DraftTesiMazza

February 2025

1 K-Medoids

K-medoids is a clustering algorithm that selects actual data points as cluster centers (medoids) instead of computing centroids like k-means. It minimizes the total dissimilarity (e.g., distance or travel time) between points and their assigned medoid, making it more robust to outliers and suitable for arbitrary distance metrics.

K-medoids is more suitable than k-means when working with distance metrics like travel times, which may not be symmetric, since avoids the need for computing a ”mean” point in a non-Euclidean space.

1.1 MIP Formulation

The formulation is for each day d and session s. This dependency is avoided in the notation.

1.2 Sets

- $\mathcal{P} := \{1, ..., P\}$: set of patients of day d and session s;
- $\mathcal{R} := \{1, ..., R\}$: set of requests of day d and session s;
- $\mathcal{O} := \{1, ..., O\}$: set of operators available in day d session s

1.3 Parameters

- $\tau_{ij} \geq 1$ = time in minutes between patient i and j;
- $K \in \mathbb{N}, K > 1$ number of clusters;
- $w_p \in [0, 1] = \frac{|\{i \ | \ p_i = p \quad \forall i \in \mathcal{R}\}|}{R}$

1.4 Variables

- $x_{ij} \in \{0, 1\} = \mathbb{1}\{\text{patient i is associated to medoid j}\}$;
- $y_i \in \{0, 1\} = \mathbb{1}\{\text{patient i is a medoid}\}$;

1.5 Formulation

$$\min_{x,y} \sum_i \sum_j \tau_{ij} x_{ij}$$

$$\sum_j x_{ij} = 1 \qquad \forall i \in \mathcal{P} \tag{1}$$

$$\sum_i y_i = K \tag{2}$$

$$x_{ij} \leq y_j \qquad \forall i, j \in \mathcal{P} \tag{3}$$

2 Greedy Routing and Scheduling Algorithm (GRS)

Once we have some balanced clusters of requests, we have to schedule all of them to some operators.

To each request i is associated a feasible temporal window $[\alpha_i, \beta_i]$ when the request must be served, its duration t_i in minutes, and the relative patient $p_i \in \mathcal{P}$. The goal of this method is to assign each request to an operator in a feasible manner. We have to guarantee that

- The total work time assigned to each operator is compatible with its time shift.
- All sets of requests assigned to each operator must be feasible for each operator, i.e., there must exist a feasible sequence $(i_o^1, i_o^2, ..., i_o^n)$ with $n \leq N$ such that operator o can serve all of them in their time windows, $\forall o \in \mathcal{O}$.
- We want to create these feasible sequence trying to minimize also the routing cost in a greedy manner.

2.1 Variables

- $p_o \in \mathcal{P} \cup \{h\}$ $\forall o \in \mathcal{O}$: the current patient of operator o. h is a dummy node that represents the depot. We can think to h as the starting point of each operator.
- $e_o \in \mathbb{R}, e_o > 0$: is the first moment available for operator $o \in \mathcal{O}$ to serve a request.
- $w_o \in \mathbb{R}, w_o > 0$: is the amount of time worked by operator $o \in \mathcal{O}$
- $L_o = \{(i, \max\{e_o + \tau_{p_o p_i}, \alpha_i\})\}$: is the list of requests assigned to operator o
- $h_o \in \mathbb{R}, h_o > 0$: is the remaining time till the end of the session for operator $o \in \mathcal{O}$

2.2 Pseudocode

Algorithm 1: GRS - Time

Input : Set of requests $\mathcal{R}$, set of operators $\mathcal{O}$
Output: A set of feasible sequences of requests for each operator

```
// Sort requests by their min time of begin
```
Sort $\mathcal{R}$ with respect to α_i

```
// Initialize variables
```
$\tau_{hp_i} \leftarrow 0 \quad \forall i \in \mathcal{R}$
$h_o \leftarrow 300 \quad \forall o \in \mathcal{O}$ `// 5-hours shift`
$p_o \leftarrow h \quad \forall o \in \mathcal{O}$
$e_o \leftarrow 420 \quad \forall o \in \mathcal{O}$ `// Morning shift begins at 7`
$w_o \leftarrow 0 \quad \forall o \in \mathcal{O}$
$L_o \leftarrow \emptyset \quad \forall o \in \mathcal{O}$

foreach $i \in \mathcal{R}$ **do**
  Assign i to $o \in \arg\min_o\{\tau_{p_o p_i}\ s.t\ e_o + \tau_{p_o p_i} \leq \beta_i$ **and** $t_i \leq h_o\}$
  $L_o \leftarrow L_o \cup (i, \max\{e_o + \tau_{p_o p_i}, \alpha_i\})$
  $w_o \leftarrow w_o + t_i + \tau_{p_o p_i}$
  $e_o \leftarrow \max\{e_o + \tau_{p_o p_i}, \alpha_i\} + t_i$
  $h_o \leftarrow 720 - e_o$ `// 720 (12 am) for morning shift, 1290 (21.30) for afternoon shift`
  $p_o \leftarrow p_i$

Algorithm 2: GRS - SaturamiPlease

Input : Set of requests $\mathcal{R}$, set of operators $\mathcal{O}$
Output: A set of feasible sequences of requests for each operator

```
// Sort requests by their min time of begin
```
Sort $\mathcal{R}$ with respect to α_i

```
// Initialize variables
```
$\tau_{hp_i} \leftarrow 0 \quad \forall i \in \mathcal{R}$
$h_o \leftarrow 300 \quad \forall o \in \mathcal{O}$ `// 5-hours shift`
$p_o \leftarrow h \quad \forall o \in \mathcal{O}$
$e_o \leftarrow 420 \quad \forall o \in \mathcal{O}$ `// Morning shift begins at 7`
$w_o \leftarrow 0 \quad \forall o \in \mathcal{O}$
$L_o \leftarrow \emptyset \quad \forall o \in \mathcal{O}$

foreach $i \in \mathcal{R}$ **do**
- Assign i to $o \in \arg\min_o\{h_o \ s.t \ e_o + \tau_{p_o p_i} \leq \beta_i$ **and** $t_i \leq h_o\}$
- $L_o \leftarrow L_o \cup (i, \max\{e_o + \tau_{p_o p_i}, \alpha_i\})$
- $w_o \leftarrow w_o + t_i + \tau_{p_o p_i}$
- $e_o \leftarrow \max\{e_o + \tau_{p_o p_i}, \alpha_i\} + t_i$
- $h_o \leftarrow 720 - e_o$ `// 720 (12 am) for morning shift, 1290 (21.30) for afternoon shift`
- $p_o \leftarrow p_i$

Algorithm 3: GRS - LasciamiInPace

Input : Set of requests $\mathcal{R}$, set of operators $\mathcal{O}$
Output: A set of feasible sequences of requests for each operator

```
// Sort requests by their min time of begin
```
Sort $\mathcal{R}$ with respect to α_i

```
// Initialize variables
```
$\tau_{hp_i} \leftarrow 0 \quad \forall i \in \mathcal{R}$
$h_o \leftarrow 300 \quad \forall o \in \mathcal{O}$ `// 5-hours shift`
$p_o \leftarrow h \quad \forall o \in \mathcal{O}$
$e_o \leftarrow 420 \quad \forall o \in \mathcal{O}$ `// Morning shift begins at 7`
$w_o \leftarrow 0 \quad \forall o \in \mathcal{O}$
$L_o \leftarrow \emptyset \quad \forall o \in \mathcal{O}$

foreach $i \in \mathcal{R}$ **do**
- Assign i to $o \in \arg\max_o\{h_o \ s.t \ e_o + \tau_{p_o p_i} \leq \beta_i$ **and** $t_i \leq h_o\}$
- $L_o \leftarrow L_o \cup (i, \max\{e_o + \tau_{p_o p_i}, \alpha_i\})$
- $w_o \leftarrow w_o + t_i + \tau_{p_o p_i}$
- $e_o \leftarrow \max\{e_o + \tau_{p_o p_i}, \alpha_i\} + t_i$
- $h_o \leftarrow 720 - e_o$ `// 720 (12 am) for morning shift, 1290 (21.30) for afternoon shift`
- $p_o \leftarrow p_i$

Algorithm 4: GRS - TradeOff

Input : Set of requests $\mathcal{R}$, set of operators $\mathcal{O}$
Output: A set of feasible sequences of requests for each operator

```
// Sort requests by their min time of begin
```
Sort $\mathcal{R}$ with respect to α_i

```
// Initialize variables
```
$\tau_{hp_i} \leftarrow 0 \quad \forall i \in \mathcal{R}$
$h_o \leftarrow 300 \quad \forall o \in \mathcal{O}$ `// 5-hours shift`
$p_o \leftarrow h \quad \forall o \in \mathcal{O}$
$e_o \leftarrow 420 \quad \forall o \in \mathcal{O}$ `// Morning shift begins at 7`
$w_o \leftarrow 0 \quad \forall o \in \mathcal{O}$
$L_o \leftarrow \emptyset \quad \forall o \in \mathcal{O}$

foreach $i \in \mathcal{R}$ **do**
- Assign i to $o \in \arg\min_o\{w_o + \tau_{p_o p_i} + t_i \; s.t \; e_o + \tau_{p_o p_i} \leq \beta_i$ **and** $t_i \leq h_o\}$
- $L_o \leftarrow L_o \cup (i, \max\{e_o + \tau_{p_o p_i}, \alpha_i\})$
- $w_o \leftarrow w_o + t_i + \tau_{p_o p_i}$
- $e_o \leftarrow \max\{e_o + \tau_{p_o p_i}, \alpha_i\} + t_i$
- $h_o \leftarrow 720 - e_o$ `// 720 (12 am) for morning shift, 1290 (21.30) for afternoon shift`
- $p_o \leftarrow p_i$

For the afternoon shift, we have $t_o = 960$ (16).

3 Method Overview

Algorithm 5: METHOD OVERVIEW

Input : Set of requests $\mathcal{R}$, set of operators $\mathcal{O}$, set of patients $\mathcal{P}$
Output: Heuristic scheduling and routing for the week

foreach $d \in \mathcal{D}$ **do**
  foreach $s \in \{m, a\}$ **do**
    $\mathcal{R}_{ds} \leftarrow \{i, i \in \mathcal{R}, d_i = d, s_i = s\}$
    $\mathcal{P}_{ds} \leftarrow \{p, p \in \mathcal{P} \mid \exists i \in \mathcal{R}_{ds}, p_i = p\}$
    $w_{pds} \leftarrow \frac{|i \in \mathcal{R}_{ds}, p_i = p|}{|\mathcal{R}_{ds}|} \quad \forall p \in \mathcal{P}_{ds}$
    Plot $(\{(lat_p, lon_p), \forall p \in \mathcal{P}_{ds}\})$
    foreach $k = 1, ..., 20$ **do**
      $Clusters \leftarrow k - medoid(w_{pds}, \mathcal{P}_{ds}, \tau)$
    `// Select the best clusters configuration` C^* `according to some measures`
    `// Define the partition`
    $\{\mathcal{O}_{dsc} \mid \bigcup_c \mathcal{O}_{dsc} = \mathcal{O}_{ds}, \mathcal{O}_{dsc_i} \cap \mathcal{O}_{dsc_j} = \emptyset \ \forall \ c_i \neq c_j, \ c_i, c_j \in C^*\}$
    `// according to some heuristics`
    foreach $c \in C^*$ **do**
      $GRS(c, \mathcal{O}_{dsc})$
      `// Print statistics of the cluster` c `of day` d `in session` s

`// Print all the statistics with total routing costs and working shifts`
`// Create a directed graph with the scheduling of the requests for each operator/day/session`