Lecture 13 - 6: Similarity Measures-Cost functions

Tabaré Pérez

May 6, 2020

So the question that we now need to discuss is how to define the clustering cost.

And this is a real question. Let's consider the following sequence of points. I'm intentionally going to draw twice the same points.

This is one set of your points, and I am copying it as much as I could to make it look similar:

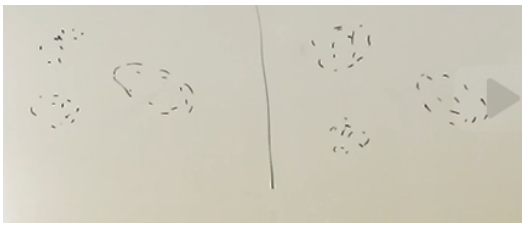

Figure 1: Fig. 01

So looking at these points, let's say I'm forced to divide it into two clusters. Somebody told me I have to divided into two clusters. So maybe I can divide them this way into two clusters.

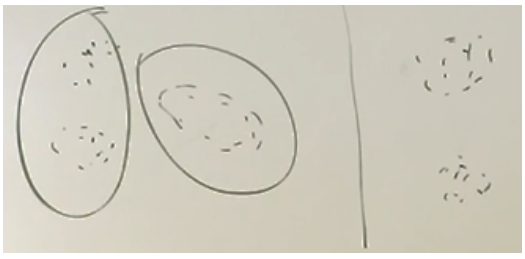

Figure 2: Fig. 02

Alternatively, I can divide them this way into two clusters.

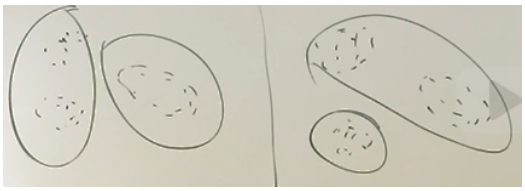

Figure 3: Fig. 03

So how do I know which one of them is better? So I need to have a cost which will take any possible partitioning of this points and tell me, this one has cost 5, this one maybe has cost 2, that's why this one is preferable for me.

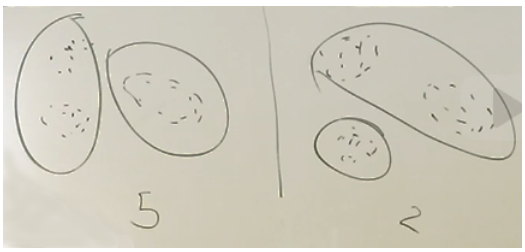

Figure 4: Fig. 04

So what we will need to do, and let's just look one step farther where we are going with it in the lecture, if we have this kind of cost, we can then look at the optimization algorithm which will go and find the best partitioning with this cost. But first of all, we need to decide how do we define the cost?

And there are many, many ways to define the cost of the partitioning.

So the general assumption that we will make here, when we are looking at the cost of a specific partitioning, (and you notice I'm going for 1 to K because we have K clusters) we would assume that it will be sum of the costs of individual clusters:

$$\text{cost}(C_1 \dots C_K) = \sum_{j=1}^{K} \text{cost}(C_j) \tag{1}$$

So in other words, it's very intuitive that you would say, the goodness of this whole partition would be sum of how good is every individual cluster.

So now we reduce our question to the question of cost of the specific clusters. For instance, here we are looking at the specific cluster and we need to define its cost.

$$\text{cost}(C) \tag{2}$$

This is a specific cluster. So how can we define cost? Again, there are many ways. In some ways, this cost should reflect how homogeneous, how consistent is this cluster. For instance, if we look at a specific cluster, we can ask how far are the most remote points. We can look at the **DIAMETER**:

$$\text{cost}(C) = \text{DIAMETER} \tag{3}$$

This is option one. We can do something else. We can look the average distance between all the points:

$$\text{cost}(C) = \begin{cases} \text{DIAMETER} \\ \text{AVERAGE DISTANCE} \end{cases} \tag{4}$$

And you already can see that whenever you are committing to specific measurements usually defining how it would look like:

- If you're looking at the diameter, then the outlier will be determining your cost.
- If you look at average distance and all the members in some way will be contributing.

The measures, the cost that I would like to introduce today, which we would use very commonly would actually bring in these representatives, and what this measure will do, it will determine the cost of the cluster by comparing the distance between all the members of the cluster and the specific representative. It means that whenever I am giving you a specific cluster, I have to give you the representative because then you would be able to go and compute for me the cost. So given some class C and a particular representative z, I would just say that my cost would be:

$$\text{cost}(C, z) = \sum_{i \in C} \text{distance}(x^{(i)}, z) \tag{5}$$

The sum for all the points i in C of some measurements of distance between $x^{(i)}$ and z.

So we went through all the points that belong to this particular cluster and compare its distance with the representative z.

So now the question of computing the cost of a cluster reduced to a question of computing the similarity of the distance between two vectors, like in this case, between the representative and a specific point in a cluster.

So the first similarity measures that I will introduce you may have seen in many other contexts. It's called **COSINE SIMILARITY**.

What it does, it takes two vectors and looks at the angle between these two vectors. So in this case, if you're given a vector $x^{(i)}$ and the vector $x^{(j)}$, then the cosine similarity between them would be their corresponding dot product divided by the product of their norms:

$$cos(x^{(i)}, x^{(j)}) = \frac{x^{(i)} \cdot x^{(j)}}{\left\|x^{(i)}\right\| \left\|x^{(j)}\right\|} \tag{6}$$

Another common measures that is used is actually a distance. Not a similarity, a distance metric which just looks at the distance between points in the space and it's called **EUCLIDEAN SQUARED DISTANCE**:

$$\text{distance}(x^{(i)}, x^{(j)})^2 = \left\|x^{(i)} - x^{(j)}\right\|^2 \tag{7}$$

$$\text{distance}(x^{(i)}, x^{(j)})^2 = [U - V]^\intercal \cdot [U - V] \tag{8}$$

So this distance measure takes again, two points, and looks at the square of that norm.

Now let's just look at this two different ways to compare vectors in order to understand the difference.

So, for instance, the cosine similarity is not sensitive to the magnitude of the vectors. So what it tells us, for instance, in terms of our Google News application, that if we choose cosine to compare between two vectors representing two different stories, that this measurement will not take it into account how long is the story.

On the other hand, Euclidean distance will be sensitive to the lengths of the story.

And there are lots and lots of other measurements that you can find in the literature.

I would be using the Euclidean square distance and you would shortly discover why.

But the questions that you need to be asking yourselves, and I will ask you later, is what makes the match between the similarity measure and the algorithm and when it is important and when it is not important.

But at any rate, let's say I convinced you at this point to pick up Euclidean squared distance.

And now I want to write for you the cost of the partitioning.

I am coming back to this question, how to compute the cost of the partitioning? And as I said earlier, what I would want to do, I would want to define

the cost of the partitioning in terms of the distance, in this case, Euclidean squared distance between the elements of the cluster and the representative.

So in order for me to write this cost, now it is not enough to write just the partitions themselves. I need to have also the representatives. So you would be given a partitioning of K clusters and then K representatives. And we will use squared Euclidean distance to go over every single cluster. That's what I'm going to do. I am going to go from cluster 1 to cluster K. And then, within each cluster, I will take the points that belong to this cluster, all the indexes of points that belong to cluster C_j, and then compute squared Euclidean distance between this point and the representative of this cluster:

$$\text{cost}(C_1 \dots C_K, z^{(1)}, \dots, z^{(K)}) = \sum_{j=1}^{K} \sum_{i \in C_j} \left\| x^{(i)} - z^{(j)} \right\|^2 \tag{9}$$

And you may be looking at it and thinking, why do I need to carry both? Maybe I can compute representative if I know the clusters. This is true. And as we will continue our lecture, we will see how to unify them together. But at this point, I would want you to think about this cost, which we will be optimizing.

I would assume that we are given the clusters, we are given the representatives, and this is our way to compute the cost with squared Euclidean distance.