



UNIVERSITY
OF TRIESTE

Algorithmic Design

University of Trieste – a.a. 2026/2027

Prof. Antonio Emanuele Cinà

`antonio.cina@units.it`

Department of Mathematics, Informatics and Geosciences

University of Trieste

a.a. 2026/2027

Module B – Data Mining through Algorithmic Methods

Chapter 7:

From data to an algorithmic mining question

Start with a practical question

A transport analyst has a table of urban journeys.

What can we do with these records?

Retrieve trips, compare observed routes, find repeated sequences, or identify groups of stations.

The table may contain a journey ID, an anonymised user ID, origin, destination, departure and arrival times, duration, and transport mode.

Module A taught us to analyse an algorithm. Here we first decide **what task the data can support**.

One table, several possible tasks

Journey ID	User	Origin	Destination	Duration (min)
J1	U1	A	B	8
J2	U2	A	B	12
J3	U1	B	C	6
J4	U3	A	C	25
J5	U2	C	A	11

Illustrative data. These five journeys lie in one declared time window.

Retrieving trips by U1, building a route graph, and counting frequent transitions are different tasks on the same rows.

What is missing?

Without timestamps, these rows alone do not establish a chronological sequence of trips.

What counts as data mining?

Data mining

Data mining is the algorithmic process of extracting candidate patterns, relations, summaries, or structures from data collections, subject to explicit computational and validation criteria.

A repeated route, a group of similar users, or a dense group of locations is a **candidate finding** in a chosen representation.

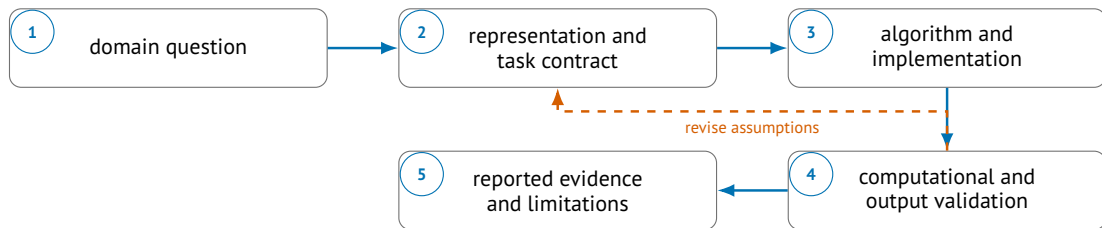
It is not automatically an explanation, a causal claim, or a universal fact about travel behaviour.

Choose the application before the algorithm

Application	Concrete question
Retrieval	Which journeys belong to U1 or to a given time interval?
Route optimisation	Which observed route has minimum total travel time?
Pattern discovery	Which stations or modes occur together unusually often?
Sequence search	Does a given succession of locations recur?
Structure discovery	Which stations form groups under a declared relation?

Retrieval and shortest paths can be enabling services inside a larger mining workflow.

From a question to evidence



A practical checkpoint

Before interpreting a route or a station group, verify both the computation and the assumptions used to construct its input.

Six questions to ask before implementation

1. Which representation makes the intended relation explicit?
2. Which assumptions does the algorithm require?
3. What determines the input size and possible output size?
4. Which cost is paid once, and which is paid per query?
5. Is the answer exact, approximate, or probabilistic?
6. Which interpretations does the result support?

Keep these answers beside the implementation and the experiment report.

Turn a request into a task contract

“Find important mobility patterns” does not specify a valid output.

Write this before choosing the method

Domain question; represented input; required output; validity conditions; quality criterion; size parameters; resource limits; unsupported interpretations.

For a route query: find a path from s to t using observed transitions, with minimum total aggregated duration.

Dijkstra's algorithm is a possible *method* under suitable weight assumptions; it is not the problem itself.

The contract in mathematical form

$$\mathcal{P} = (I, S, R, \mu, q).$$

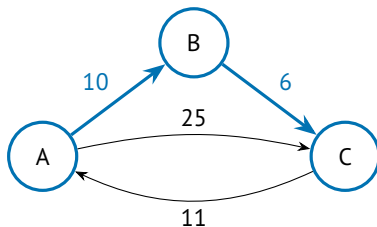
I	Set of valid inputs.
S	Space of possible outputs.
$R \subseteq I \times S$	Which outputs are valid for each input.
$\mu : I \rightarrow \mathbb{N}^r$	Size map: r parameters that may vary independently.
$q : R \rightarrow Q$	Optional quality map: one score or several criteria for comparing valid outputs.

An exact decision problem may not need q . A discovery task often does.

Specify the route that we want to compute

- **Input:** a directed graph $G = (V, E)$ built from validated journeys; stations are vertices, observed direct transitions are edges.
- **Weights:** non-negative median observed duration per transition, within a declared time window.
- **Output:** an $s-t$ path, or a statement that none exists in G .
- **Validity and quality:** follow graph edges; minimise total edge weight among valid paths.
- **Size:** $|V|$ stations and $|E|$ directed transitions.

Build a graph and compare routes



From the illustrative table:

$$w(A, B) = 10.$$

Route $A \rightarrow B \rightarrow C$ costs $10 + 6 = 16$ minutes.

Direct $A \rightarrow C$ costs 25 minutes.

The two-edge route is better under this declared objective.

Interpret the answer precisely

This is a path in an aggregate graph. It does not predict future travel time, include unobserved connections, or establish route safety.

A representation is a contract

Representation contract

A representation contract states what each stored object means, which operations the representation supports, and which invariants must hold for algorithms to interpret it correctly.

For the route graph: each vertex denotes one canonical station; each directed edge denotes an observed transition; all weights use the same duration unit and time window.

Representation determines what is explicit, what is inexpensive, and what is lost.

Three useful views of journey data

View	Stored object	Natural operations
Table	One entity/event per row; declared attributes	Filter by user; group and aggregate; construct features.
Graph	Objects as vertices; declared relations as edges	Reachability, routes, connectivity, network structure.
Transactions	A set or multiset of items per event	Frequent itemsets and association rules.

A graph may discard event order. A transaction preserves co-occurrence, not the order of its items.

When order, geometry or arrival time matters

View	Stored object	Natural operations
Sequence/string	Ordered symbols or events	Exact patterns, repeated substrings, edit-based similarity.
Vectors	Numerical features for each object	Distances, nearest neighbours, geometric grouping.
Stream	Time-ordered arrivals, possibly not stored in full	One-pass summaries, filtering, change detection.

Geometric comparisons require justified coordinates and scales. These representations are not interchangeable.

The same records can answer different questions

Question 1: find an observed route

Use a directed weighted graph. Preserve transition direction and aggregated duration; individual trip details may disappear.

Question 2: find a repeated succession of locations

Use ordered events with reliable timestamps and a declared grouping rule. A set of visited stations would lose the order being queried.

Practical check: write down what your transformation discards before running the algorithm.

Measure the input that the method actually uses

Representation/workload	Independently relevant parameters
Table	N rows, d attributes
Sparse matrix X	Dimensions and $\text{nnz}(X)$ stored nonzeros
Graph	$n = V $ vertices, $m = E $ edges
Query workload	N_q requests
Reported matches or pairs	L output items

A linear scan can be feasible when all-pairs comparison is not. Even a small dataset can be challenging if it is high-dimensional or changes rapidly.

Thank you!



UNIVERSITY
OF TRIESTE