Lecture 8 - Discovering Groups

Summary

Being able to meaningfully cluster data into groups using **Clustering** or **Cluster Analysis** is a key part of the process of exploratory and descriptive data mining. Clustering techniques are all a form of unsupervised machine learning. Numerous techniques for clustering exist; in this lecture we'll look at two of the most common and useful: **Hierarchical Clustering** and **K-Means Clustering**. We'll also look briefly at a more advanced, but computationally intensive algorithm called **Mean-shift Clustering** that both produces clusters and finds the *modes* of the data.

Being able to group data into clusters is a good basis for understanding that data, however, in many cases these clusters can be difficult to interpret. An alternative approach is to attempt to produce 2D visualisations (images) that highlight the key relationships in the data by projecting the data from a high dimensionality to two dimensions. We'll look at three key algorithms: **Principal Component Analysis (PCA)**, **Multidimensional Scaling (MDS)** and **Self Organising Maps (SOMs)**

Key points

- Clustering is an unsupervised machine learning technique, that learns to group data without prior knowledge of what the groups should look like.
    - Hierarchical Clustering attempts to iteratively break data into a hierarchy of clusters
        - Hierarchical Agglomerative Clustering builds a binary tree of clusters from the leaf nodes upwards towards the root
            - Known as a bottom-up approach
            - Requires three things:
                - a set of items to cluster
                - a distance measure to measure how close items are to each other
                    - e.g. an Lp distance, a similarity measure converted to a distance (i.e. 1-Pearson or 1-cosine)
                    - Doesn't necessarily have to be a distance computed over a vector; some forms of agglomerative clustering allow only need a matrix of distances or similarities computed between all items as input (see below)
                - a **linkage criterion** which measures dissimilarity of clusters as a function of the pairwise distances of items in the clusters
            - Basic approach:

                - Initially every item is in a cluster of its own
                - While there is more than one single cluster:
                    - The closest pair of clusters according to the linkage criterion are merged into a bigger cluster

            - By recording the

        - Two categories of linkage criterion:
            - Centroid-based linkage functions that measure similarity between clusters based on the distance between their centroids
                - Requires that each item is represented by a numeric feature vector that can be interpreted as a position in space
                - Examples:
                    - **Weighted Centroid Clustering** *(WPGMC - Weighted Pair Group Method with Centroids; often also known as the "median" method)*
                    - **Unweighted Centroid Clustering** *(UPGMC - Unweighted Pair Group Method with Centroids)*
            - Distance-based linkage functions that measure distances between clusters as a function of the distances between items within those clusters.
                - Clustering can be performed purely as a function of a **distance matrix** in which each element $\mathbf{D}_{i,j}$ represents the distance between items i and j
                - Examples:
                    - **Maximum** or **complete-linkage clustering**:
                    - **Minimum** or **single-linkage clustering**:
                    - **Mean** or **average linkage clustering** (UPGMA - Unweighted Pairwise Group Method with Arithmetic Mean):
                    - **Minimum energy clustering**:
        - In general, complexity is $O(n^3)$, which can be a problem for large data sets, however there are some $O(n^2)$ variants for the single-linkage and complete-linkage cases
    - Divisive clustering algorithms ("top-down" approaches), which start with all the data in the root node and recursively split do exist
        - Not widely used in practice.
            - One major reason is that in general complexity is $O(2^n)$, which is worse than the agglomerative methods.
- The K-Means algorithm is a simple, but powerful, approach to clustering that attempts to group data in a feature space into K groups or clusters represented by centroids (i.e. the mean point of the class in feature-space).

> * The K-value must be set a priori (beforehand)
> * To begin, K initial cluster centres are chosen (typically randomly or from a sample of the existing data points, although note that better initialisation procedures exist - e.g. the KMeans++ algorithm)
> * Then the following process is performed iteratively until the centroids don't move between iterations (or the maximum number of iterations is reached):
> * Each point is assigned to its closest centroid
> * The centroid is recomputed as the mean of all the points assigned to it. If the centroid has no point s assigned it is randomly re-initialised to a **new** point.
> * The **final** clusters are created by assigning all points to their nearest centroid.

    - K-Means always converges, but not necessarily to the most optimal solution
- Mean-shift is a standard algorithm to efficiently find the modes of a **Probability Density Function (PDF)** from a set of samples of that PDF (i.e. the featurevectors representing a set

of items).

- The only variable of the mean-shift algorithm is the **kernel bandwidth** of a **kernel density estimator**.
    - This means it automatically chooses the number of clusters!
- The probability density function (PDF) of a continuous random variable is a function that describes the relative likelihood for this random variable to take on a given value
    - The PDF is non-negative everywhere and sums to 1
- In the context of a feature space, the PDF is a function that tells you how likely it is that a featurevector is *drawn* from a specific location in a feature space.
    - A feature vector drawn from part of the space where there are lots of similar items would have a higher probability density than if the drawn feature vector were from a part of the space with very few similar items
        - or in other words, dense parts of the space with more items have a higher probability density
    - Generally speaking, for arbitrary features describing a set of items the PDF cannot be described empirically
        - Must be estimated using some other method
            - Simple, but crude, way to do this would be to quantise the feature space into bins in order to build a histogram
                - Each bin would contain the count of the number of items with feature vectors falling into that bin divided by the number of total items
                - Major disadvantage of this approach is that it isn't *continuous* and only gives a discrete approximation of the PDF
            - Better way to do this is to use a **Kernel Density Estimator** (also known as a **"Parzen Window"**)
                -

##Further Reading

- k-means++: the advantages of careful seeding (http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf) . Arthur and Vassilvitskii. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035. 2007.