Notes - k-means

Samuele Nicolò Straccialini

January 2025

Some pseudocodes and notes on k-means clustering algorithms.

1 Initializations

Let k the number of clusters, X an $N \times M$ matrix of N datapoints in M dimensions.

1.1 random

Algorithm 1: random initialization

Data: k, X
Select k points at random

1.2 random-data

Algorithm 2: random-data initialization

Data: k, X
for *datapoint* $x \in X$ **do**
    assign x to one of the k clusters
end

1.3 greedy

Algorithm 3: greedy initialization

Data: k, X
choose μ_1 randomly from X
for $i = 2, \ldots, k$ **do**
    for *datapoint* $x \in X$ **do**
        $D(x) = \min_j \|x - \mu_j\|^2$ // Squared distance to closest centroid
    end
    $\mu_i = \arg\max D(x)$ // Select point with max distance
end

1.4 k-means++

Algorithm 4: k-means++ initialization

Data: k, X
choose μ_1 randomly from X
for $i = 2, \ldots, k$ **do**
  for *datapoint* $x \in X$ **do**
    $D(x) = \min_j \|x - \mu_j\|^2$ // Squared distance to closest centroid
  end
  $P(x) = \frac{D(x)}{\sum_{x' \in X} D(x)}$
  Select μ_i based on the probability distribution $P(x)$
end

2 Clustering

2.1 lloyd

Algorithm 5: lloyd

Data: k, X, initial centroids $\{\mu_1, \mu_2, \ldots, \mu_k\}$
repeat
  for *datapoint* $x \in X$ **do**
    $\mu(x) = \arg\min_j \|x - \mu_j\|^2$ // assign x to the closest centroid
  end
  for *cluster* $j = 1, \ldots, k$ **do**
    C_j = set of points in cluster j
    $\mu_j = \frac{1}{|C_j|} \sum_{x \in C_j} x$ // update centroid to the mean of its points
  end
until *convergence*;
return *final centroids* $\{\mu_1, \ldots, \mu_k\}$ *and clusters* $\{C_1, \ldots, C_k\}$

2.2 hartigan

Algorithm 6: hartigan

Data: k, X, initial centroids $\{\mu_1, \mu_2, \ldots, \mu_k\}$
repeat
    for *datapoint* $x_i \in X$ **do**
        C_d = cluster to which x_i belongs
        for $j = 1, \ldots, k$ *with* $j \neq d$ **do**
            $\Delta\mathrm{cost}_{d \to j} = \frac{|C_j|}{|C_j|+1} \|x_i - \mu_j\|^2 - \frac{|C_d|}{|C_d|-1} \|x_i - \mu_d\|^2$
        end
        if $\min_j \Delta cost_{d \to j} < 0$ **then**
            $s = \arg\min_j \Delta\mathrm{cost}_{d \to j}$
            reassign x_i to cluster C_s
            update μ_d and μ_s
        end
    end
until *no point is reassigned*;
return *final centroids* $\{\mu_1, \ldots, \mu_k\}$ *and clusters* $\{C_1, \ldots, C_k\}$

2.3 safe-hartigan

Algorithm 7: safe-hartigan

Data: k, X, initial centroids $\{\mu_1, \mu_2, \ldots, \mu_k\}$

repeat

  initialize empty *candidates_list*

  1. Find candidates

  for *datapoint* $x_i \in X$ **do**

    C_d = cluster to which x_i belongs

    for $j = 1, \ldots, k$ *with* $j \neq d$ **do**

      $\Delta\text{cost}_{d \to j} = \frac{|C_j|}{|C_j|+1} \left\| x_i - \mu_j \right\|^2 - \frac{|C_d|}{|C_d|-1} \left\| x_i - \mu_d \right\|^2$

    end

    if $\min_j \Delta cost_{d \to j} < 0$ **then**

      $s = \arg\min_j \Delta\text{cost}_{d \to j}$

      add x_i to *candidates_list* together with d, s and $\Delta\text{cost}_{d \to s}$

    end

  end

  2. Safe Mode

  sort *candidates_list* in order of increasing Δcost

  for $x^{(i)}$ *in candidates_list* **do**

    if $C_{d^{(i)}}$ *and* $C_{s^{(i)}}$ *are unchanged* **then**

      reassign $x^{(i)}$ to cluster $C_{s^{(i)}}$ // accept at most one edit per cluster

    end

  end

  3. End Iteration

  update each centroid to the mean of points assigned to it

until *candidates_list is empty*;

return *final centroids* $\{\mu_1, \ldots, \mu_k\}$ *and clusters* $\{C_1, \ldots, C_k\}$

2.4 extended-hartigan

Algorithm 8: extended-hartigan

```
Data: k, X, initial centroids {μ_1, μ_2, ..., μ_k}
repeat
    initialize empty candidates_list
    1. Find candidates
    for datapoint x_i ∈ X do
        C_d = cluster to which x_i belongs
        for j = 1, ..., k with j ≠ d do
            Δcost_{d→j} = |C_j|/(|C_j|+1) ||x_i − μ_j||^2 − |C_d|/(|C_d|−1) ||x_i − μ_d||^2
        end
        if min_j Δcost_{d→j} < 0 then
            s = arg min_j Δcost_{d→j}
            add x_i to candidates_list together with d, s and Δcost_{d→s}
        end
    end
    2. Unsafe Mode
    for x^(i) in candidates_list do
        reassign x^(i) to cluster C_{s^(i)}                 // accept all candidates
    end
    if total cluster cost is decreased then
        accept all reassignments
        proceed to (4.) and start next iteration
    else
        rollback to original assignment
        proceed to safe mode (3.)
    end
    3. Safe Mode
    sort candidates_list in order of increasing Δcost
    for x^(i) in candidates_list do
        if C_{d^(i)} and C_{s^(i)} are unchanged then
            reassign x^(i) to cluster C_{s^(i)}   // accept at most one edit per cluster
        end
    end
    4. End Iteration
    update each centroid to the mean of points assigned to it
until candidates_list is empty;
return final centroids {μ_1, ..., μ_k} and clusters {C_1, ..., C_k}
```

where the cost update is

$$\Delta\text{cost}_{d\to j} = \frac{|C_j|}{|C_j|+1}\left\|x_i - \mu_j\right\|^2 - \frac{|C_d|}{|C_d|-1}\left\|x_i - \mu_d\right\|^2$$

2.5 binary-hartigan

Algorithm 9: binary-hartigan

Data: k, X, initial centroids $\{\mu_1, \mu_2, \ldots, \mu_k\}$

Function *accept_candidates(candidates)* **is**
    for $x^{(i)}$ *in candidates* **do**
        reassign $x^{(i)}$ to cluster $C_{s^{(i)}}$ // accept all candidates
    end
    return *new_cluster_cost*
end

repeat
    initialize empty *candidates_list*

    1. Find candidates
    for *datapoint* $x_i \in X$ **do**
        C_d = cluster to which x_i belongs
        for $j = 1, \ldots, k$ *with* $j \neq d$ **do**
            $\Delta\text{cost}_{d\to j} = \frac{|C_j|}{|C_j|+1} \|x_i - \mu_j\|^2 - \frac{|C_d|}{|C_d|-1} \|x_i - \mu_d\|^2$
        end
        if $\min_j \Delta cost_{d\to j} < 0$ **then**
            $s = \arg\min_j \Delta\text{cost}_{d\to j}$
            add x_i to *candidates_list* together with d, s and $\Delta\text{cost}_{d\to s}$
        end
    end

    2. Unsafe Mode
    new_cluster_cost = *accept_candidates*(*candidates_list*)
    if *total cluster cost is decreased* **then**
        accept all reassignments
        proceed to (**4.**) and start next iteration
    else
        rollback to original assignment
        proceed to binary mode (**3.**)
    end

    3. Binary Mode
    *part*1, *part*2 = split *candidates_list* in half
    new_cluster_cost = *accept_candidates*(*part*1)
    if *total cluster cost is decreased* **then**
        accept *part*1 reassignments
        proceed to (**4.**) and start next iteration)
    else
        new_cluster_cost = *accept_candidates*(*part*2)
        if *total cluster cost is decreased* **then**
            accept *part*2 reassignments
            proceed to (**4.**) and start next iteration)
        else
            execute (**3.**) on *part*1 and on *part*2
        end
    end

    4. End Iteration
    update each centroid to the mean of points assigned to it
until *candidates_list is empty*;
return *final centroids* $\{\mu_1, \ldots, \mu_k\}$ *and clusters* $\{C_1, \ldots, C_k\}$