0.1 Clustering and Impossibility

The only way of discovering the limits of the possible is to venture a little way past them into the impossible. Arthur C. Clarke

0.2 The importance of clustering

Clustering in the *analysis of data* is one of the most critical operations in many application areas, particularly in the growing fields of *machine learning* and *data science*. There are hundreds of methodological papers on proposed ways to do clustering, and tens of thousands of papers where clustering methods have been applied to different data-sets to try to extract important insights from the data.

0.2.1 Clusters and clustering

In a large set of data, S, a *cluster* is a *subset* of S where the elements in the subset are highly related to each other, and much less related to the data outside of the subset. Then *clustering* is the task of dividing the elements of S into several non-overlapping clusters. We want a clustering which gives meaningful insight into the sub-structure and possible communities in the data. Figure 1 panels a) and b) illustrate these definitions with points on the plane where, in this example, the relatedness of two points is measured by their straight-line distance from each other. Straight-line distance is usually called *Euclidian distance*.

More generally, for each pair of elements (i, j) in set S, there is a distance, denoted $d(i, j)$ between elements i and j. We call d a *distance function*. Note that in general, distances need not have any relation to a Euclidian distance.

The key questions Having seen an abstract example of clustering, the key question are: what makes a good clustering, and how do we find them? Despite many proposed clustering algorithms, and a huge number of empirical studies where clusters in data-sets have been suggested, there has been very little research on fundamental and foundational answers to these key questions. One of the few such efforts is discussed next.

In this short chapter In this chapter we will discuss an *axiomatic* approach to defining what a good cluster *algorithm* (or mechanism) should be, and then prove a theorem that shows that it is *impossible* to devise such an algorithm. This axiomatic approach, and the resulting impossibility results follow the intellectual tradition initiated by Arrow and his impossibility theorem, discussed in the previous chapter.

The specific proof we present will also show that with somewhat more *realistic* axioms, it is still impossible to develop a clustering algorithm that satisfies all of the axioms. However, that result is not as informative as it seems at first, since it is made invalid by a small change in the model. Thus, this chapter illustrates the interplay of

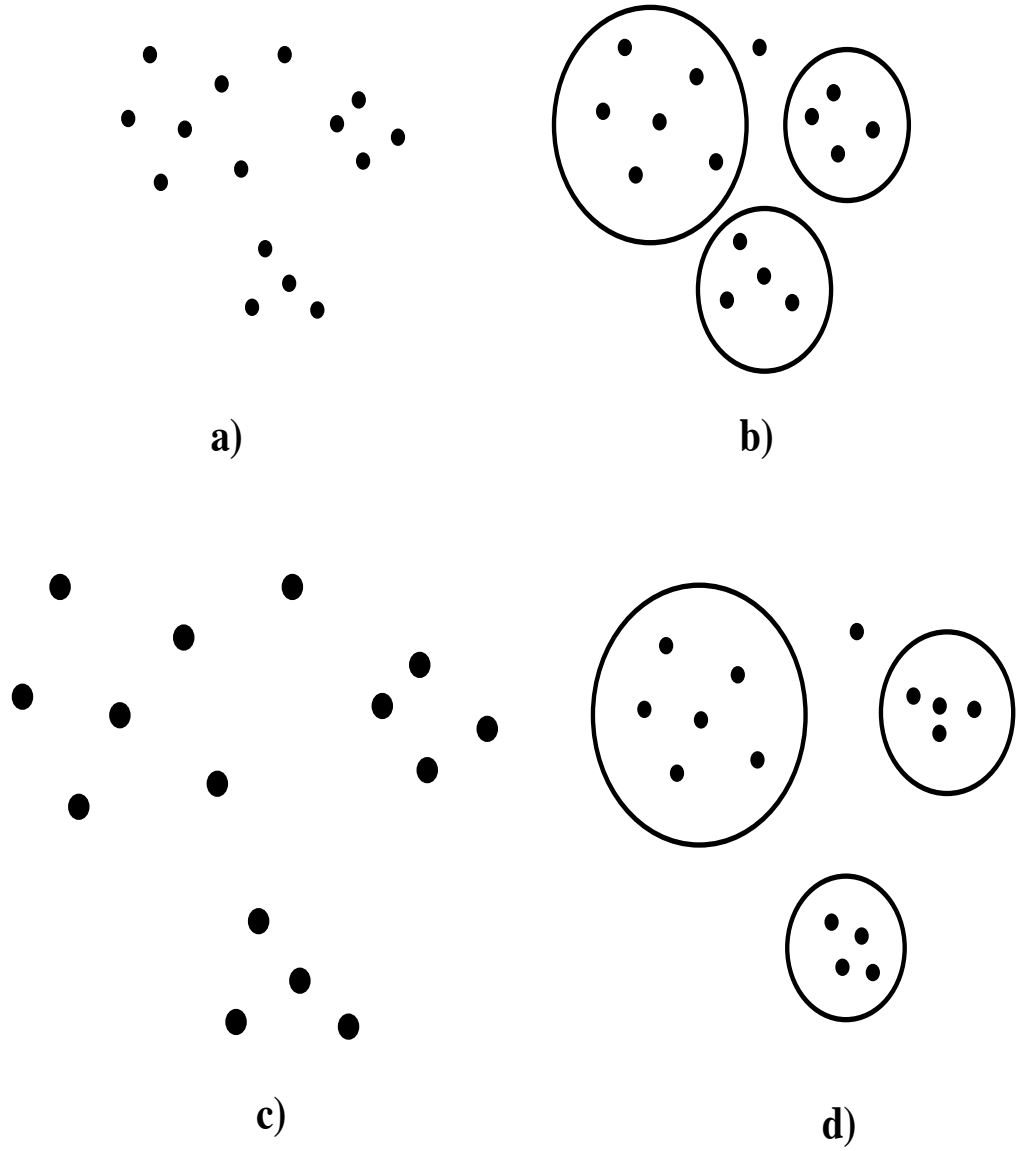

Figure 1: Clusters and clustering. Figure a) shows fifteen points drawn on the plane. The set of distances d between any two points is just the straight-line (Euclidean) distance, making it sensible to cluster the points by eye. Figure b) shows a plausible clustering into four clusters, including one cluster just consisting of a single point. An alternative clustering might add that point to one of the two neighboring clusters. Figure c) shows the same points, where all the distances have been proportionally scaled up, so the relative distances remain the same as before. Figure d) show altered distances (not simple proportional scaling) that are *consistent* with the distances and clustering shown in Figure b).

models and *theorems*, an issue that also arose in our discussion of Bell's and Arrow's impossibility theorems.

0.3 Clustering axioms and impossibility

We will use F to denote a clustering algorithm, and define $F(S, d)$ as the clustering created by algorithm F when given the set of elements S and the set of distances d between the pairs of elements in S. We can refer to d either as a set of distances between pairs of elements in S, or as a *distance function* of pairs of elements in S. Recall that the distance function d is very general, and need not have any relation to pairwise *Euclidean* distances of the elements.

What is a good clustering? Now, the distances (given by d) can vary, and the resulting clustering $F(S, d)$ may change as the distances change. So, for a fixed set of elements, S, and a fixed clustering algorithm, F, the resulting clustering $F(S, d)$ is a *function* of d. With these definitions, we can ask:

> What properties should a good clustering algorithm, F, have? Equivalently, what properties should a good clustering function, $F(S, d)$, have?

0.3.1 Kleinberg's axioms

Jon Kleinberg, in 2002 [3], proposed three *axioms* (or *requirements*), that a good clustering algorithm, F, should obey,[1] and then proved that *no* clustering algorithm can simultaneously obey all of them – it is *impossible*.

The axioms and their meanings Kleinberg's three axioms are:

1. Scale-invariance:

   If two sets of distances differ only by a *multiplicative constant*, then the resulting clusterings given by a good clustering algorithm F should be the same.

   In more detail, suppose d and d' are two sets of distances between pairs of elements of set S. If there is a constant number α, where $d'(i, j) = \alpha \times d(i, j)$ for each pair of elements (i, j) in S, then clustering $F(S, d)$ should be the same as clustering $F(S, d')$.

   The intuition for this axiom is that good clustering algorithms should *not* be influenced by the *absolute* values of the distances given to them, but only by the *relative* values of the distances. Scaling of the distances should have no affect on

[1] As we will see, there is disagreement about the appropriateness and robustness of Kleinberg's three axioms, but the paper was widely read because despite the importance of clustering, there were very few theoretical results concerning the foundations and fundamental properties of clustering. Kleinberg's axioms and paper opened the door to more research on foundational issues of clustering.

the resulting clustering. In Figure 1 c), the distances are proportionally scaled up from the distances in panel a), and it is reasonable that good clusterings should be the same for both sets of distances.

The scale-invariance axiom is generally accepted as reasonable.

2. Richness:

   A good clustering algorithm, F, should have the property that for *any* clustering, $C(S)$ of set S, there is some set of distances, d, such that $F(S, d) = C(S)$.

   In other words, no clustering is impossible if all sets of distances are possible. For any way of dividing up the elements of S into non-overlapping clusters, $C(S)$, there will be some set of distances (maybe not realistic ones in a given application) d, such that algorithm F, given S and d, will create exactly those clusters.

   Note that because of the Richness Axiom, there must be an extreme set of distances, d, where algorithm F puts every element in S into it's own, *separate* cluster. We call that clustering the *anti-social* clustering. Symmetrically, there must be an extreme set of distances, d', where algorithm F puts all of the elements into one *single* cluster. These two extreme clustrings are in addition to all of the other clusterings created by algorithm F for less extreme distances.

   The Richness Axiom has been criticized for allowing extreme clusterings, something that is not allowed by many clustering algorithms. However, the Richness axiom does not require that those extreme clusterings would be created by "reasonable" distances, i.e., ones that would ever be encountered in practice.

3. Consistency:

   Consider a clustering $F(S, d)$ for a set of distances, d, and let d' be any different set of distances, $d' \neq d$. Then d' is called *consistent* with the pair $(d, F(S, d))$, if $d'(i, j) \geq d(i, j)$ for every pair of elements (i, j) that are in *different* clusters of $F(S, d)$; and $d(i, j) \leq (d(i, j)$ for every pair of elements (i, j) that are in the *same* cluster of $F(S, d)$.

   Stated differently, d' is consistent with the pair $(d, F(S, d))$ if, when changing distances from d to d', the distances *between* clusters of $F(S, d)$ only *increase* or stay the same, and the distances *inside* clusters of $F(S, d)$ only *decrease* or stay the same. Figure 1 d) gives an example a set of distances, d', that is consistent with the pair $(d, F(S, d))$ from panel b).

   A clustering algorithm, F, is called *consistent* if for any two set of distances, d and d', when d' is consistent with the pair $(d, F(S, d))$, then the clustering $F(S, d')$ is the *same* as the clustering $F(S, d)$.

   That is, starting from a clustering $F(S, d)$, if the distances between pairs of elements inside the same cluster stay the same or *decrease*, and the distances between

pairs of elements in different clusters stay the same or *increase*, then the clustering does *not* change.

Note that these uses of the word "consistency" are different from its use in our discussion of the GS theorem in Chapter **??**.

Consistency is a very strong and highly disputed axiom. For example, in Figure 1 b), if the upper right cluster, along with the cluster containing only a single point, were pulled a million miles to the right of the upper left cluster, it would then be sensible to merge the single point cluster into the upper right cluster. This is already visually suggested in Figure 1, panel d). But the consistency axiom does not allow such mergings.

Later in this chapter, we will discuss alternatives to the consistency axiom that are less extreme and still lead to impossibility. But first, we present the main impossibility result from [3].

Theorem 0.3.1 *There is no clustering algorithm that simultaneously obeys the axioms of scale invariance, richness, and consistency.*

Theorem 0.3.1 was first proved in [3], but our discussion follows the simpler proof presented in the dissertation of Margarita Ackerman [1].

Proof of Theorem 0.3.1: Consider two clusterings: the anti-social clustering, we call C_0, which puts each element in its own cluster; and C_1, which is some other clustering different than C_0. By the *richness* axiom, each clustering C is associated with some set of distances, d, where C is the clustering $F(S, d)$ created by algorithm F given the distances d. So, there must be some set of distances, d_0, which leads F to create clustering C_0; and also some set of distances, d_1, which leads F to create clustering C_1.

We define $\max(d_0)$ be the *largest* number in the distances d_0. For example, if there are three elements in S, numbered 1 through 3, and the pairwise distances are $d_0(1,2) = 4; d_0(1,3) = 2; d_0(2,3) = 7$, then $max(d_0)$ is 7.

Next, let α be some number where $\alpha \times d_1(i,j)$ is greater than $max(d_0)$, for each pair of elements (i,j) in S. Then, define a new set of distances $d'(i,j) = \alpha \times d_1(i,j)$, for each pair of elements (i,j) in S. Algorithm F will create a clustering, $F(S,d')$, for the distances d'. What will that clustering be? We will first argue that it must be C_1; and we will next argue that it must be C_0.

It must be C_1 Since clustering C_1 is generated by F for the distances d_1, and the distances in d' are all created by multiplying the distances in d_1 by a single number α, the *scale-invariance* axiom requires that algorithm F creates the same clustering, C_1, for the distances in d'. That is, $F(S,d') = F(S,d_1)$.

No, it must be C_0 In C_0 every element in S is in its own, separate cluster, so that each (i,j) is a *between-clusters* pair. Then, since $d'(i,j) > d_0(i,j)$ for each pair (i,j), distance

d' is consistent with the pair (d_0, C_0). So, by the *consistency* axiom, algorithm F must create the same clustering C_0 for the distances in d'. That is, $F(S, d') = F(S, d_0)$.

Thus, one line of reasoning leads to the conclusion that the clustering that F will produce for d' will be C_1, while the other line of reasoning leads to a different conclusion, namely that the clustering for d' will be C_0. Since clusterings C_0 and C_1 are different, this is a contradiction. In reaching this contradiction, we used all three of Kleinberg's clustering axioms, and hence we conclude that it is *impossible* to have an algorithm that creates clusterings from distances and obeys all three requirements: *scale invariance*, *richness*, and *consistency*. ■

0.3.2 More realistic axioms

The consistency axiom is pretty restrictive, and it is not clear why we should expect any clustering algorithm to obey it. But there are two more-realistic consistency-related axioms that have been studied, where both still lead to impossibility results. One axiom, called *refinement-consistency*, was introduced in [3]; and another, called *outer-consistency*, was introduced in [1].

Outer-consistency As in the discussion of consistency, let d and d' be any two sets of distances, where $d' \neq d$. Then d' is called *outer-consistent* with the pair $(d, F(S, d))$, if $d'(i, j) \geq d(i, j)$ for every pair of elements (i, j) that are in *different* clusters of $F(S, d)$; and $d'(i, j) = d(i, j)$ for every pair of elements (i, j) that are in the *same* cluster of $F(S, d)$.

A clustering algorithm, F, is called *outer-consistent* if for any two sets of distances, d and d', when d' is outer-consistent with the pair $(d, F(S, d))$, the clustering $F(S, d')$ is the *same* as the clustering $F(S, d)$.

That is, starting from a clustering $F(S, d)$, if the distances between pairs of elements inside the same cluster stay the same, and the distances between pairs of elements in different clusters stay the same or increase, then the clustering does *not* change.

The condition of outer-consistency is somewhat more justified than the condition of consistency, because there are well-known clustering algorithms that are outer-consistent (see [1]).

Note that the definition of outer-consistency allows an outer-consistent algorithm to produce *different* clusterings when the distances inside the same cluster *decrease*, and the distances between clusters stay the same or increase. That outcome is not allowed by an algorithm that obeys Kleinberg's original consistency axiom.

Review question: Explain why every consistent clustering algorithm is also outer-consistent. Is it then correct to say that outer-consistency is a weaker assumption than consistency?

Since outer-consistency allows outcomes that are not allowed by consistency, it is *conceivable* that impossibility results involving consistency would not hold when only outer-consistency is required.

However In the proof we presented for Theorem 0.3.1, every element in S is in its own cluster in $C_0 = F(S, d)$, so d' is outer-consistent with the pair $(d, F(S, d))$. Therefore, the same proof also establishes:

Theorem 0.3.2 *There is no clustering algorithm that simultaneously obeys the axioms of scale invariance, richness, and outer-consistency.*

Refinement Suppose $F(S, d)$ and $F(S, d')$ are two different clusterings of a set S, for distance functions d and d', respectively, and recall that a cluster is a subset of elements of S. If every cluster in $F(S, d')$ is a subset of a cluster in $F(S, d)$, then clustering $F(S, d')$ is called a *refinement* of clustering $F(S, d)$. That is, clustering $F(S, d')$ may differ from clustering $F(S, d)$ by *splitting* some clusters in $F(S, d)$ into two or more clusters. Note that a set is considered a subset of itself, so the definition of refinement also allows two clusterings to be identical. See Figure 2.

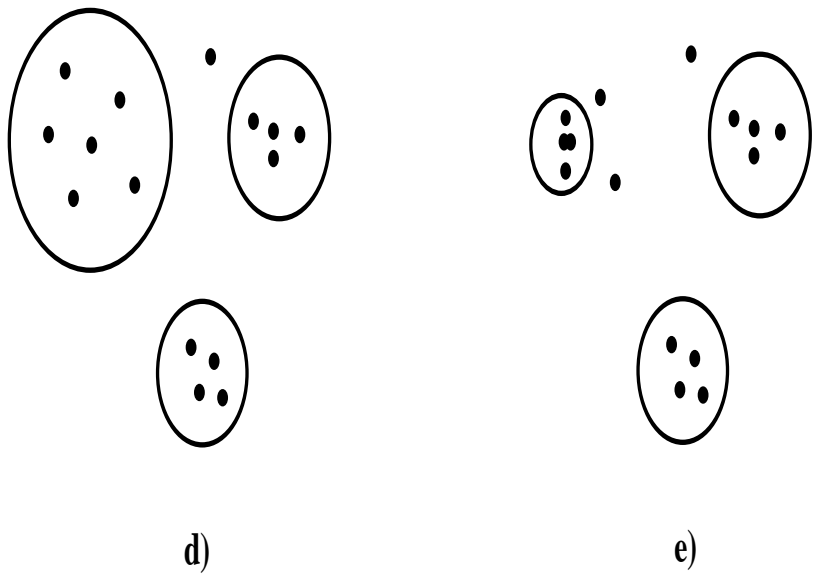

Figure 2: Panel d) is the same clustering shown in panel d) of Figure 1. In panel e), the points in the upper-left cluster have been moved so that the distance between any two points in the cluster has either decreased or remains unchanged. Every other point in Panel e) is unchanged from it placement in Panel d). Visually, it now seems more natural to refine (split) the upper-left cluster into three clusters, two of which contain only a single point. The refined clustering is shown in panel e).

Refinement-consistency When should refinements of clusterings be allowed by a clustering algorithm?

When we decrease, or maintain, distances within clusters and increase or maintain distances between clusters, the consistency axiom requires that a clustering algorithm keeps the *same* clusterings. But with such changes in distances, it may be more meaningful to allow the second clustering to be a *refinement* of the first one. This leads to a change in the axiom of consistency to:

The axiom of refinement-consistency A clustering algorithm, F, is called *refinement-consistent* if for any two sets of distances, d and d', when d' is consistent with the pair $(d, F(S, d))$, then the clustering $F(S, d')$ must be a *refinement* of the clustering $F(S, d)$. Kleinberg in [3], claims:

Theorem 0.3.3 *There is no clustering algorithm that simultaneously obeys the axioms of scale invariance, richness, and refinement-consistency.*

It is brittle Note that if a clustering algorithm F does *not* allow the anti-social clustering, where each element in S is in its own cluster, then the proofs given for Theorems 0.3.1 and 0.3.2 break down. The same break-down is stated in [3] to be true for Theorem 0.3.3. In fact, in [3] it is claimed that there is an algorithm that *does* obey the axioms of *scale-invariance*, *richness*, and *refinement-consistency*, *if* the algorithm is explicitly not allowed to produce the anti-social clustering. Because of that, Kleinberg calls Theorem 0.3.3 true, but "brittle".

A more comprehensive critique of Kleinberg's axioms and impossibility theorem appear in [2]. That paper also proposes an alternative set of axioms that are similar in spirit to Kleinberg's, but allow clustering algorithms that *do* obey the alternative *scale-invariance*, *richness* and *consistency* axioms.

Review question In the proof we presented for Theorem 0.3.1, C_0 is the anti-social clustering. So, if the anti-social clustering is not allowed, that proof does not work. Does this prove Kleinberg's claim that Theorem 0.3.1 does not hold if an algorithm is never allowed to produce the anti-social clustering? Explain.

0.4 Take-home lessons

The story of Kleinberg's impossibility theorems, and the critiques of them, illustrates an important point. As was true for Bell's theorems and for Arrow's theorems, when trying to prove that some task or phenomena is impossible, we must first detail a precise *model*

of what we mean by that task or phenomena.[2] Only after a model is fully specified do we have the possibility of proving impossibility. But, the proof of impossibility in terms of the chosen model, while suggestive of possible impossibility outside of the model, is only assured for that precise model. One can hope (or fear) that the impossibility result extends beyond the precise model, but it is not always so.

In the case of Bell's theorems, impossibility does extend beyond Bell's original model based on EPR experiments. For example, Bell's impossibility results were later extended (and strengthened) to GHZ experiments. Arrow's impossibility theorems also extend beyond his original voting model, for example in the GS theorem where the voting mechanism determines a *single* winner, rather than a full rank-ordering of the candidates. But, Kleinberg's original result, as suggestive and impressive as it is, seems to have a more limited scope beyond the precise model used for that result. This is not a failing, but just part of the ebb-and-flow of applied research. It is worthwhile repeating the quote from Clarke:

> The only way of discovering the limits of the possible is to venture a little way past them into the impossible.

0.4.1 Proof vs practice

There is another subtle point I want to make about the value of impossibility proofs, illustrated by the brittleness of Theorem 0.3.2. As noted earlier, there are clustering algorithms that obey the axioms of *scale-invariance*, *richness* and *refinement-consistency*, *provided* that those algorithms *forbid* the anti-social clustering, where each element is in its own cluster. And since the anti-social clustering is not likely to be informative in real applications, this is a pretty good outcome for practical clustering.

But, the *Gold-Standard* of algorithm design is a mathematical *proof* of whatever properties are claimed or desired for an algorithm. The goal of mathematically-oriented algorithm designers is to create an algorithm whose properties are *proved*, not just observed in practice.[3]

Now imagine such an algorithm designer, before Kleinberg's paper, who is trying to design a clustering algorithm that *always* obeys the above three axioms, and who wants to *prove* that their proposed algorithm obeys the axioms. That algorithm designer would always fail, no matter what algorithm they devised, or how well it performed in practice. Theorem 0.3.2 guarantees that. But, before Keinberg's paper, they wouldn't know this for sure, or know why they were failing. So, they might just keep trying to find that ellusive algorithm.

[2] When we discuss Gödel's theorems, we will see this issue on steroids, because the gap there between what the precise model says, and the imprecise way that people sometimes state Gödel's theorems, is huge.

[3] The classic joke (but actually making a serious point about the importance of theory) is the question: "Sure it works in practice, but does it work in theory?"

After the proof of Theorem 0.3.2, the designer, while perhaps disappointed, would now understand the futility of their efforts, and the need to restrict their focus. And, seeing the way that the proof fails, after a small change in the model, can guide the designer in that refocussing.

The take-home lesson here is that while Kleinberg's theorems may have limited impact on the practice of clustering, they have real impact on the logic and mathematical investigation of clustering.

0.5 Optional Exercises

1. Generally, the anti-social clustering (where every element is in its own cluster) is not likely to be informative or useful for real clustering problems. But that is not always true. Give a pictorial example where the anti-social clustering might be highly informative.

2. Theorem 0.3.1 can be proved by letting C_0 be the clustering where all of the elements in S are in *one, single* cluster, instead of where C_0 is the anti-social clustering. That proof is a modification of the proof presented for Theorem 0.3.1. Find this modified proof.

3. *Inner-consistency* is a property of a clustering method that is symmetric to outer-consistency. It is defined as:

   For two sets of distances $d' \neq d$, d' is *inner-consistent* with $(d, F(S,d))$, if $d'(i,j) = d(i,j)$ for every pair of elements (i,j) that are in *different* clusters of $F(S,d)$; and $d'(i,j) \leq (d(i,j)$ for every pair of elements (i,j) that are in the *same* cluster of $F(S,d)$.

   A clustering algorithm, F, is called *inner-consistent* if for any two sets of distances, d and d', when d' is inner-consistent with $(d, F(S,d))$, the clustering $F(S,d')$ is the same as $F(S,d)$.

   Prove that there is no clustering algorithm that obeys the axioms of *scale-invariance*, *richness*, and *inner-consistency*.

4. If an algorithm is both outer-consistent and inner-consistent, is it necessarily consistent? If no, give an example; and if yes, prove it.

5. The axioms of *consistency*, *outer-consistency*, *refinement-consistency* and *inner-consistency* all involve increasing or maintaining distances between clusters; and decreasing and maintaining distances inside clusters. But, changes in distances might also decrease distances between clusters, and/or increase distances inside clusters.

   Give examples (in dot figures, as in Figure 1) where such changes might be reasonable.

Do any of the impossibility results discussed in this chapter apply when such changes are allowed? At first it seems that they might, because if distances between clusters increase when moving from distance d to distance d', then distances decrease when moving from d' to d. Is this line of reasoning helpful?

Bibliography

[1] M. Ackerman. *Towards theoretical foundations of clustering*. PhD thesis, University of Waterloo, Department of Computer Science, 2012.

[2] M. Ackerman and S. Ben-David. Measures of clustering quality: A working set of axioms for clustering. In *Neural Information Processing Systems 21*, 2008.

[3] J. Kleinberg. An impossibility theorem for clustering. In *Advances in Neural Information Processing, 2002*, pages 463 – 470, 2002.