A quantitative approach to Consistency Theorem in Clustering

G14PJA

Mathematics 3rd Year Project

Autumn 2018/19

School of Mathematical Sciences

University of Nottingham

Zehui Li

Supervisor: Dr. Yves van Gennip

Project code: XX P99

Assessment type: Review

I have read and understood the School and University guidelines on plagiarism. I confirm that this work is my own, apart from the acknowledged references.

Abstract

Contents

1 Introduction

Clustering analysis can be defined as a process of segmenting the data points into several subsets, or clusters, with the goal of making the data points within a cluster to be similar to each other, while the data points in distinct clusters to be different. Clustering has been widely used in many fields, such as pattern recognition, bio-informatics and image processing, however, most of the study toward the uniform notion of clustering only stop at the very general level. The algorithms to achieve the clustering task are called clustering algorithm: depending on the definition of the clusters and the way to find the clusters, these clustering algorithms differ from each other significantly. In 2003, Kleinberg [1] published a highly influential paper, in which he set up a general framework to study clustering algorithms as a whole, and proposed three properties that any clustering algorithms could have.

Clustering algorithms fall into three categories [2]: combinatorial algorithms, mixture modelling, and model seeking, the algorithms in each category follows different underlying principal. The advantage of Kleinberg's framework is that it can be applied to all these clustering algorithms regardless of these principal. The core of the framework - three properties proposed by Kleinberg - are called scale invariance, richness and consistency respectively. Scale invariance states that if the distance(dissimilarity) between the data points is multiplied by a positive number, the clustering algorithm should partition the data into the same clusters as before. Richness requires that for any given partition of the data points, it will be possible to come up with a pair wise distance between the data points, so that the clustering algorithm can produce the given partition. Finally, a clustering algorithm satisfy the consistency theorem if we decrease distances between the data points within a cluster, increase the distance between the cluster, the algorithm should produce the same partition. The most important conclusion from Kleinberg's paper is that there is no clustering algorithm which could satisfy three properties at the same time.

Following the impossibility theorem proposed by Kleinberg, numerous relaxation methods on the axiomatic system are proposed in recent years. In particular, relaxation of

consistency theorem is the focus of this paper. The consistency theorem proposed by Kleinberg, while reasonable and simple, it give a relative strict restriction on clustering algorithms - it require the clustering algorithm to give the same partition results even when the data set is perturbed profoundly (we will explain why this theorem is not sensible in more details in the following section). In this paper, we start from reviewing the work of Kleinberg, adding the missing proof to statements made in his paper, and do simulations on the computer to show the validity of these statements. The rest of the paper will focus on the study of consistency theorem - we come up with a quantitative framework to investigate the consistency property of clustering algorithm. and identify the highly separable distribution of partition results given legitimate perturbation. Finally, we tried several machine learning method to capture the relation between the extend of perturbation and change of partition results. It turns out the methods can capture the model with a very high accuracy and f-measure. At the end of the paper, we try to construct a quantity to measure the extend of perturbation and explore correlation between the the measurement and the change of partition.

1.1 Related work

1.2 Our contribution

2 Review of Kleiberg's work

This section begins by introducing the mathematical notations used for clustering, and give the formal definition of scale invariance, richness and consistency theorem. After having these definition/knowledge in mind, we will move on to prove three statements about a very simple clustering algorithm - single linkage. Then we will present a process of showing the scale invariance property using computer simulation(in python). Finally, we will discuss why consistency theorem is a more strict restriction compared to the others, Which gives us motivation to explore the ways to change consistency theorem.

2.1 Priliminaries

Every clustering algorithm can be denoted by a "clustering function" f, the input of this function is a set S consisting of n data points and the pairwise distances among them. Each points in set S is represented by a integer, so $S = \{1, 2, 3, ..., n\}$ with $n >= 2$. One the other hand,there are multiple ways to represent the pairwise distances, for example, for $S = \{1, 2, 3, ..., N\}$, a $N \times N$ distance matrix M can be used to represent the distances, in which each entry $M_{i,j}$ refer to the distance between points i and j. However, in our case, instead of using the distance matrix, we will use *distance function* to denote the pairwise distances, which is more convenient when dealing with the theorems we defined below. *Distance function* is define as a function d: $S \times S \to \mathbb{R}_{\geq 0}$ with symmetric property, thus, $d(i, j)$ equals $d(j, i)$,and both represent the distance between the points $i, j \in S$. In particular, $d(i, i) = 0$ for any $i \in S$. Distance function are not required to be *metrics*, in other words, d don't need to satisfy triangle inequality, but adding such restriction will not affect results we have below.

Naturally, *clustering function* f take a data Set S and a distance function d as the inputs, and output the a partition Γ of S, where $\Gamma = \{C_1, C_2, ..., C_k\}$, each of the cluster C_k contains some of the data points in S. For example, let $S = \{1, 2, 3, 4, 5\}$, then we could have $f(S, d) = \Gamma = \{\{1, 2\}, \{3, 4, 5\}\}$. For simplicity, we can also write $f(S, d)$ as $f(d)$ without explicate referring to data set S. These three properties - scale invariance, richness and consistency - are all defined around this clustering function f.

Definition 2.1. *Scale-Invariance. f satisfy Scale-Invariance $\iff$ For any given distance function d and any $\alpha > 0$, $f(d) = f(\alpha \cdot d)$*

Scale Invariance simply requires that the clustering algorithm don't rely on the fixed quantity to cluster the data set.

Definition 2.2. *Richness. f satisfy Richness $\iff$ For any given partition Γ of S, $\exists d$ such that $f(d) = \Gamma$*

This property is called richness, because the by feeding in the clustering algorithm different distance functions d, we can reach any possible partition of the given data set S. The third property is called consistency, and it contains more details than the first

two. The basic idea of consistency is that, if we perturb the data in a desirable way, our algorithm should produce the same partition Γ. We first give a formal definition to the "desirable perturbation", calling it Γ *-transformation.*

Definition 2.3. *Given a partition* $\Gamma = \{C_1, C_2, ..., C_m\}$ *on data set* S, d' *is a* Γ *-transformation of* $d \iff$ *For any points* $i, j \in C_k$, $d'(i,j) <= d(i,j)$*; and if* $i \in C_k, j \notin C_k$, $d'(i,j) >= d(i,j)$.

It may seems a little messy at the first glance, but we can interpret Γ *-transformation* as a specific way to perturb the dataset. Suppose we have a data set at the beginning, then we apply a clustering algorithm on this data set, and obtain several clusters. We will squash the points within the same cluster together, and move the points in one cluster away from the other clusters. The resulting distance, will be d' in our definition above. And consistency property simply require that, if we apply the clustering algorithm on the perturbed version of data set, we will still have the same points assigned to the same clusters.

Definition 2.4. *Consistency.* f *satisfy consistency* $\iff$ *Given that* d' *is* Γ*-transformation of distance function* d, $f(d) = f(d')$

Kleinberg's framework starts from abstraction: it abstracts clustering algorithm into a clustering function f, then defines several properties to analyse the function. Once we have this framework, there are two direction to continue the study: the first is to come down to specific algorithms and study whether or not this clustering algorithm satisfy these properties; the other way is to modify the existing properties or add new properties into this framework. Both approaches are mentioned in this paper: We study **single linkage** clustering algorithm in the next section, then the rest of the paper will seek ways to modify the **consistency** theorem.

2.2 Missing Proofs for single linkage

Single linkage is a bottom-up hierarchical clustering algorithm, it begin with each clusters representing a single group, then at each step, it will merge the two "nearest" (with least

dissimilar) clusters into a single cluster, where dissimilar is defined in following manner [2]. The algorithm will terminate until some termination condition is satisfied.

Definition 2.5. *Let G,H represent two clusters, d is the distance function of the data set, then dissimilar d_{SL} is defined as:* $d_{SL}(G, H) = \min_{i \in G, i' \in H} d_{ii'}$

An alternative way to describe single linkage is to treat clustering as a Graphs Construction process [3]. Tuple (S, d) naturally form a complete Graph $G(S, d)$, whose node set is the data set S, and weigh of edges between nodes $i, j \in S$ is the distance function d_{ij}.Single linkage will first order the edges in the non-decreasing order, then for each iteration, it will take one edge from the ordered list,then terminate when the termination condition is satisfies. At this point, all the picked edges form a new partially connected graph G_c, where the node set is the still the data set, but edge set is a set formed by all picked edges. We will use this graph perspective in the following proofs.

By controlling the "termination conditions", we can construct three single linkage algorithms, such that each of the them can satisfy two properties out of Scale-invariance, Richness and Consistency. Three stop conditions are listed below[1].

- *k-cluster termination condition.* Stop adding edges when the partially connected graph G_c consists of k connected components.
- *distance-r termination condition.* Only add edges of weight at most r.
- *scale-α termination condition.* Let ρ^* denote the maximum pairwise distance; i.e $\rho^* = \max_{i,j} d(i, j)$. Only add edges of weight at most $\alpha\rho^*$.

Each of the termination condition is a trade-off between three properties: for example, single linkage with *distance-r* termination condition has a built-in scale, so it will not satisfy the scale-invariance property. But this algorithm indeed satisfy Richness and Consistency. Similarly, we have the following theorem:

Theorem 2.1. *For any $\alpha >= 1$, and any $n >= 3$, single-linkage with the scale-α termination condition satisfies Scale-Invariance and Richness.*

Theorem 2.2. *For any $k >= 1$, and any $n >= k$, single-linkage with the k-cluster termination condition satisfies Scale-Invariance and Consistency.*

Theorem 2.3. *For any $r > 0$, and any $n >= 2$, single-linkage with the distance-r termination condition satisfies Richness and Consistency.*

Here we present the proof of *Theorem 2.1* and enclosed the proof of *Theorem 2.2* and *2.3* in the appendix A

Proof. theorem 2.1

Given data set S and distance function d, let $\rho^* = \max_{i,j} d(i,j)$, and f be the single-linkage with scale-α termination condition.

Let's first prove that f satisfy Scale-invariance property. Assume another distance function d', and d' satisfy that for $\forall\ i,j \in S, d'(i,j) = \beta d(i,j)$, where $\beta > 0$. Let $\Gamma = f(S,d)$, $\Gamma' = f(S,d')$ respectively. If we can show that $\Gamma = \Gamma'$, we will prove f satisfy Scale-invariance. Following the Graph interpretation of single-linkage, the resulting partition can be represented by a partially connected Graph $G_c(S,E)$, where the node set is data set S, and edge set E consists of picked edges. Let $G_c(S,E) = \Gamma = f(S,d)$, and $G'_c(S,E') = \Gamma' = f(S,d')$. Now, if we can prove that $G_c(S,E) = G'_c(S,E')$, we are done. To prove $G'_c = G_c$, we only need to prove that edge set $E = E'$, because G and G' have the same node set.

Let's look inside E and E': Due to "scale-α" termination condition, edge Set E will contain all the edges which has weights smaller than $\alpha\rho^*$. Formally, $d(e_i) <= \alpha\rho^*$, for $e_i \in E$; $d(e_j) > \alpha\rho^*$, for $e_j \notin E$. Similarly, E' will only contain the edges which are smaller or equal to the "threshold value", let's denote this value by ρ_2^*. So for $e_i \in E'$, $d'(e_i) <= \rho_2^*$, and for $e_j \notin E'$, $d'(e_j) > \rho_2^*$. But $d'(.) = \beta d(.)$ and $\rho_2^* = \max_{i,j} d'(i,j) = \max_{i,j} \beta d(i,j) = \beta \max_{i,j} d(i,j) = \beta\rho^*$. If we substitute the values of $d'(e)$ and ρ_2^* with $\beta d(e)$ and $\beta\rho^*$ in the inequality for E', we will have $\beta d(e_i) <= \beta\alpha\rho^*$, for $e_i \in E'$, which is equivalent to the constraints for elements in E, thus E' has the same elements as E, which indicate that $\Gamma = \Gamma'$, and we are done with the proof of Scale-invariance.

The proof for richness much easier: Given a partition Γ, if we can construct a distance function d such that $f(d) = \Gamma$, we are done. Let's assume we are given a partition $\Gamma = G_c(S,E)$, where $E = \{e_1, e_2, \ldots e_m\}$. To make f produce such a edge set, we can

define d as following:

$$d(e_i) = \begin{cases} \alpha^2, & i \in \{1, 2, 3, \dots m\} \\ 1, & i \notin \{1, 2, 3, \dots m\} \end{cases}$$

In this manner, $\rho^* = \max_{i,j} d(i,j) = 1$, because *theorem 2.1* assumes $\alpha < 1$. Let $f(d) = G(S, E_{new})$, f will put all the elements which are smaller and equal to $\alpha\rho^* = \alpha$ into E. Because $d(e_i) = \alpha^2 < \alpha$, for $i = 1, 2 \dots m$, we have $E_{new} = \{e_1, e_2 \dots e_m\} = E$ □

In this proof, we show how to prove Scale-invariance and Richness. The way to prove Consistency is somewhat similar to Scale-invariance, it starts from treating the partition Γ as a partially connected Graph $G_c(S, E)$, then use the property of f and the relation between d and d' to build the equivalence of two edge set E and E'. The details are put in Appendix A.

2.3 Why modify the consistency property?

Kleiberg's **impossibility theorem** states that it is impossible for any clustering algorithm to have three properties at the same time, in other words, it indicates that we shoud not have unrealistic expectation to have a "perfect" clustering algorithm, but are these three properties desirable? Following the discussion in the blog of Willians[4], we will argue that, consistency property doesn't really reflect our expectation to clustering algorithm.

The idea of consistency theorem is that after we apply Γ-transformation to the data set, the clustering algorithm is not influenced by this perturbation, and can still produce the same partition results. As *figure 1* illustrate, this property seem very intuitive at first glance: if we increase the distance between each cluster or squeezing the points within a cluster together, it should be more obvious to the algorithm that which points should be classified into the same group.

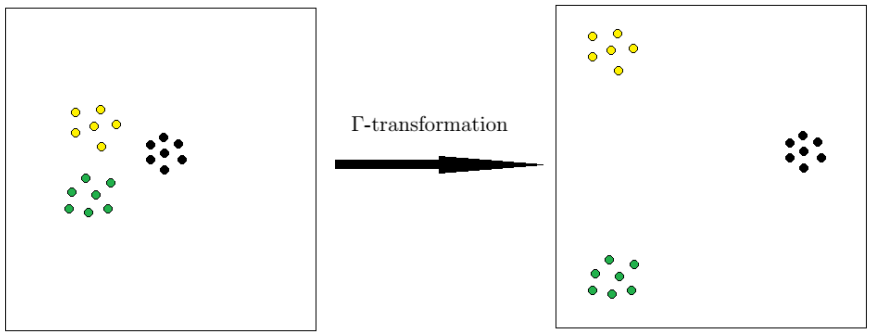

Figure 1: *This is example of* Γ*-transformation for two dimensional data: After the transformation the data points in different clusters are moved away from each other, and the boundary between "clusters" should be more clear than before.*

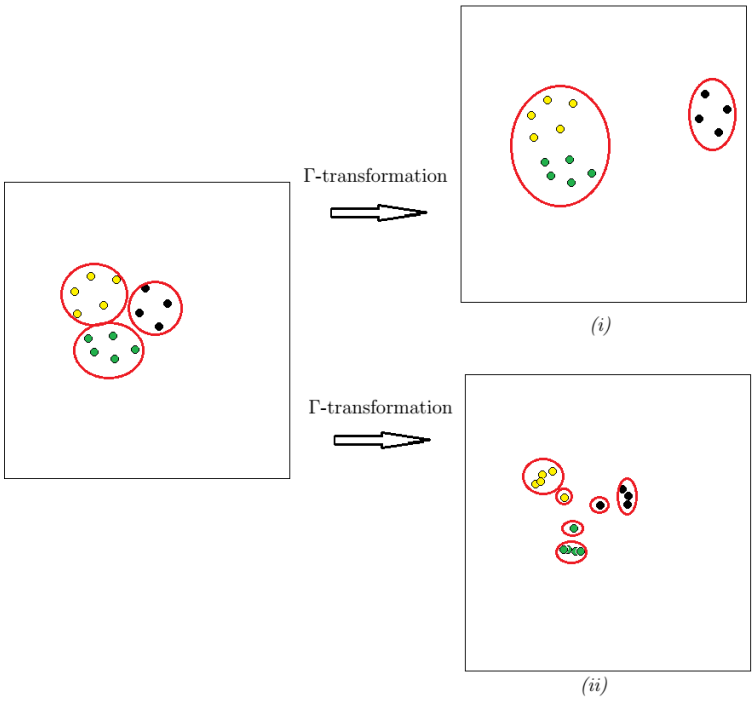

Figure 2: *This is the example of* Γ*-transformation which results in changing the structure of the data: In (i), one of the clusters is pulling away from the other two clusters significantly. In (ii), for each of the cluster, we the shrink all the points together but leave one point out.*

Despite this seemingly reasonable assumption that Γ-transformation always keep the structure of the data the same, actually, in many cases (as *Figure 2* shows), Γ-transformation

will create undesirable transformation, in which the perturbed version of data should be clustered differently. For example, in (i), suppose we move one cluster infinitely away from other clusters, obviously it is a legitimate Γ-transformation, but the resulting data should be partitioned into two clusters. However, consistency theorem, if hold, will require the clustering algorithm to produce three clusters as before. Similarly, in (ii), for each cluster, without breaking the constraints of Γ-transformation, if we shrink all the points together, but leave one points out, it should be more reasonable to partition the data in the way shown in *Figure 2*.

These problems with consistency theorem motivate us to study the property of Γ-transformation, try to figure out in which case does Γ-transformation change the structure of the data set and in which case it does.

3 Framework

It is very important to specify what we are studying here:

3.1 Generate data points by perturbation

3.2 Measurement of partition result

- Rand Index and other measurements

4 Classification with SVM and other methods

5 Conclusions

A Remaining Proof for Single-linkage

In this appendix, we provide the proof for *Theorem 2.2* and *Theorem 2.3*.

B Calculations for section ??

In this appendix we verify equation (**??**).

References

[1] Jon M Kleinberg. An impossibility theorem for clustering. In *Advances in neural information processing systems*, pages 463–470, 2003.

[2] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

[3] D Manning Christopher, Raghavan Prabhakar, and Schutza Hinrich. Introduction to information retrieval. *An Introduction To Information Retrieval*, 151(177):5, 2008.

[4] Alex Williams. Is clustering mathematically impossible, 2015.