Lecture 13 - 7: The K-Means Algorithm: The Big Picture

Tabaré Pérez

May 6, 2020

Now, you have the cost.

And the next question is, fine, I'm giving partitioning and I'm giving the representative: How do I find the best?

Because it's exactly what we have in the supervised case when somebody gave you an objective, and given this objective, you needed to provide a mechanism to find the best separators that satisfy this objective. So here it's exactly the same story.

We have the cost now and we need to find a way to find the best partitioning and the best representatives. Now, as we know, if we have a number of points, how many different partitions can we produce, meaning how distinct partitions of size K we can produce? So this would be exponential in the number of points, which means that we cannot just go one partition at a time, check the cost, and go to the next one.

We cannot do this computation explicitly because the search space of partitions is so large. Therefore, we need to have an algorithm which would tell us how to navigate in this space to get to the right partitions. So at this point, we finished talking about a clustering cost and similarity measure. And the only remaining point for today is actually how, given this cost, we can find the best partition.

And I will start introducing you now the K-means algorithm, which achieves exactly this goal.

So the last part of our lecture for today is **K-MEANS CLUSTERING ALGORITHM**.

So the way I would like to explain you, before I actually write the formulas, I would like to show you how this algorithm works, how does it defines the clusters.

And intuitively speaking, what it will do, it will take a selection of points, the whole space of points, and then randomly assigned the representatives.

Then every representative will draw to itself the participants which are closest to it.

Now we will have several clouds, which are associated with this representative. But when this assignment happen, we may decide to select a new representative.

And after the representatives are reassigned, we can now go and draw a different partitions. And we will continue this process until convergence.

So let me show you how it works in a specific example.

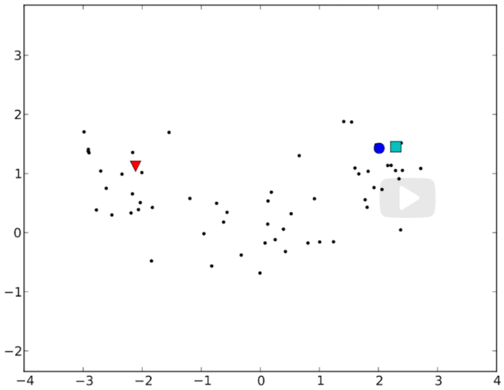

Figure 1: K-means example

So in this particular instance, let's just look at the example before we see the detail writing of the algorithm.

Let's see how it may be working on a set of points. So let's say here, I am deciding to identify three clusters. So we would have red, blue, and cyan. This is our initialization. You can see that these centers were totally randomly selected.

The next thing that, we will do, we would let every representative to draw the constituents which are closest to it.

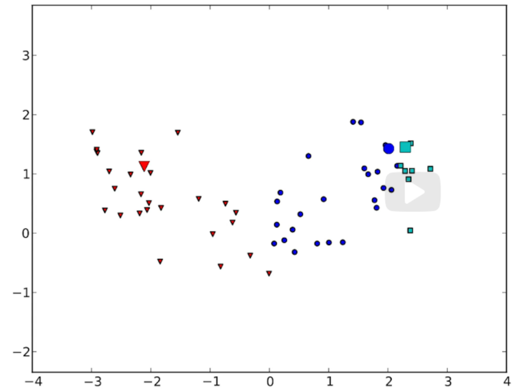

Figure 2: K-means example

So you see that all the points which are closer to the red representative happen to be close to. And similarly, you can see the blue and the green ones.

Now even before I will show you the next slide, you can look at it and say, oh, this is really funny because in this blue cloud the representative, which is supposed to represent all of them, is somewhere on the outskirts. So the natural thing is actually to move blue to be really here closer to the center and similarly to move red to be closer to the center. And that's exactly what we will do now. And we move them closer.

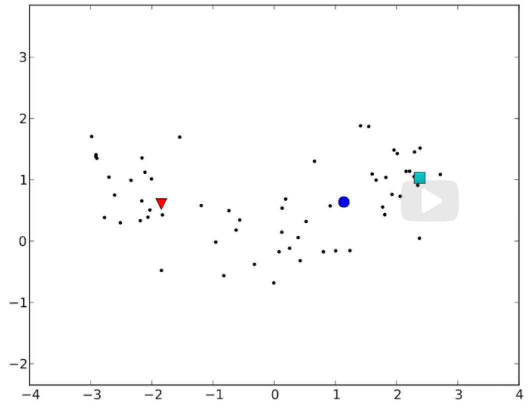

Figure 3: K-means example

So we have a red in the middle, blue in the middle, and cyan closer to its own original representatives. But now what we can do, we can again find for each point the representative which is closest to it. And we need to do it anew because, as you realize here, now since the representative moves, for every point, they need to decide again who will be the closest representative. And in this case, we can see, again, there was some movement. We see more red and some points which were originally maybe blue became red and so on.

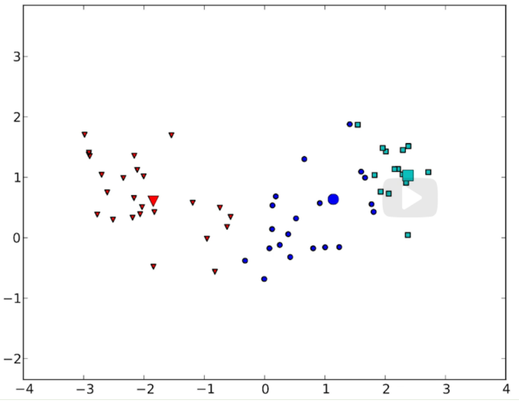

Figure 4: K-means example

And we can continue this process again. Blue is located very far from its constituent. The more natural move is to move in more central location. And that's exactly what we see here.

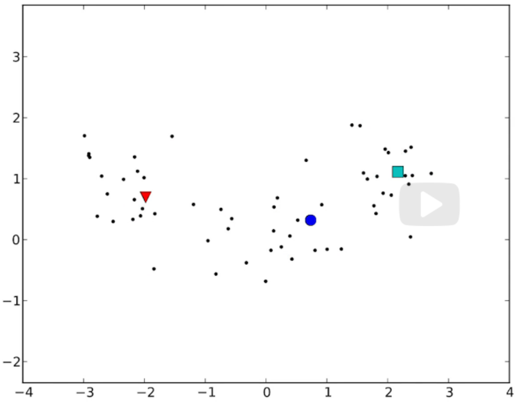

Figure 5: K-means example

And we can continue this process.

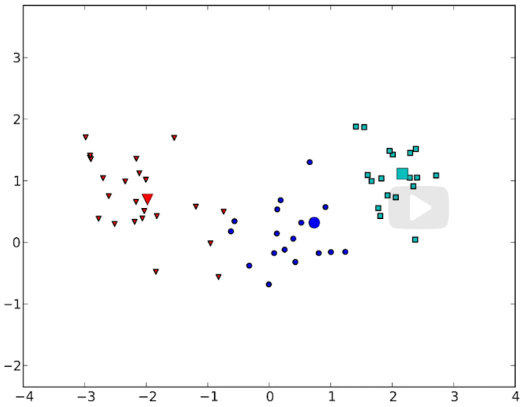

Figure 6: K-means example

And eventually, we will generate more stable clustering, which divides all this points.

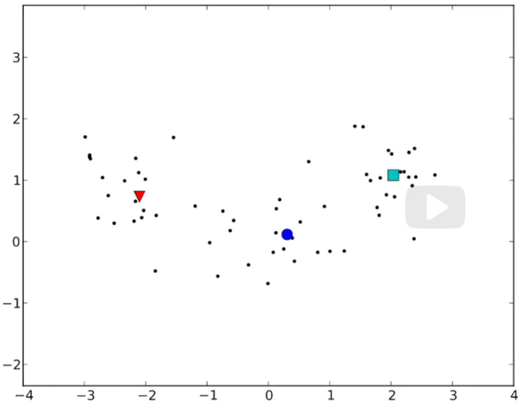

Figure 7: K-means example

So what you've seen here is an example of a run of K-means algorithms. And we see that there were two parts.

One part is when we randomly assigned the representative, and then every point was looking for the best one, for the best color for itself.

And then we find we devoted the representative according to that cloud and we continue this process until convergence.

So now, let's look more formally on how this can be described in terms of our cost that was just introduced before we went into this example.

So now let's start writing the K-means clustering algorithm. We would follow the steps that I just demonstrate to you on the example. So K-means clustering we will start by randomly initializing our representative.

1. Randomly select $z^{(1)} \dots z^{(K)}$. And we will talk shortly about different strategies of how to do random initialization. But for now, just you assume it totally randomly selected them.

2. And then we will iterate over the following two steps:

   - 2.1) Given $z^{(1)} \dots z^{(K)}$, assigns xs to the closest z. So the first step would be, given fixed $z^{(1)} \dots z^{(K)}$ the representatives are given to you, assign x_s to the closest z. So at this point, we are given the representatives. We will cycle over all the our xs and find the one which is closest. What I would like to do is actually to write the cost of this assignment:

$$\operatorname{cost}(z^{(1)} \dots z^{(K)}) = \sum_{i=1}^{n} \min_{j=1 \dots K} \left\| x^{(i)} - z^{(j)} \right\|^2 \tag{1}$$

So our cost would be driven by this $z^{(1)} \dots z^{(K)}$ that are fixed, and what we will have to do, is to go from all the points in our set and then find for it the closest representative.

So we have the representative from 1 to K, j from 1 to K and now we will compute $\left\| x^{(i)} - z^{(j)} \right\|^2$ and so this is our cost.

And we completed this assignment and now we have a new partitions.

Since we have a new partition, we need to reassign that representative which would be good for that partition:

- 2.2) Given $C_1 \dots C_K$, find the best representatives z.

Now what do we mean by the best? We mean the ones which actually minimize our cost. So in this case, the cost is driven by the partition $C_1 \dots C_K$ and the best we can do here is to find:

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

over all possible z_s, $z^{(1)} \dots z^{(K)}$, those which will minimize the total cost of the partitioning. So we will go j from 1 to K and then for every point in the cluster, we will compute its distance from the representative.

So again, what are we doing here?

We have the partitions. We will go and try to find the best assignments of the representatives in such a way that if for every point we compute its distance from the representative, we achieve the minimum.