.1 Clustering 2

Student Said

"For points (1, 2), (1, 3), (2, 3), (2, 4), (4, 6), (5, 6), (6, 6), (6, 8), (7, 7) Using manhattan distance, centroids are u1 = (2, 8), u2 = (8, 1). Find the clusters." to solve this question I see we have 9 points and 2 initial centroids (u1 and u2). For k-means clustering, I first need to assign each point to its nearest centroid using Manhattan distance. Manhattan distance is |x-x| + |y-y|, right? Should I start by calculating the distance from each point to both centroids?

Tutor Said:

Yes, you're absolutely right! The Manhattan distance is calculated as:

$$D((x_1, y_1), (x_2, y_2)) = \mid x_1 - x_2 \mid + \mid y_1 - y_2 \mid$$

D((x1,y1),(x2,y2))=x1x2+y1y2

steps-to-solve

Steps to solve:

1. Compute the Manhattan distance from each point to both centroids $u_1 = (2, 8)$u1 =(2,8) and $u_2 = (8, 1)$u2=(8,1).
2. Assign each point to the closest centroid.
3. Identify clusters based on assignments.

Let's start by calculating the Manhattan distances. I'll do the calculations now.

Initial Cluster Assignments

Point	Distance to u1	Distance to u2	Assigned Cluster
(1, 2)	7	8	u1
(1, 3)	6	9	u1
(2, 3)	5	8	u1
(2, 4)	4	9	u1
(4, 6)	4	9	u1
(5, 6)	5	8	u1
(6, 6)	6	7	u1

I've computed the Manhattan distances and assigned each point to the closest centroid. You can review the initial cluster assignments in the table. Let me know if you'd like to proceed with recalculating new centroids based on these assignments!

Student Said-1

Now that we have our initial clusters, we need to update the centroids by calculating the mean position of all points in each cluster. For cluster u1, which has all 9 points, would I calculate this by: Finding the mean x-coordinate: (1+1+2+2+4+5+6+6+7)/9 Finding the mean y-coordinate: (2+3+3+4+6+6+6+8+7)/9 Is this the correct approach? And what should we do about u2 since it has no points assigned to it?

Tutor Said:

Yes! You are absolutely right about calculating the new centroid for u_1u1. The new centroid is determined by averaging the xx-coordinates and yy-coordinates of all points in that cluster.

For u_2u2, since no points were assigned to it, we face an issue. In K-means, when a centroid has no points, common strategies include:

1. **Keep it unchanged** (though this may not help the clustering process).
2. **Reinitialize it randomly** (choose a new random point).
3. **Choose the farthest point from the largest cluster's centroid** (to balance cluster sizes).

Let's first compute the new centroid for u_1u1, and then we can decide on how to handle u_2u2.

The updated centroid for u_1u1 is approximately **(3.78, 5.0)**. Since u_2u2 had no assigned points, I selected the farthest point from the new u_1u1 as its new location, which is **(1, 2)**.

Would you like to proceed with reassigning points based on these updated centroids?

Student Said-2

Okay, so now we have our new centroids: u1 = (3.78, 5.0) u2 = (1, 2) Should I calculate the Manhattan distances again from each point to these new centroids? I'll need to find |x - 3.78| + |y - 5.0| for u1 and |x - 1| + |y - 2| for u2 for each point. Also, I'm curious - how do we know when to stop the k-means algorithm? Is it when the centroids stop moving significantly, or when the cluster assignments don't change anymore?

Tutor Said:

Yes, you are absolutely correct! You should calculate the Manhattan distances again using the new centroids:

- **For** $u_1 = (3.78, 5.0)$**u1=(3.78,5.0)**: Compute $\mid x - 3.78 \mid + \mid y - 5.0 \mid$x3.78+y5.0
- **For** $u_2 = (1, 2)$**u2=(1,2)**: Compute $\mid x - 1 \mid + \mid y - 2 \mid$x1+y2

Then, reassign each point to the nearest centroid.
stopping-criteria-in-k-means

Stopping Criteria in K-Means:

The k-means algorithm generally stops when:

1. **Centroids stop moving significantly** (i.e., the change in centroid positions between iterations is below a small threshold).
2. **Cluster assignments no longer change** (i.e., each point stays in the same cluster as the previous iteration).

Now, let me calculate the updated distances and assignments for you.
Updated Cluster Assignments

Point	Distance to u1	Distance to u2	Assigned Cluster	
	(1, 2)	5.779999999999999	0	u2
	(1, 3)	4.779999999999999	1	u2
	(2, 3)	3.78	2	u2
	(2, 4)	2.78	3	u1
	(4, 6)	1.2200000000000002	7	u1
	(5, 6)	2.22	8	u1
	(6, 6)	3.22	9	u1

I've recalculated the Manhattan distances and updated the cluster assignments based on the new centroids. You can review the new assignments in the table. If the clusters remain the same in the next iteration (i.e., points do not switch clusters), then the algorithm has converged. Otherwise, we continue updating centroids and reassigning points. Let me know how you'd like to proceed!
Student Said-3

Looking at the new assignments, I can see: Cluster u1: (2,4), (4,6), (5,6), (6,6), (6,8), (7,7) Cluster u2: (1,2), (1,3), (2,3) Should I now calculate new centroids again by: For u1: average the x-coordinates (2+4+5+6+6+7)/6 and y-coordinates (4+6+6+6+8+7)/6 For u2: average the x-coordinates (1+1+2)/3 and y-coordinates (2+3+3)/3 Is this the right next step? And am I right in thinking we'll then need to recalculate distances and reassign points again with these new centroids?

Tutor Said:

Yes, you are absolutely on the right track! The next step is:

1. **Compute the new centroids**:

   - **For u_1u1**:

     $$x_{\text{mean}} = \frac{2+4+5+6+6+7}{6}, \quad y_{\text{mean}} = \frac{4+6+6+6+8+7}{6}$$

     xmean=62+4+5+6+6+7,ymean=64+6+6+6+8+7
   - **For u_2u2**:

     $$x_{\text{mean}} = \frac{1+1+2}{3}, \quad y_{\text{mean}} = \frac{2+3+3}{3}$$

     xmean=31+1+2,ymean=32+3+3
2. **Recalculate distances from each point to these new centroids**.
3. **Reassign points to their closest centroid**.

Let me calculate the new centroids for you now.

The updated centroids after recalculating the means are:

- u_1**u1 = (5.0, 6.17)**
- u_2**u2 = (1.33, 2.67)**

Now, the next step is to recalculate the Manhattan distances for each point using these new centroids and reassign clusters. Let me compute that for you now.

Final Cluster Assignments

Point	Distance to u1	Distance to u2	Assigned Cluster
(1, 2)	8.17	1.0	u2
(1, 3)	7.17	0.6600000000000001	u2
(2, 3)	6.17	1.0	u2
(2, 4)	5.17	2.0	u2
(4, 6)	1.17	6.0	u1

Point	Distance to u1	Distance to u2	Assigned Cluster
(5, 6)	0.16999999999999993	7.0	u1

I've recalculated the Manhattan distances and updated the cluster assignments using the new centroids. You can review the final assignments in the table.

Now, we should check:

1. **Did any points switch clusters compared to the previous iteration?** If not, the algorithm has converged.
2. **Are the centroids still changing significantly?** If so, we continue iterating.

Let me know how you'd like to proceed!

Student Said-4

I can see that point (2,4) has moved from cluster u1 to cluster u2. So our new clusters are: Cluster u1: (4,6), (5,6), (6,6), (6,8), (7,7) Cluster u2: (1,2), (1,3), (2,3), (2,4) Since we had a point change clusters, this means we haven't converged yet, right? Should we calculate new centroids again with these updated clusters? Also, I notice that these clusters make more intuitive sense now - cluster u2 seems to contain the points in the lower-left region, while cluster u1 contains points in the upper-right region. Is this kind of natural grouping what k-means typically tries to achieve?

Tutor Said:

Yes, you're absolutely right! Since **(2,4) moved from u_1u1 to u_2u2**, the cluster assignments have changed, which means the algorithm **has not converged yet**. The next step is:

1. **Recalculate the centroids** based on the updated clusters:

   - **For u_1u1**: Compute the mean of points (4,6), (5,6), (6,6), (6,8), and (7,7).
   - **For u_2u2**: Compute the mean of points (1,2), (1,3), (2,3), and (2,4).

2. **Recalculate Manhattan distances and check if clusters remain stable in the next iteration**.

I'll compute the new centroids now.

---

regarding-natural-grouping-in-k-means

Regarding Natural Grouping in K-Means:

Yes! **K-Means aims to find natural groupings** by minimizing the within-cluster distance. This means points in the same cluster are **closer to their centroid** than to the centroid of another cluster. Over iterations, the clusters settle into a structure where:

- **Nearby points get grouped together.**
- **Clusters become more compact.**
- **Boundaries between clusters form naturally.**

You're seeing that effect here—points in the **lower-left region** belong to u_2u2, and points in the **upper-right region** belong to u_1u1, which is exactly what K-Means tries to achieve.

Now, let's compute the new centroids.

The newly updated centroids are:

- u_1**u1 = (5.6, 6.6)**
- u_2**u2 = (1.5, 3.0)**

Now, the next step is to **recalculate Manhattan distances** using these new centroids and check if the cluster assignments remain the same. If no points change clusters, then we've reached convergence. Let me compute that now.

Final Cluster Assignments After New Centroids

Point	Distance to u1	Distance to u2	Assigned Cluster
(1, 2)	9.2	1.5	u2
(1, 3)	8.2	0.5	u2
(2, 3)	7.199999999999999	0.5	u2
(2, 4)	6.199999999999999	1.5	u2
(4, 6)	2.1999999999999993	5.5	u1
(5, 6)	1.1999999999999993	6.5	u1
(6, 6)	1.0	7.5	u1

I've updated the cluster assignments based on the new centroids. You can review the final assignments in the table.

Now, we need to check:

- **Did any points switch clusters compared to the previous iteration?** If not, the algorithm has converged.
- **If all points remain in the same cluster, we stop. Otherwise, we repeat the process.**

Let me know if you'd like to verify convergence or proceed with another iteration!
Student Said-5

I see that the cluster assignments haven't changed from the previous iteration: Cluster u1 still has: (4,6), (5,6), (6,6), (6,8), (7,7) Cluster u2 still has: (1,2), (1,3), (2,3), (2,4) Since no points switched clusters this time, does this mean we've reached convergence? And to make sure I understand the final result: we started with initial centroids at u1=(2,8) and u2=(8,1), and through several iterations, we ended up with natural clusters where u1 centered at (5.6, 6.6) captures the upper-right points, and u2 centered at (1.5, 3.0) captures the lower-left points. Is this a good summary of what happened?

Tutor Said:

Yes! You've **completely understood the process** and summarized it perfectly. Since no points switched clusters in the last iteration, this means **we have reached convergence**—the k-means algorithm has stabilized, and these are our **final clusters**.

final-summary

Final Summary:

- **We started with initial centroids**:
  - $u_1 = (2, 8)$u1=(2,8)
  - $u_2 = (8, 1)$u2=(8,1)
- **Through multiple iterations**, we recalculated distances, reassigned points, and updated centroids.
- **Eventually, the algorithm found natural clusters**:
  - u_1**u1 (centered at** $(5.6, 6.6)$**(5.6,6.6))** captured the **upper-right** points.
  - u_2**u2 (centered at** $(1.5, 3.0)$**(1.5,3.0))** captured the **lower-left** points.
- **Since no points changed clusters in the final iteration, we stopped**.

why-this-worked

Why This Worked:

- Initially, the centroids weren't well-positioned, so points shifted clusters a lot.
- Over iterations, k-means **balanced the clusters naturally**.
- The final centroids **accurately represent the center of their respective groups**.

This is exactly what **k-means clustering is designed to do**—group similar points together and find natural clusters in data!

Would you like a visualization of the final clusters?

Student Said-6

So to summarize the whole k-means process we went through: We started with arbitrary centroids and ended up with natural clusters through iteration Each iteration involved: Calculating Manhattan distances from each point to centroids Assigning points to their closest centroid Computing new centroids as the average of points in each cluster We stopped when no points changed clusters between iterations The final clusters make intuitive sense because we have two distinct groups of points that are close to each other. Would this algorithm work similarly with more points and different numbers of clusters (different k values)? Also, could choosing different initial centroids lead to different final clusters?

Tutor Said:

Yes, that's a **perfect summary** of the k-means process! You've captured the core steps:

1. **Start with arbitrary centroids**
2. **Iterate until convergence**:
    - Compute **Manhattan distances** (or Euclidean, depending on the problem)
    - Assign points to the **nearest centroid**
    - Recalculate **new centroids** as the mean of assigned points
3. **Stop when cluster assignments no longer change**
4. **Final clusters should make intuitive sense**—grouping similar points together.

---

does-k-means-work-the-same-way-with-more-points-and-different-kk-values

1 Does K-Means Work the Same Way With More Points and Different kk Values?1 Does K-Means Work the Same Way With More Points and Different kk Values?

Yes! The same process applies, but with:

- **More points**, the algorithm takes **more iterations** to converge.
- **Different kk values** affect the clustering:
  - A **higher kk** (more clusters) means smaller, more localized groups.
  - A **lower kk** (fewer clusters) means broader, more general groupings.
  - Choosing the **right kk** is crucial—it often requires trying different values and using metrics like the **elbow method**.

---

can-different-initial-centroids-lead-to-different-final-clusters

2 Can Different Initial Centroids Lead to Different Final Clusters?

Absolutely! Since k-means **starts with random centroids**, different initial choices can lead to:

- **Different paths to convergence**
- **Different final clusters** (especially in ambiguous cases)
- **Potentially worse solutions** if centroids start in bad positions

Solution?

- A common fix is to run **k-means multiple times** with different initializations and pick the best result.
- The **K-Means++ algorithm** improves this by selecting **smart initial centroids** to avoid bad starting points.

---

would-you-like-to-try-running-k-means-with-a-different-kk-or-a-new-set-of-points

Would You Like to Try Running K-Means With a Different kk or a New Set of Points? Would You Like to Try Running K-Means With a Different kk or a New Set of Points?