Clustering techniques to optimize railway daily path utilization for non-daily trains

Karim Shahbaz, IIT Bombay

Co-authors:

Mohit Agarwala, Madhu N. Belur, S. P. Singh, S. V. Ramisetty, S. R. Duragkar, Narayan Rangaraj, Merajus Salekin, Raja Gopalakrishnan

Affiliations:

Industrial Engineering and Operations Research & Electrical Engineering department, IIT Bombay, Centre for Railway Information Systems, New Delhi

Table of Contents

- Objective
  - Problem statement
- Data and its preprocessing
  - GQD Data
  - Preprocessing
- Methods
  - K-Means
  - DBSCAN
  - HAC
- Results
- Conclusion & future work

Objective

Definition

Dailyzing: grouping of trains into one daily-path.

Objective:

Group non-daily trains in clusters (i.e., achieve Dailyzing) for:

- Efficient track utilization.
- Faster and efficient time-tabling.

Problem Formulation

Problem statement:

Clustering/grouping of non-daily trains which occupy the same space (station/block section) at the same time on different days.

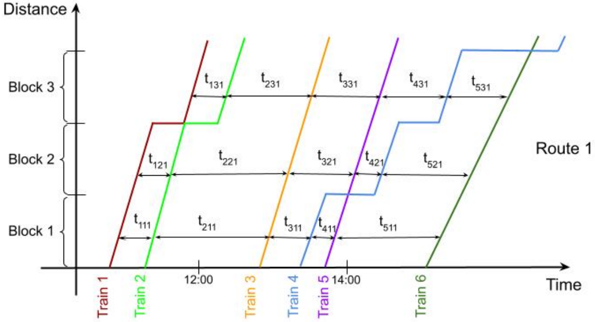
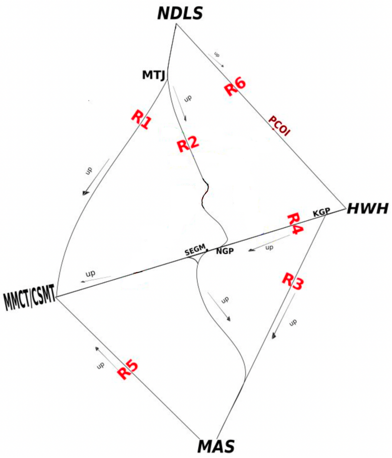


Figure: Distance Vs Time (Train path)

Table of Contents

- Objective
  - Problem statement
- Data and its preprocessing
  - GQD Data
  - Preprocessing
- Methods
  - K-Means
  - DBSCAN
  - HAC
- Results
- Conclusion & future work

Golden Quadilateral & its Diagonal (GQD) Data: Routes Map SD



HWH-Howrah
MAS-Madras
MMCT/CSMT-Mumbai
NDLS-New Delhi

Data preprocessing

Route-wise division of all trains.

Data preprocessing

Route-wise division of all trains.

Removal of **Daily trains**.

Data preprocessing

Route-wise division of all trains.

Removal of **Daily trains**.

Modulo 86400 operation.

Data preprocessing

Route-wise division of all trains.

Removal of **Daily trains**.

Modulo 86400 operation.

Removal of single touch trains in a given route.

Data preprocessing

Route-wise division of all trains.

Removal of **Daily trains**.

Modulo 86400 operation.

Removal of single touch trains in a given route.

Removal of geo-loops: train leaves the route and returns at the same station.

Data preprocessing

Route-wise division of all trains.

Removal of **Daily trains**.

Modulo 86400 operation.

Removal of single touch trains in a given route.

Removal of geo-loops: train leaves the route and returns at the same station.

Legs separation: train jumping routes.

Data preprocessing

Route-wise division of all trains.

Removal of **Daily trains**.

Modulo 86400 operation.

Removal of single touch trains in a given route.

Removal of geo-loops: train leaves the route and returns at the same station.

Legs separation: train jumping routes.

Up/down block section classification.

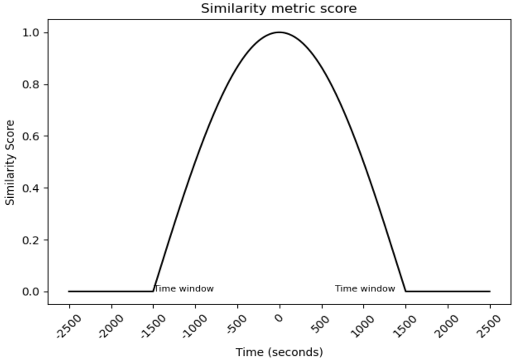
Table of Contents

- Objective
  - Problem statement
- Data and its preprocessing
  - GQD Data
  - Preprocessing
- Methods
  - K-Means
  - DBSCAN
  - HAC
- Results
- Conclusion & future work

Methods: Distance metric

Distance metric:

Similarity/dissimilarity matrix generation averaged over all block sections: distance/closeness metric.



$$\text{Similarity Score} = \begin{cases} \cos\left(\dfrac{\pi(T_i - T_j)}{2 \times \text{Time\_Window}}\right), & \text{if Time\_Window} > |T_i - T_j|, \\ 0, & \text{otherwise.} \end{cases}$$

where,
T_i, T_j: times of arrival for the Train i and Train j respectively at a given block-section,
Time_Window: allowed time within which two trains are considered to be similar.

Methods: Clustering techniques used

K-Means, [1957]:

- Centroid based clustering algorithm: partitions data into the clusters based on closeness to cluster centroids.
- Requires a pre-defined number of clusters information.
- Not always converges to global minima.

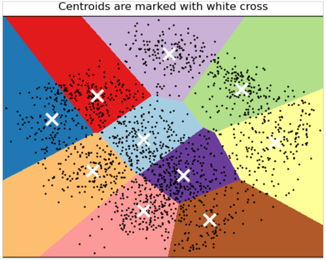


Figure: Clustering with K-means (k=10)[1]

[1] Photo courtesy: https://scikit-learn.org/

Methods: Clustering techniques contd.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise), [1996]:

- Density-based clustering algorithm: closely packed data are grouped.
- Hyper-parameters: Epsilon (radius of the neighborhood to consider) and minPoints (minimum number of points to consider as dense).
- Works with arbitrarily shaped clusters, robust to outliers.
- Not purely deterministic; not good in large density variation data.

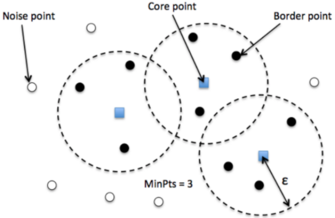


Figure: DBSCAN algorithm overview [2]

[2] Photo courtesy: https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html

Methods: Clustering techniques contd.

HAC (Hierarchical Agglomerated Clustering), [1967]:

- Connectivity based clustering algorithm
- Builds hierarchy of cluster: starts with each data point as a cluster, then merges and moves up the hierarchy.
- Hyperparameter: linkage method and affinity (distance metric) method.
- Linkage influences the shape of the cluster (For example, a single linkage method leads to a spherical shape).

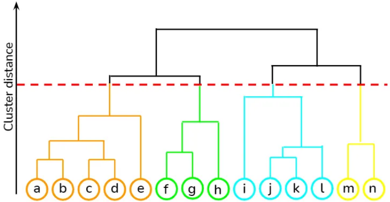


Figure: HAC: Dendogram [a]

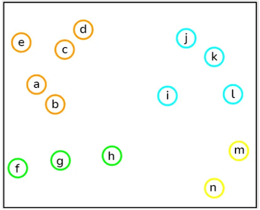


Figure: Data points

[a] Photo courtesy: https://towardsdatascience.com/hierarchical-clustering-

Method discussion : HAC hyper-parameter Tuning

Choosing the best linkage and affinity for Agglomerative clustering is done based on the number of clusters formed and the cluster size.

Route	Total unique trains	Total clustered trains	Number of clusters	Maximum number of trains in cluster	Total number of conflicting cluster
1	257	172	66	6	3
2	256	151	57	6	4
3	166	105	32	7	2
4	194	132	47	7	1
5	123	65	25	5	3
6	251	158	56	7	3

Hierarchical Clustering (Ward linkage and Variable time window).

Route	Total unique trains	Total clustered trains	Number of clusters	Maximum number of trains in cluster	Total number of conflicting cluster
1	257	187	63	6	5
2	256	151	55	6	4
3	166	98	33	6	3
4	194	129	41	7	0
5	123	79	27	7	3
6	251	155	57	6	1

Hierarchical Clustering (average linkage and Variable time window based on station distance)

Hyper-parameter tuning contd...

Route	Total unique trains	Total clustered trains	Number of clusters	Maximum number of trains in cluster	Total number of conflicting cluster
1	257	208	67	7	7
2	256	184	60	7	10
3	166	103	34	6	2
4	194	138	48	7	2
5	123	71	26	5	3
6	251	162	58	7	2

Hierarchical Clustering (average linkage and Optimized fixed time window)

Similarly, a "time window" could be an optimal "fixed" time window or "variable" time window based on block-section size.

Table of Contents

- Objective
  - Problem statement
- Data and its preprocessing
  - GQD Data
  - Preprocessing
- Methods
  - K-Means
  - DBSCAN
  - HAC
- Results
- Conclusion & future work

Results: HAC for all routes CEx Ccl

Route number	Total unique trains	Total conflict free clustered trains	Total number of clusters
1	257	208	77
2	256	178	64
3	166	123	43
4	194	146	54
5	123	73	30
6	251	198	57

Table: Hierarchical Agglomerative Clustering for all routes

Results: Comparative study of clustering techniques

		HAC		DBSCAN		K-means	
Route number	Total unique trains	Number of conflict free clustered trains	Time taken (sec)	Number of conflict free clustered trains	Time taken (sec)	Number of conflict free clustered trains	Time taken (sec)
1	257	208	**2.15**	173	13.70	198	312.70
2	256	178	**3.19**	134	19.03	179	458.18
3	166	123	**1.37**	85	5.79	115	142.20
4	194	146	**2.52**	132	5.38	150	224.95
5	123	73	**0.85**	57	1.66	78	82.15
6	251	198	**3.74**	167	16.12	190	313.70

Table: Comparative study of different clustering techniques and their execution time

Results and Discussion

The following table is generated using Indian Railway Catering and Tourism Corporation (IRCTC) website[3]. Train clusters have very similar and complementing trains.

Train No.	12882	12888	12896	15643	22804	22836
Source-Dest.	PURI-KOL	PURI-KOL	PURI-KOL	PURI-HWH	SBP-SHM	PURI-SHM
Days of the week	Monday, Wed.	Sunday	Thursday	Saturday	Friday	Tuesday
Start time	22 : 05	22 : 05	22 : 05	22 : 15	19 : 40	22 : 05
End Time	06 : 50	06 : 50	06 : 50	07 : 05	06 : 50	06 : 50

Train No.	12218	12484	12918	22660
Source-Dest.	CDG-KCVL	ASR-KCVL	NZM-ADI	YNRK-KCVL
Days of the week	Wed.,Friday	Sunday	Saturday	Monday
Start time	09 : 30	05 : 55	13 : 25	06 : 15
End Time	12 : 30	12 : 30	03 : 20	12 : 30

Train No.	12247	12907	12909
Source-Dest.	BDTS-NZM	BDTS-NZM	BDTS-NZM
Days of the week	Friday	Wed., Sunday	Tue, Thur, Saturday
Start time	17 : 30	17 : 30	17 : 30
End Time	10 : 15	10 : 15	10 : 15

Clustered trains comparison with actual running trains data from IRCTC website.

[3] https://www.irctc.co.in/nget/train-search

Table of Contents

- Objective
  - Problem statement
- Data and its preprocessing
  - GQD Data
  - Preprocessing
- Methods
  - K-Means
  - DBSCAN
  - HAC
- Results
- Conclusion & future work

Conclusion

Clusters generated based on the Hierarchical Agglomerative Clustering (HAC) method match with actual running trains data.

The clusters generated:

- complement each other very well
- very few conflicting clusters

The time required by HAC algorithm to generate clusters is within 3-4 seconds, faster than other techniques like K-means, DBSCAN clustering.

Future Work

Faster timetabling procedure: grouped non-daily trains can be represented by a daily train, scheduling which automatically schedules others in the group.

Possibility of new trains: clusters with fewer than seven members can accommodate more non-daily trains. Hence, availability to introduce new trains.

Efficient clustering will help in determining under-utilized resources.

Suggestion for better timetabling, i.e., rescheduling some trains could lead to better compaction and efficient resource utilization.

Clustering techniques to optimize railway daily path utilization for non-daily trains

Thanks on behalf of IIT Bombay & CRIS

Any questions?

For more information contact us at the following:
karimshahee@ee.iitb.ac.in, belur@ee.iitb.ac.in

GQD Data illustration B2M

TRAIN	WEEKDAYS	STATION	ARVL	BLCKSCTN	DAY	Direction
11111	0,0,1,0,0,0,0	MTJ	61500	MTJ-BTSR	1	down
11111	0,0,1,0,0,0,0	BTSR	61860	BTSR-VRBD	1	down
11111	0,0,1,0,0,0,0	VRBD	62160	VRBD-AJH	1	down
11111	0,0,1,0,0,0,0	AJH	62460	AJH-CHJ	1	down
11111	0,0,1,0,0,0,0	CHJ	62760	CHJ-KSV	1	down

GQD Sample dataset

This is a sample from the GQD dataset where:

- TRAIN: denotes train number
- WEEKDAYS: '1' denotes that train runs on that day of the week
- STATION: denotes the station through which the train passes
- ARVL: arrival time of the train at the given station
- BLCKSCTN: is the block section
- DAY: day of the journey of the train when it commenced from source
- Direction: is the direction of the train depending on the route

Results: Clusters examples in route-1 B2R

Cluster examples with 7 members
{12946, 12960, 19262, 19332, 19578, 22908, 22966}
{12959, 12979, 22933, 22935, 22963, 22989, 22993}

Cluster examples with 6 members
{16333, 16335, 16337, 19260, 22654, 22656}
{12217, 12483, 12917, 19807, 19813, 22659}

Cluster examples with 5 members
{19261, 19423, 19577, 22474, 22951}
{12218, 12484, 12918, 22660, 22964}

Results: Conflicting clusters in route-1 B2R

* Conflicting cluster: [19308, 22168, 22404]
  - {22168 22404 } are conflicting at {'PWL-RDE'} at {86100} on {'1,0,0,0,0,0,0'}
  - Conflict at 1 block-section out of 18 block-sections

* Conflicting cluster: [12283, 12643, 12645]
  - {12283 12643} are conflicting on {'0,0,0,1,0,0,0'}
  - Conflict at 17 block-section out of 17 block-sections.

Total conflicting cluster: 3 Total conflicting instance: 3