Unsupervised Learning

Situation:
You’re given lots of data and you want to find structure in it. There’s no teacher to help you by, e.g., saying: “this piece of data belongs to that class,” or “this portion of data is especially important,” etc.

Typical assumptions:
• each datum is a vector (point) in a potentially high dimensional space (an image with 1 Mega-pixels is a point in a $10^6$-dimensional space)
• points are drawn independently from a fixed unknown underlying probability distribution (i.i.d)
• estimating the underlying distribution with non-parametric techniques is too difficult
Goals in Unsupervised Learning:

• find modes of the probability distribution, e.g. market basket analysis used in data mining
• find clusters of inputs
  • partitional clustering, e.g., k-means clustering, competitive learning
  • hierarchical clustering
• compression, e.g., vector quantization
• reduce dimensionality of the data
  • principal component analysis (PCA)
  • multi-dimensional scaling (MDS)
• fit lower-dimensional manifolds through the data
  • self-organizing map (SOM)
  • ISOMAP
• find “interesting” directions through the data: projection pursuit, independent component analysis (ICA)
Main Entry: ¹cluster
Pronunciation: 'kl&-s-t&r
Function: noun
Etymology: Middle English, from Old English cl TYSTER; akin to Old English clott clot:
- a number of similar individuals that occur together: as a : two or more consecutive consonants or vowels in a segment of speech b : a group of buildings and especially houses built close together on a sizable tract in order to preserve open spaces larger than the individual yard for common recreation c : an aggregation of stars or galaxies that appear close together in the sky and are gravitationally associated
- clus·ter·y /-t(-&-)rE/ adjective

Main Entry: ²cluster
Function: verb
Inflected Form(s): clus·tered; clus·ter·ing /-t(-&-)r[ng]/
transitive senses
1 : to collect into a cluster <cluster the tents together>
2 : to furnish with clusters
intransitive senses : to grow, assemble, or occur in a cluster
Two questions:
1. Why and how do clusters form?
2. How can you identify clusters?

Formation of clusters:
- nuclei, atoms, molecules, stars, galaxies
- biological organisms: flocks of birds, schools of fish, herds of land animals

Common mechanisms:
combinations of attractive and repulsive interactions between individual elements
Craig Reynolds: a few *local* interaction rules are sufficient for flocking. Nice and simple example of *emergence*.

Each individual follows these rules:

1. fly towards the centre of mass of neighbors
2. keep small distance away from other objects (including other boids).
3. match velocity with near boids.
Finding Clusters

Example: neural recording. Every millisecond you observe a 100-dim. binary vector (100 neurons, spiking or non-spiking). The chance to see exactly the same vector twice is negligible.

Question: are there sets of similar vectors that seem to occur frequently and that are distant from other sets?
Example: grocery store checkout: assume there are $N$ items that the store sells. A single transaction is represented as an $N$-component binary vector.

For example: a purchase where the customer bought bread, peanut butter, and jelly but nothing else may be represented by:

$$x=(0001000100000000000000100000000000\ldots)^T$$

There are $2^N$ possible states for $x$, whose probability $p(x)$ needs to be estimated. For large $N$ this is infeasible. The amount of necessary data is astronomical. (“Curse of dimensionality”)
An objective function for clustering

Consider you have somehow partitioned all the \( N \) data points into \( k \) subsets (clusters) \( D_i, i=1,\ldots,k \). How can you assess the quality of this partition?

A natural idea is the following:
1. calculate the centers (of mass) \( c_i \) of the clusters
2. consider the squared distance of the data points \( x \) to their cluster centers \( c_i \)
3. the sum of all these distances should be as small as possible

\[
J_e = \sum_{i=1}^{k} \sum_{x \in D_i} \| x - c_i \|^2
\]

Global optimization of this function is usually too hard, since there are \( k^N \) possible assignments to the clusters.
**k-means clustering**

- **iterative** algorithm to find clusters in data
- **k**: number of clusters
- **$x_1, \ldots, x_n$**: data points
- **$c_1, \ldots, c_k$**: estimated cluster centers

begin **initialize** $n$, $k$, $c_1$, ..., $c_k$

do
    **classify** all $x_i$ to closest cluster center
    **recompute** $c_i$ by calculating mean of data points “belonging” to $c_i$
until no change in the $c_i$
return $c_1$, ..., $c_k$
end

$$J_e = \sum_{i=1}^{k} \sum_{x \in D_i} \|x - c_i\|^2$$
FIGURE 10.3. Trajectories for the means of the $k$-means clustering procedure applied to two-dimensional data. The final Voronoi tessellation (for classification) is also shown—the means correspond to the “centers” of the Voronoi cells. In this case, convergence is obtained in three iterations. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
Notes:

- the k-means algorithm is maybe the simplest and most-widely used partitional clustering algorithm
- it requires to specify k in advance
- each step can only reduce the objective function
- can get stuck in local minima
- it expects clusters to be spherical
- many extensions: soft-k-means, mixture of Gaussians, general expectation maximization (EM) algorithm, ...
What are desirable properties of a clustering function that don’t imply a specific objective function?

- **Scale-Invariance**: if you measure distances between points in different units, clustering should stay the same.
- **Richness**: every partition of the points should be a possible output, i.e., you can always find distances between the points such that any partition is the clustering result.
- **Consistency**: if distances of points within a cluster are made smaller and distances between points from different clusters are made bigger, then the clustering result should remain the same.

**Theorem**: J. Kleinberg (2002):
There is no clustering function that simultaneously satisfies Scale-Invariance, Richness, and Consistency.
hierarchical clustering

- k-means gives “flat” description (partitioning). Now: clusters with sub-clusters with sub-sub-clusters ...
- agglomerative vs. divisive
- different distance functions (e.g., min, max, avg., mean)

Agglomerative hierarchical clustering:

begin initialize k, k*:={n}, D_{i}:={x_{i}}, i=1,...,n
   do
      k*:={k*-1
      find nearest or most similar clusters, say, D_{i} and D_{j}
      merge D_{i} and D_{j}
   until k*=k
   return k clusters
end
FIGURE 10.11. A dendrogram can represent the results of hierarchical clustering algorithms. The vertical axis shows a generalized measure of similarity among clusters. Here, at level 1 all eight points lie in singleton clusters; each point in a cluster is highly similar to itself, of course. Points $x_6$ and $x_7$ happen to be the most similar, and are merged at level 2, and so forth. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
FIGURE 10.12. A set or Venn diagram representation of two-dimensional data (which was used in the dendrogram of Fig. 10.11) reveals the hierarchical structure but not the quantitative distances between clusters. The levels are numbered by $k$, in red. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
Competitive Learning (Online)

\[ h_i = \sum_j w_{ij} x_j = w_i^T x \]

Simple linear units, input is \( x \)

Assume weights \( > 0 \) and possibly weights normalized. If weights normalized, \( h_i \) maximal if \( w_i \) has smallest Euclidean distance to \( x \). Define winner: \( i^* = \arg \max_i h_i \)

Output of unit: \( o_i = \begin{cases} 1: i = i^* \\ 0: i \neq i^* \end{cases} \) “Ultra-sparse code”

Winner-Take-All Mechanism:
may be implemented through lateral inhibition network
FIGURE 10.15. The two-layer network that implements the competitive learning algorithm consists of \( d + 1 \) input units and \( c \) output or cluster units. Each augmented input pattern is normalized to unit length (i.e., \( \|\mathbf{x}\| = 1 \)), as is the set of weights at each cluster unit. When a pattern is presented, each of the cluster units computes its net activation \( \text{net}_j = \mathbf{w}_j^T \mathbf{x} \); only the weights at the most active cluster unit are modified. (The suppression of activity in all but the most active cluster units can be implemented by competition among these units, as indicated by the red arrows.) The weights of the most active unit are then modified to be more similar to the pattern presented. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
\[ \Delta w_{ij} = \eta O_i (x_j^\mu - w_{ij}) \] 

learning rule is Hebb-like plus decay term

Weight of winning unit is moved towards current input \( x^\mu \).

Geometric interpretation: units (their weights) move in input space. Winning unit moves towards current input.

Can be viewed as on-line version of k-means algorithm.
FIGURE 10.16. All of the two-dimensional patterns have been augmented and normalized and hence lie on a two-dimensional sphere in three dimensions. Likewise, the weights of the three cluster centers have been normalized. The red curves show the trajectory of the weight vectors, which start at the red points and end at the center of a cluster. From: Richard O. Duda, Peter E. Hart, and David G. Stork, Pattern Classification. Copyright © 2001 by John Wiley & Sons, Inc.
Vector Quantization

Idea: represent input by weight vector of the winner
(can be used for data compression: just store/transmit label representing the winner)

Question: what is set of inputs that “belong” to a unit $i$, i.e. for which unit $i$ is the winner?

Answer: Voroni tesselation (note: matlab command for this)

Figure taken from Hertz et.al.