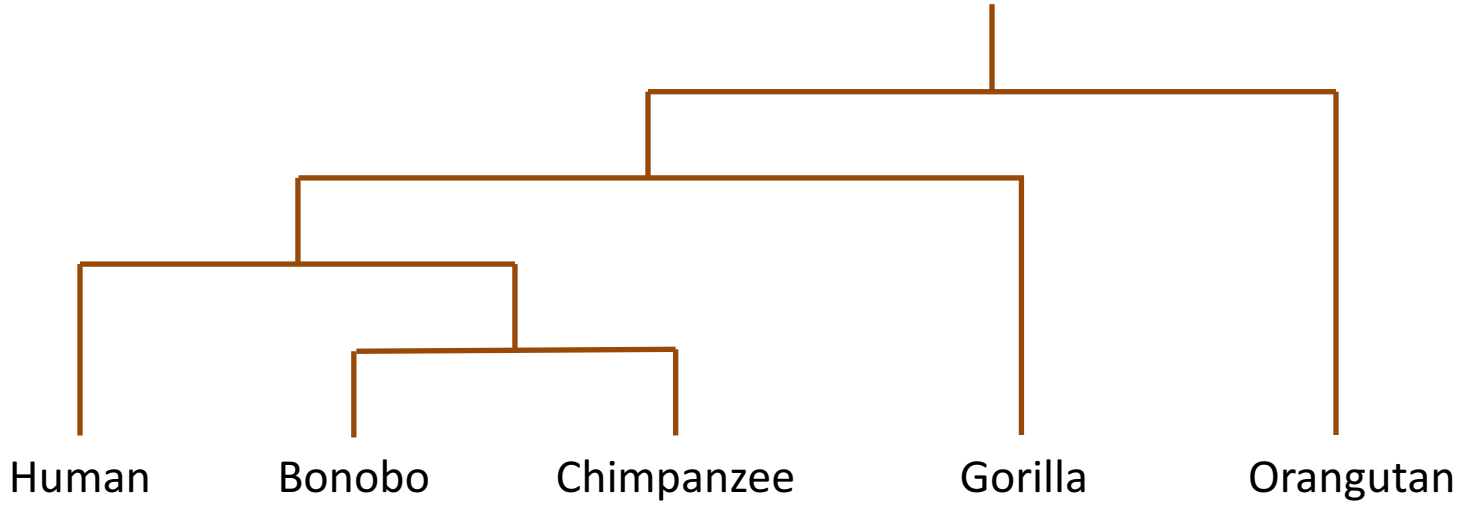
- Requires all samples to be available at the start.
- Requires a **similarity metric** that can determine the similarity between groups/clusters.
- Requires a **human analyst** to decide on the optimal number of classes/clusters (the third criterion).

[NB. Maximising similarity could be implemented by minimising distance.]

Agglomerative Hierarchical Clustering (2)

An example dendrogram:

Here is a dendrogram for clustering great apes using similarity between given DNAs of the apes.



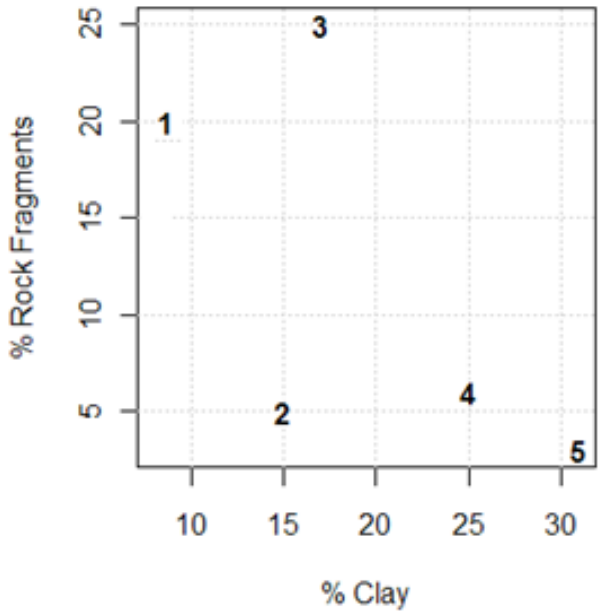
In your opinion (if you are the human analyst), what should be the optimal number of clusters/classes: 1, 2, 3, 4, or 5?

Agglomerative Hierarchical Clustering (3)

Another example : using similarity/distance between soils' attributes

Five soil samples

Soil	%Rock Fragments	%Clay
S1	20	5
S2	5	15
S3	25	17
S4	6	25
S5	3	31

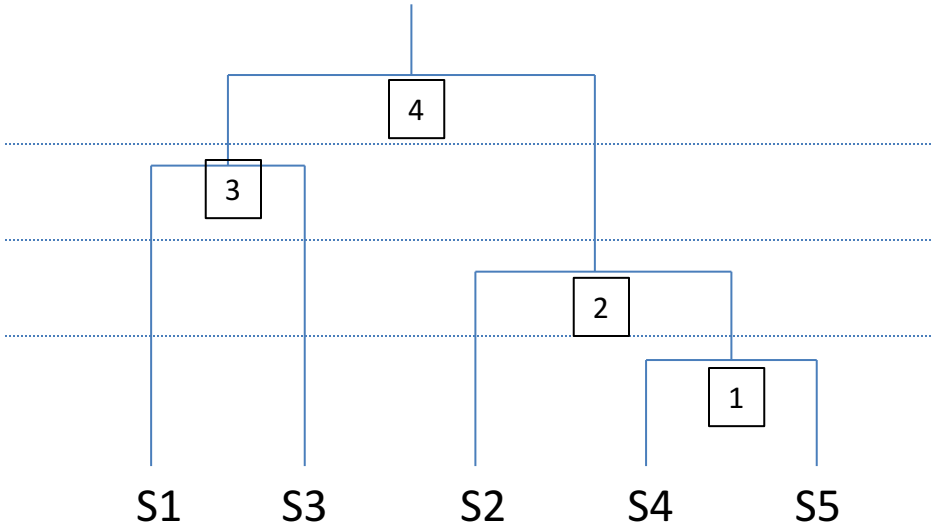


The **similarity of two soils** is defined as 1/(Euclidean distance between two attribute vectors), and the **similarity of two clusters or groups** of soils is defined as the similarity of the most similar pairs of soils where each member of the pair is from a different group.

Agglomerative Hierarchical Clustering (4)

Soil Pair	Distance
S1, S2	18.03
S1, S3	13
S1, S4	24.41
S1, S5	31.06
S2, S3	20.10
S2, S4	10.05
S2, S5	16.12
S3, S4	20.62
S3, S5	22.36
S4, S5	6.71

(minimum distance = maximum similarity)



For 2 clusters: S1 and S3 in cluster 1,
S2, S4 and S5 in cluster 2.
For 3 clusters: S1 in cluster 1,
S3 in cluster 2,
S2, S4 and S5 in cluster 3.
For 4 classes: …

Optimal number of clusters?

K-Means Clustering Method

The Basic idea can be described by the following pseudo code:

```
METHOD K-Means(samples, K) //K is number of clusters to be formed;
Choose K samples randomly as initial cluster centroids
REPEAT
    Assign each sample to a cluster whose centroid is the
    closest to it
    Update the cluster centroid to the mean value of all
    samples currently in that cluster
UNTIL no sample changes the existing clusters
Return K cluster centroids
//one sample may be used many times, and it may be assigned to different
//clusters in different iterations.
```

Input to the program is a set of unlabelled samples and the number of clusters to be formed. Output of the program is the K cluster centroids.

Number of iterations needed will depend on how well the formed clusters are.

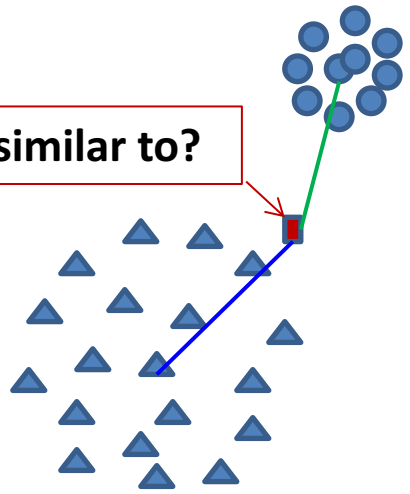
K-Means Clustering Method (2)

Demos (Pay attention to centroid initialisation and updating):

https://www.youtube.com/watch?v=BVFG7fd1H30

https://www.youtube.com/watch?v=5FmnJVv73fU

Which cluster is this new sample more similar to?



Discussions:

Cluster representation: mean, variance, ……

Similarity metric: statistical distance based, ……

If the number of clusters is unknown, how to conduct K-means clustering? – growing, pruning, separability checking, ……

Summary

Clustering

- The task of clustering
- Three partitioning criteria

Agglomerative Hierarchical Clustering

- Similarity metric
- Dendrogram
- Number of clusters

K-Means Clustering Method

- Similarity metric or distance measure
- Cluster centroids
- Number of clusters