Data Mining

Impossibility theorem for clustering*

Mauro Sozio

***advanced topic, not covered at the exam**

Impossibility theorem for clustering

- A clustering function takes a distance function d and a set of points S (|S| >=2) and returns a clustering (partition) of S.
- A distance function is a function S x S -> R, s.t., 1) d(i,j)>=0, d(i,j)=0 iff i=j, d(i,j)=d(j,i). All results hold with or without triangle inequality.
- We will list three desirable properties that no clustering algorithm can have and show that there are algorithms satisfying any 2 of them.

Property 1: Scale-Invariance

- Scale-invariance: for any distance function d and any alpha > 0, f(d,S) = f(d x alpha,S) for any S.

- This simply implies that the clustering function is not sensitive to changes in the units of distance measurement.

Property 2: Richness

- The clustering function f should be able to produce any possible clustering of S.

- In other words, suppose we are given only the "names" of the points in S and not their distances. Then for any partition C of S we should be able to define a distance function d such that f(d,S)=C.

Property 3: Consistency

- Let d and d' be two distance functions. Let f(d)=C and let d' have the following two properties: 1) if points i,j belong to a same cluster in C then d'(i,j)<= d(i,j); 2) if i,j belong to two different clusters in C then d'(i,j)>= d(i,j). Then f(d')=C.

- That is, if we decrease the distances between points in a same cluster and increase the distances between points in different clusters we should still get the same clustering.

Impossibility Theorem for Clustering

- **Theorem:** There is no clustering function f that satisfies Scale-Invariance, Richness, and Consistency.

- We now show that there are algorithms that satisfy and two of them.

Single-linkage (aka agglomerative clustering)

- Let G=(S,E,d) be a complete graph where nodes are elements in S and edges (i,j) are associated with the distance d(i,j).

- Let e_1,...,e_k be the edges in G sorted non-decreasingly according to their weights, i.e. d(e_1) <= d(e_2) <= ... <= d(e_k).
- H=(S,\emptyset)
- For i=1,...,k
  - add e_i to H
  - if some stopping condition is verified stop.
- Let the connected components in H be the clustering of S.

Stopping conditions

- By carefully defining the stopping condition, we can satisfy any 2 of the 3 properties.

- Stopping conditions:
  - **k-cluster stopping condition**. Stop as soon as H contains k connected components.
  - **distance-r stopping condition**. Add all and only the edges of weight at most r.
  - **scale-alpha stopping condition.** let d_max be the max. distance between any points. Add all and only the edges with weight at most alpha*d_max.

Observations

- The k-cluster stopping condition violates *richness*
- Distance-r violates *scale-invariance*
- Scale-alpha violates *consistency*

Theorem

- For any k>=1, n>=k single-linkage with the k-cluster stopping condition satisfies SI and Cons.
- For any 0< alpha <1, n>=3, single linkage with the scale-alpha condition satisfies SI and Rich.
- For any r>0, n>=2 single linkage with the distance-r condition satisfies Rich and Cons.

K-means: which properties?

- Which of the previous properties are satisfied by the k-means algorithm?

  – scale invariance? **Yes** (provided we choose the same centroids).
  – richness? **No** (k-means produces at most k-clusters not any possible partition).
  – consistency? **No** see [1] for a proof.

Reference: [1] An Impossibility Theorem for Clustering, J. Kleinberg, NIPS 2002. (https://www.cs.cornell.edu/home/kleinber/nips15.pdf)