Clustering

October 16, 2012

1 Basics

1.1 The Idea of Clustering

- Clustering is *generalizing* over a similarity measure.
- Elements have a similarity between one another and between itself and not-elemental objects like for example the hypothetical average element (*centroid* in a vector space respresentation).
- The goal of clustering is to maximize *intra-cluster similarity* and minimize *inter-cluster similarity*, that is we are looking for a partition that best groups similar elements and separates different elements.

1.2 The Goal of Clustering

The best partition of the set to be clustered is the one with the best inter-intra-similarity ratio: we maximize $p_1 = \frac{intra-sim}{inter-sim}$ or minimize $p_2 = \frac{inter-sim}{intra-sim}$, where p_1, p_2 are partition quality measures.

1.2.1 Overall Intra-Cluster Similarity

But how do we measure inter- and intra-similarity? Getting the intra-cluster similartiy is rather easy. If we use an *element* × *element* matrix for the new cluster and enter the similarity for each $\langle element, element\rangle$ pair, we can sum over all values in the matrix, devide it by the number of edges and get the overall similarity:

```
intrasim := 0;
numberofedges := 0;
for each element e1:
        for each element e2:
                if e1 != e2:
                        intrasim += sim(e1,e2);
                        numberofedges++;
                endif
        endfor
endfor
return intrasim/numberofedges;
```

1.2.2 Overall Inter-Cluster Similarity

Measuring inter-cluster similarity is more complicated. We can for example generate a hypothetical average element (*centroid* / *medoid*) representing each new cluster and measure the similarity between the new clusters' representatives. We can also use the similarity between the most similar (*single-link* similarity function) or dissimilar (*complete-link* similarity function) $\langle c_1, c_2 \rangle$ pair, where c_1 is from one cluster C_1 and c_2 from the other (C_2). Instead of thinking of the inter-cluster similarity we can also measure the overall similarity in the set of all new clusters. We want to get clusters that have a high similarity in comparison to the overall similarity in the preceding separation step: we maximize $p_3 = \frac{sim(A)+sim(B)}{sim(A \cup B)}$. This is of course best when each element has its own cluster. Therefore we need to either set a constant for the number of clusters generated in each separation step or introduce a cost-factor that increases with the number of clusters (see *Minimal Description Length*).

Another way of describing the goal of clustering is to maximize the mutual information.

1.3 Similarity vs. Difference

There are several standart similarity or distance measures like for example the *euclidean distance* in a vector space representation or the *jaccard coefficient* (and many others). In principle they are applied on AVM's representing the elements to be clustered. The crucial part when it comes to applying clustering to a specific set of entities is thus to decide on which features to use as attributes and how to derive a value for this attribute from the entity. It is possible to learn this from a training set of already clustered elements (requires a lot of manual work).

The relation between similarity and difference is of course opposed. We can turn any similarity measure in a distance measure: $\frac{1}{1+sim}$ and the other way round: $\frac{1}{1+dist}$. As similarity increases, distance decreases. As distance increases, similarity decreases. We say that the similarity between two identical elements is maximal and 1. The minimum similarity is 0.

1.4 Soft- vs. Hard Clustering

In soft clustering, an element can belong to more than one cluster. It is even possible to assign a degree of belonging to each $\langle element, cluster \rangle$ pair. None of this is allowed in hard clustering. So we get that in soft clustering we have a $element \times cluster$ matrix that is filled with the respective value for the degree of belonging. In soft clustering without degrees we can use the same matrix

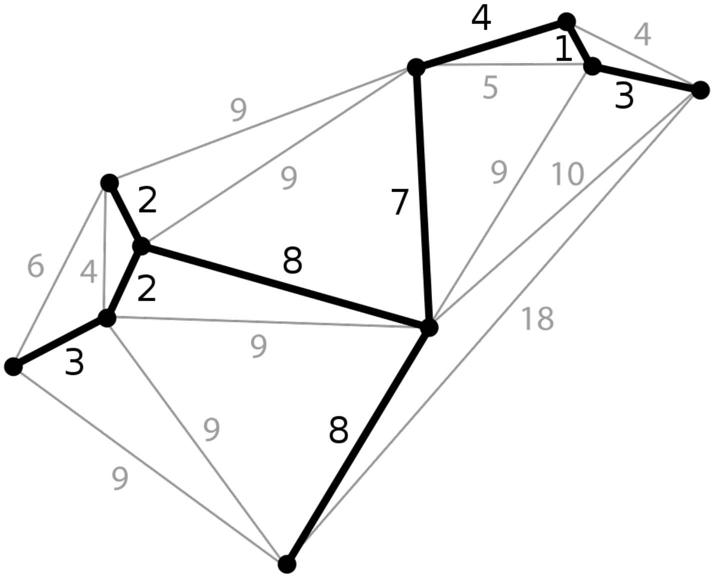

Figure 1: A *minimum spanning tree* (*MST*)

only assigning 0 and 1's and in hard clustering we allow only 0 and 1's and in addition only one 1 per element. In the latter case it is more compact to think of a *element* × *cluster* function.

1.5 Coherence and the Minimal Spanning Tree

On each set to be clustered can be projected a *Minimal Spanning Tree* (*MST*) that treats all elements in the set as nodes and has the smallest overall edge length of all possible trees combining these nodes. Using the single-link measure (see below) we can define the coherence of a cluster as the smallest similarity between two nodes in the MST. Using the complete-link or group-average measure, we cannot use the MST. In the first case we take the smallest similarity of all $\langle element, element \rangle$ pairs in the cluster as the coherence and in the latter we use the average similarity of all the pait-similarities. The coherence of a cluster reflects the case that the most distinct element in the cluster would be a separate cluster and the inter-cluster similarity of the two clusters would be computed.

1.6 Ending Conditions of Clustering Algorithms

Hierarchical Clustering needs no ending condition since it terminates automatically. However, for flat clustering we need to determine a maximum number of clusters to keep it from splitting into separate clusters for each single element.

Another possibility is to use *Minimal Description Length* (*MST*) or terminate when the clustering does not change anymore or the overall clustering quality begins to decrease.

1.7 Reallocations

Some clustering algorithms perform reallocations during runtime. That means that a cluster is not clearly assigned to one cluster after some iteration because it might be reassigned to another cluster lateron.

1.8 Medoid vs. Centroid

The notion of the centroid in vector space is a imaginary element that is not in the set of elements to be clusters but projected into the vector space by taking the average of all values for all attributes. The medoid is the element in the set that is closest to the centroid. Some algorithms use the medoid instead of the centroid as the center of a cluster.

1.9 Clustering vs. Classification

While clustering uses a similarity measure to compare elements with elements and derive a structural ordering by itself (unsupervised learning), classification uses a similarity measure to compare elements with already existing patterns defined in advance for specific groups (supervised learning).

2 Hierarchical Clustering

We can think of the hierarchical clustering process in two ways: iteratively separating clusters top-down starting with an initial Über-Cluster (*Divisive Clustering*) or iteratively grouping bottom-up from initial 1-elemental clusters (*Agglomerative Clustering*).

2.1 Top-down Hierarchical Clustering

A cluster is devided into sub-clusters such that the similarity between the evolving clusters is minimized and the similarity between the elements within each of the clusters is maximized. One can think of a binary tree as the simplest form of an hierarchical clustering result. In each separation step the current cluster is devided into exactly 2 new subclusters. Note that if we cannot find an optimal partition, we must at least make sure that the partition does not reduce the partition quality (Monotonicity) because otherways the hierarchy is invalid. We know that unless we are already on the bottom level, there is always a partition that satisfies the monotonicity criterion, that is $p \geq 1$.

2.2 Bottom-up Hierarchical Clustering

In Agglomerative Clustering we start with seed clusters. We can use all elements as seeds or have an initial influence on the process by selecting especially distinct seed clusters. In each agglomeration step we add one external element or cluster to each cluster such that the similarity between each of the combined pairs is maximal. Obviously, an element can only be paired with one cluster and a 1-elemental cluster that is paired with another cluster will not pair with another element or cluster anymore. We say two clusters were *merged*. If a cluster c is to be merged with another complex cluster (and not just an element or a 1-elemental cluster) we have to choose the cluster that will lead to the greatest intra-cluster similarity after merging. So of all the possible $\langle c, c' \rangle$ pairs, we are looking for the one that maximizes $sim(c \cup c')$. In this case it might also make sense to use a notion of weight for clusters such that heavy clusters (with many elements) can incorporate lighter surrounding clusters. On might imagine a "universe" of clusters with clusters having their own gravity.

3 Flat Clustering

A *Flat Clustering* does not result in a hierarchy. So the set of elements to be clustered is just devided in a number of clusters on top level. The most popular algorithms for flat clustering are *k-means* and *Expectation Maximization* (*EM*). K-means depends heavily on the notion of the centroid or medoid while the EM algorithm uses statistics to calculate the cluster model that maximizes the likelihood of the data given the model.

3.1 K-Means

Using k-means cluster we start with k seed "clusterpoints" that can be set randomly or automatically or manualy. The results will vary depending on where the initial "clusterpoints" were placed (note that in the following for clarity we distinguish between a cluster as a set of elements and a "clusterpoint" that is treated as the center of the cluster). After that the following steps are repeated until an ending condition is reached:

1. Assign each element to the closest clusterpoint
2. Move the clusterpoint into the actual center of the cluster

3.2 EM-Algorithm

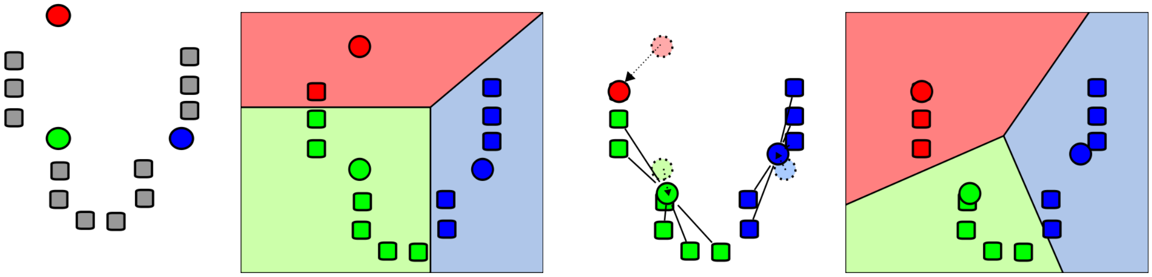

Figure 2: An illustration of the k-means algorithm