Map-Matching on Big Data: a Distributed and Efficient Algorithm with a Hidden Markov Model

Matteo Francia, Enrico Gallinucci, Federico Vitali

{m.francia,enrico.gallinucci} @ unibo.it

{federico.vitali} @ studio.unibo.it

BUSINESS INTELLIGENCE GROUP

Map-Matching: problem definition

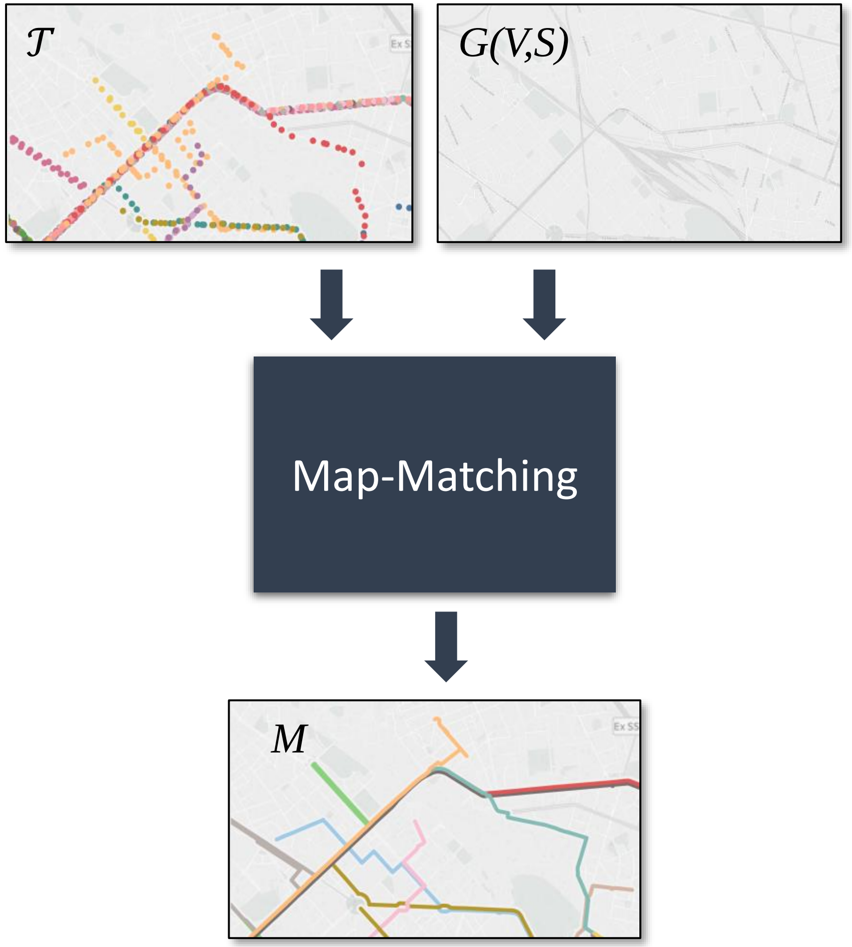
Trajectory $T = (p_0, \ldots, p_t)$: sequence of GPS points ordered by time [1]

Map-Matching: attach points to the route along which an object is moving

- Inputs: trajectory dataset $\mathcal{T}$, road network $G(V, S)$
- Output: matched trajectories M

Issues

- Millions of GPS points
- … with systematic measurement errors
- … from different road networks



Up to now

- Sequential implementations achieve best accuracy
  - Naïve: generate all possible routes to identify the most-likely [2]
  - HMMs model road network topology to achieve higher accuracy [3, 4]
    - Highest accuracy (up to now)

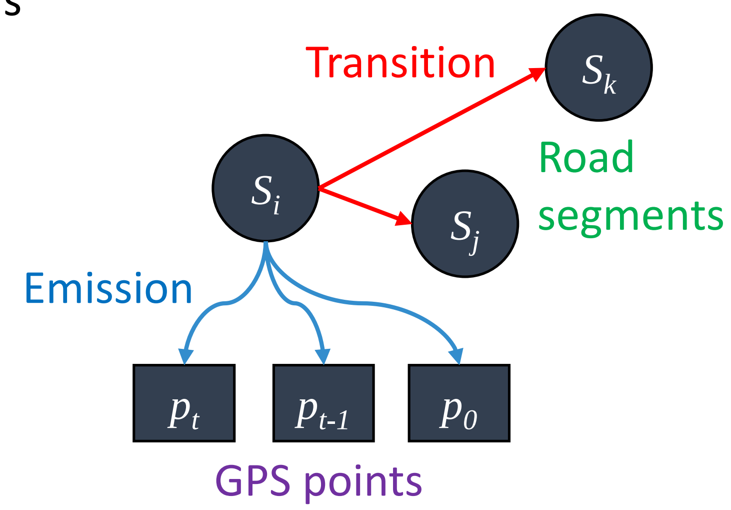
- Distributed implementations
  - Scalable approximations of exact algorithms [5]
  - Indexing structures facilitating the matching process [6, 7]

- There is room to improve accuracy of distributed implementation

Hidden Markov Model (HMM)

HMM describes probability distribution over sequences

- Find most likely sequence of hidden states producing observed symbols
- Markov chain: state transition depends only on current state

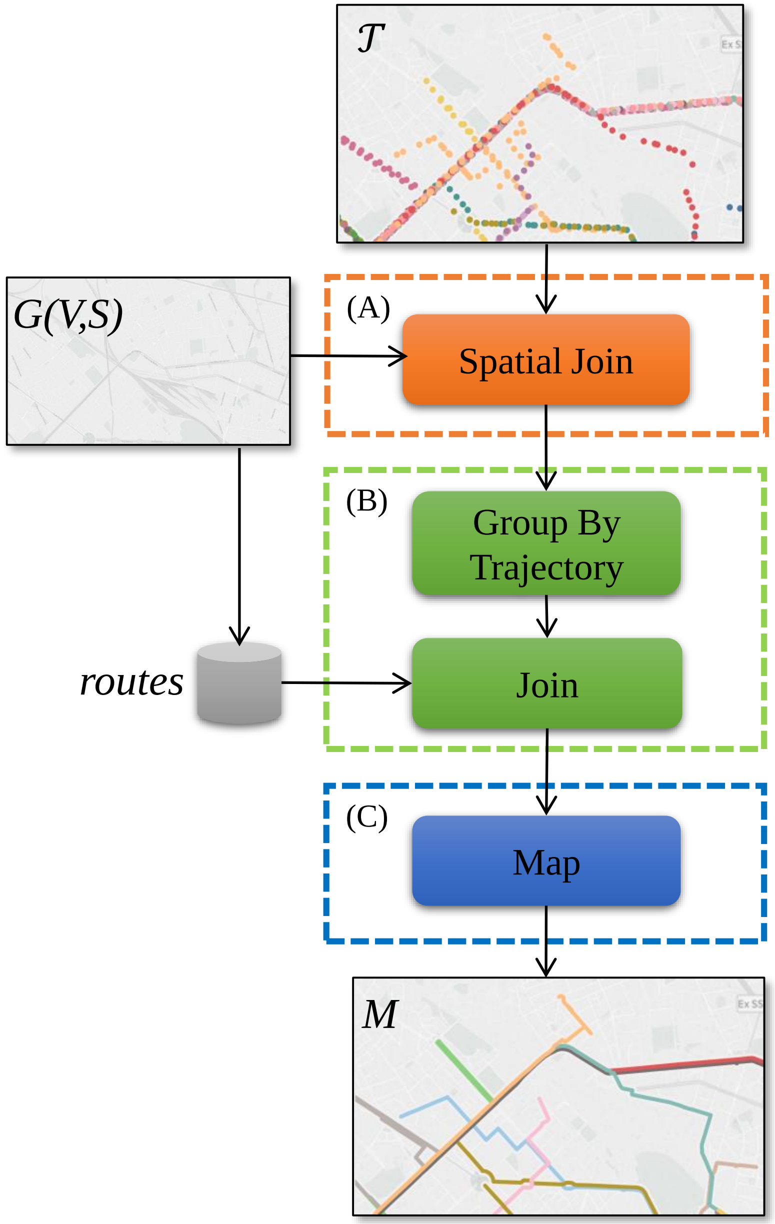


From HMM to Map-Matching of trajectory T

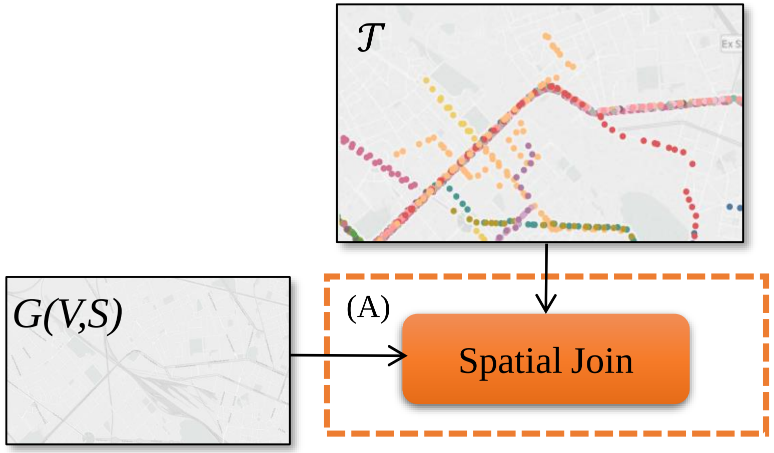
HMM	Map-Matching
Hidden states (not observable)	Road segments (S)
Initial state probability	Equally probable segments
Observable symbols	GPS points ($p_0, \ldots, p_t$)
Emission probability	Point/Segment geometrical relationship
Transition probability	Segment/Segment topological relationships

Our contribution

- Map-Matching on big data
  - Based on HMM (inspired by [4])
  - Extend transition probability to fragmented networks
    - [4] considers only 3 segments
  - Scale up to millions of GPS points
    - Implemented in Spark & GeoSpark
- Given $\mathcal{T}$ and $G(V,S)$, three-step approach
  (A) Emission probability estimation
  (B) Transition probability estimation
  (C) Viterbi algorithm

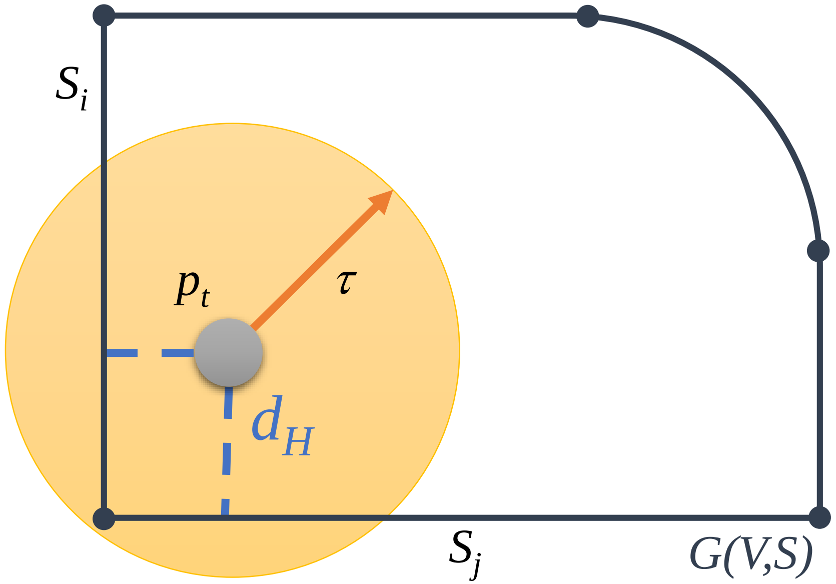


Map-Matching: Step A



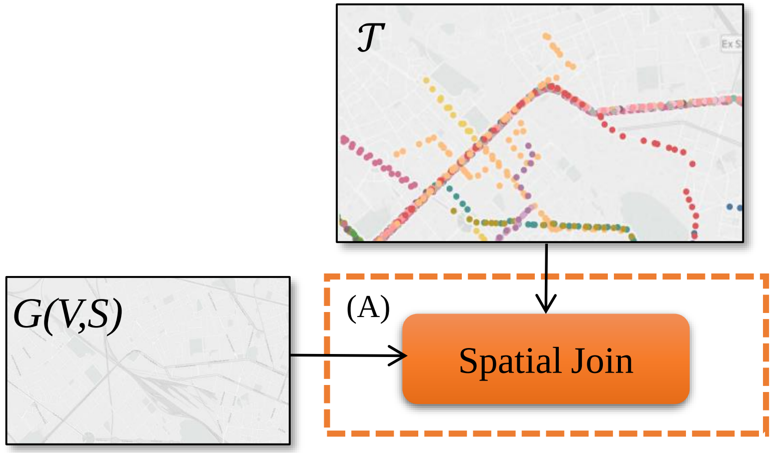
Emission probability estimation

$$P(p_t|X_t=i)=\frac{d_H(p_t,s_i)^{-1}}{\sum_{s_j\in N(p_t,S,\alpha,\tau)} d_H(p_t,prj(p_t,s_j))^{-1}}$$



d_H = Haversine distance

Map-Matching: Step A



Emission probability estimation

$$P(p_t|X_t=i)=\frac{d_H(p_t,s_i)^{-1}}{\sum_{s_j\in N(p_t,S,\alpha,\tau)} d_H(p_t,prj(p_t,s_j))^{-1}}$$

Compute the neighborhood N for each point p_t in $\mathcal{T}$

- Join $\mathcal{T}$ and $G(V,S)$ on element-wise distance

Efficiency: road segments far from p_t have a null emission probability

- τ bounds the neighborhood range
- α bounds the neighbors cardinality

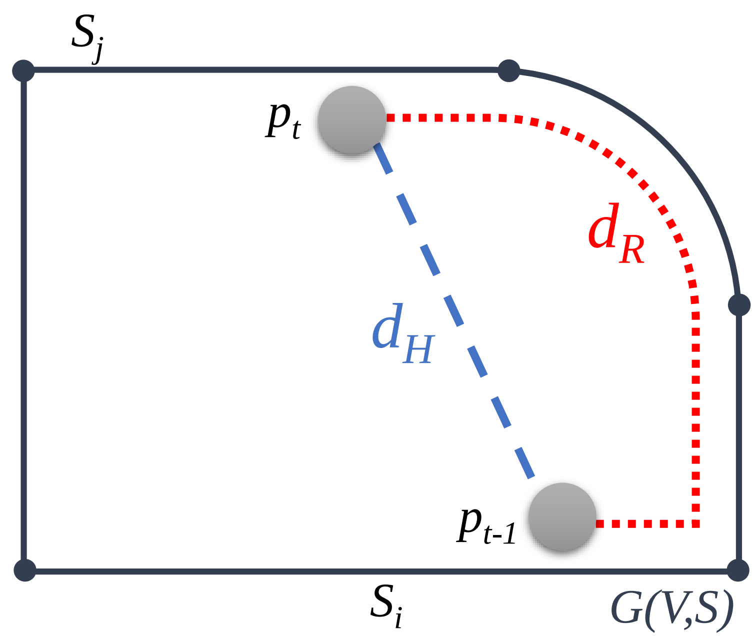
Map-Matching: Step B

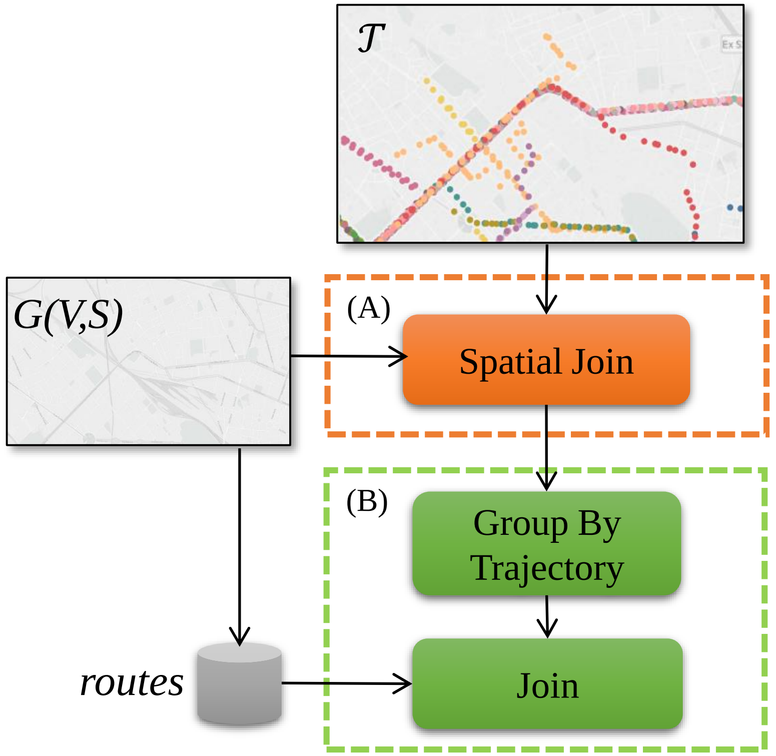
Transition probability estimation

$$P(X_t = j | X_{t-1} = i) = \frac{1}{\beta} e^{-d/\beta}$$

where

$$d = |d_H(p_t, p_{t-1}) - d_R(prj(p_t, s_j), prj(p_{t-1}, s_i))|$$





d_H = Haversine distance

d_R = shorthest path in $G(V,S)$

Map-Matching: Step B

Transition probability estimation

$$P(X_t = j | X_{t-1} = i) = \frac{1}{\beta} e^{-d/\beta}$$

where

$$d = |d_H(p_t, p_{t-1}) - d_R(prj(p_t, s_j), prj(p_{t-1}, s_i))|$$

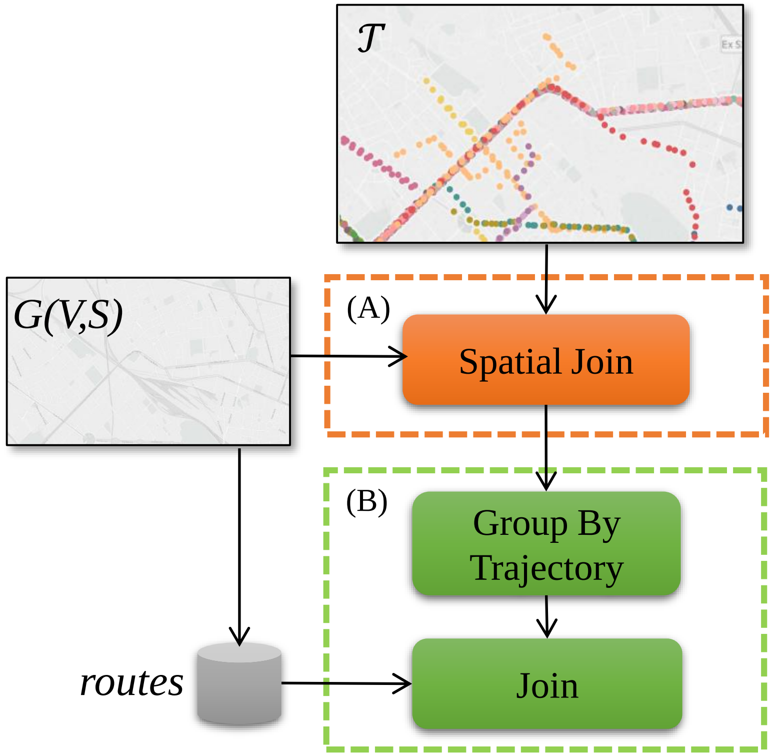
Compute *routes* (Segment/Segment shortest path)

Group $\mathcal{T}$ by trajectory T

Join consequent points in T to *routes* to estimate d_R

Efficiency: far road segments have a null transition probability

- ϑ bounds the routes depth
- γ bounds the routes length



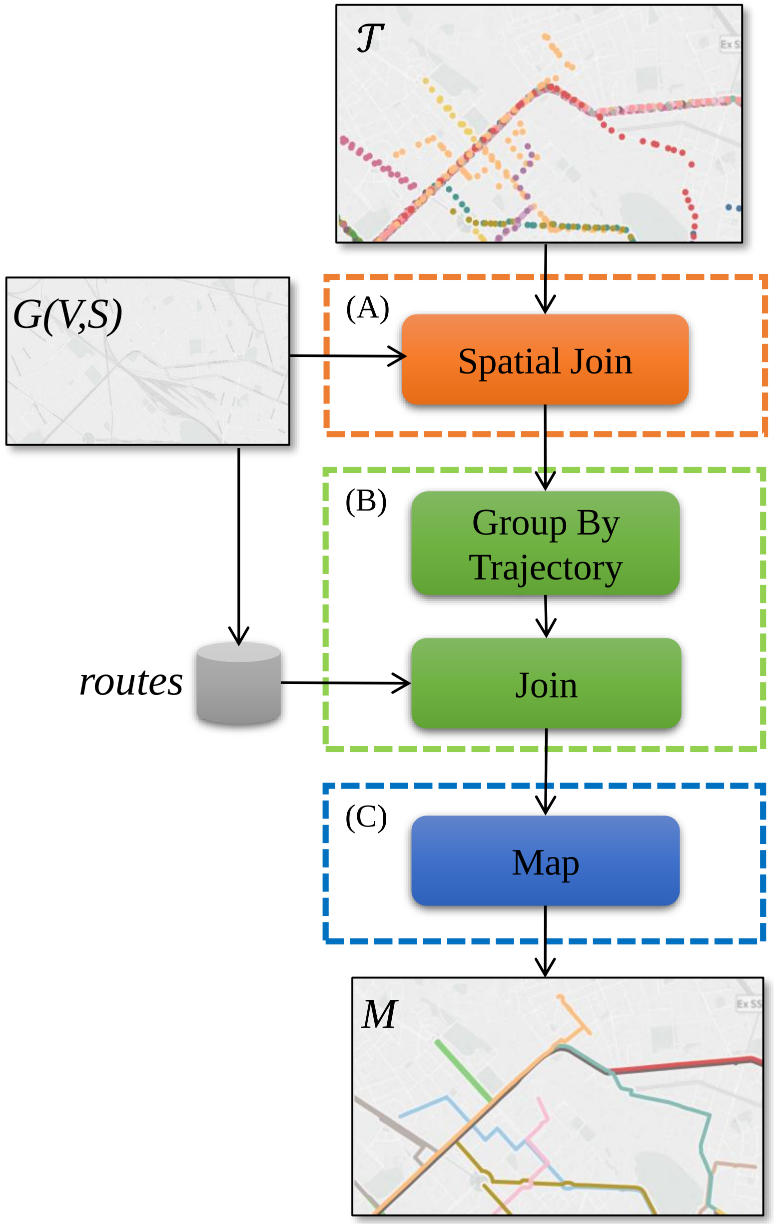
Map-Matching: Step C

Map-matching and aiding

- Viterbi algorithm returns matched routes
  - Bag-of-task (for each trajectory)
- Aiding fragmented routes
  - Consequent points can map to non-adjacent segments
  - Match all segments along the shortest path

Efficiency: tackle Viterbi algorithm complexity $O(|S|^2|T|)$

- $|T|$: trajectory simplification (out of scope)
- $|S|^2$: far road segments have a null probability
  - Tune neighborhood range τ and cardinality α in Step A



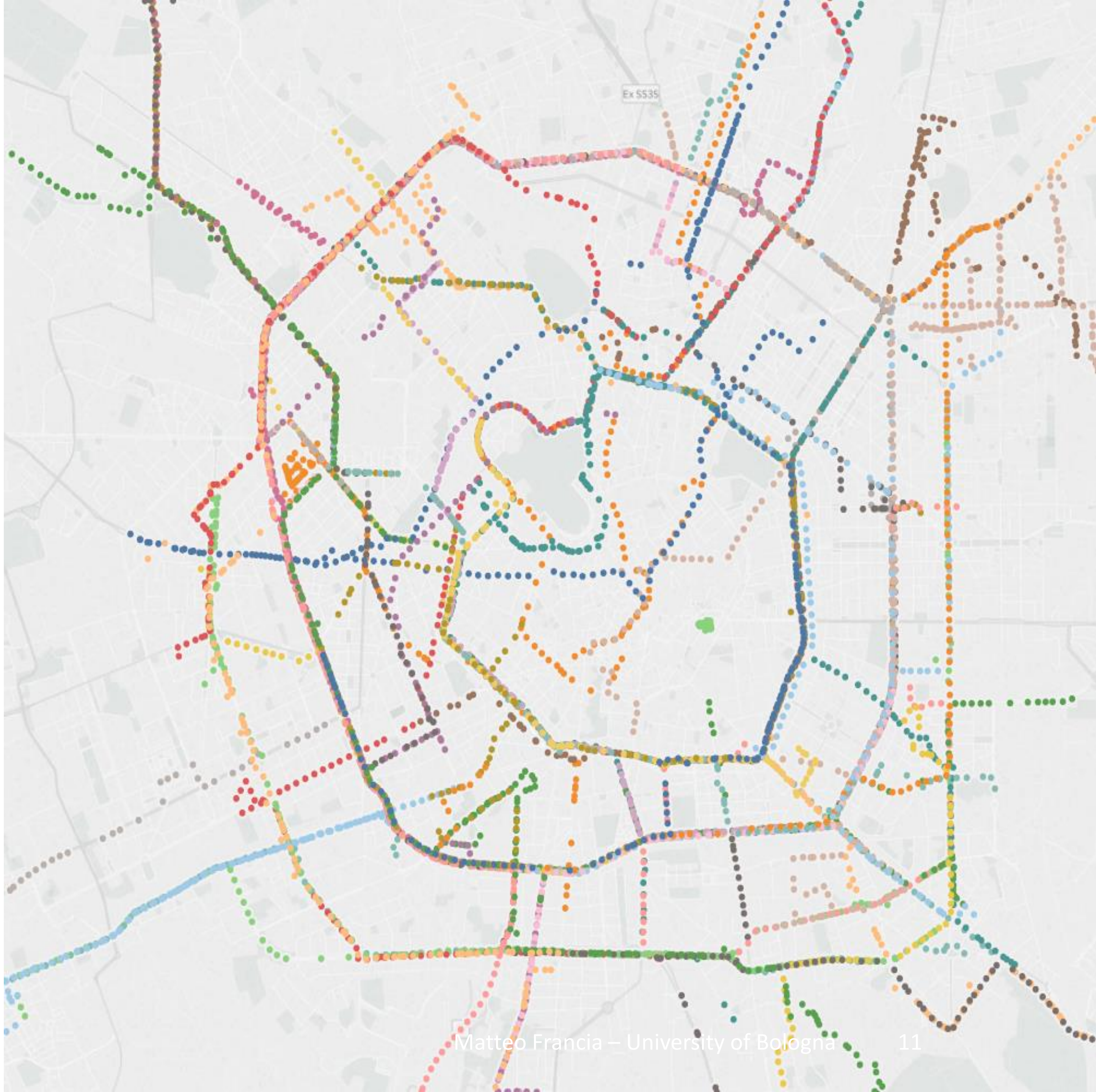
Evaluation: dataset

Case study in Milan

- 120K trajectories
- 8M GPS points
- Ground truth: 50 trajectories

Tests

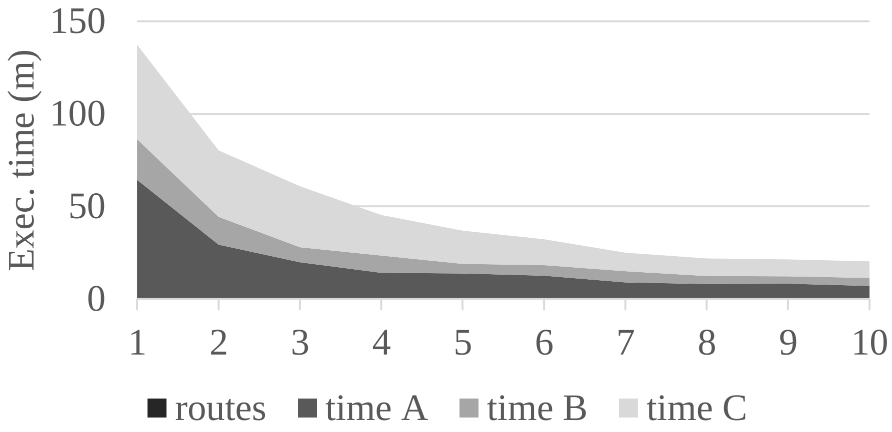
- Scaling up
- Step complexity



Evaluation: scaling up

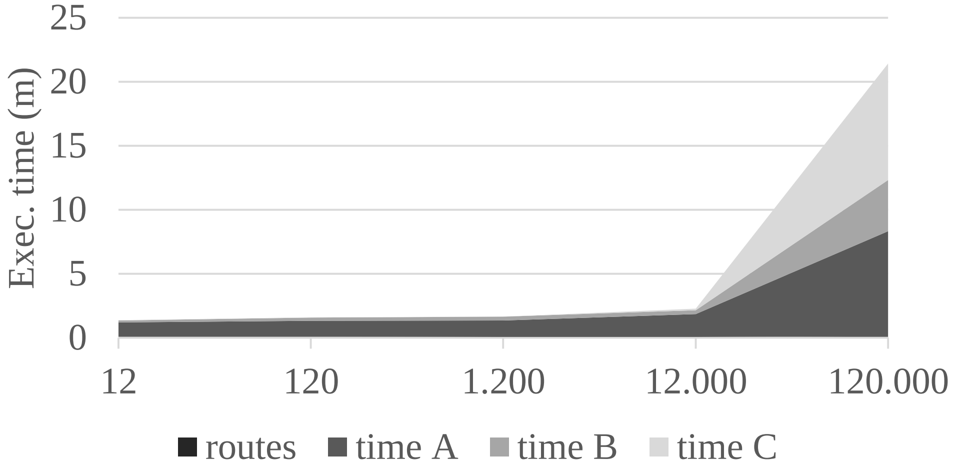
#Executors: [1 , 10]

- Execution time from 2h20m to 20m
- Speedup (of 7x) bounded by Viterbi and parallelization overhead

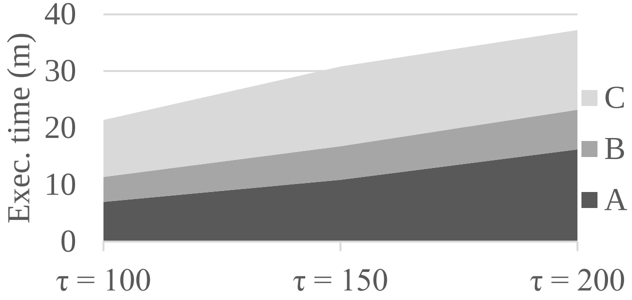


#Trajectories: [12, 120K]

- Execution time increases linearly with $|\mathcal{T}|$
- For small $|\mathcal{T}|$ the Spark's overhead overcomes actual workload times

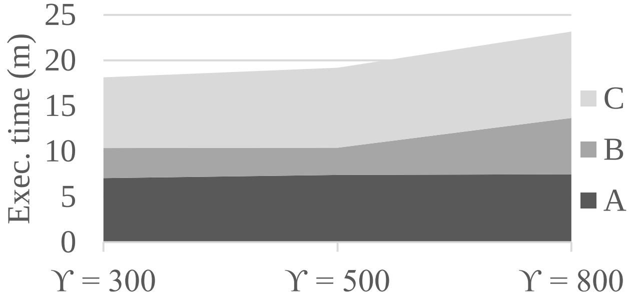


Evaluation: step complexity

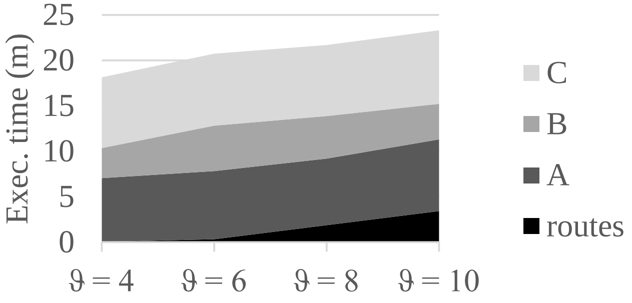


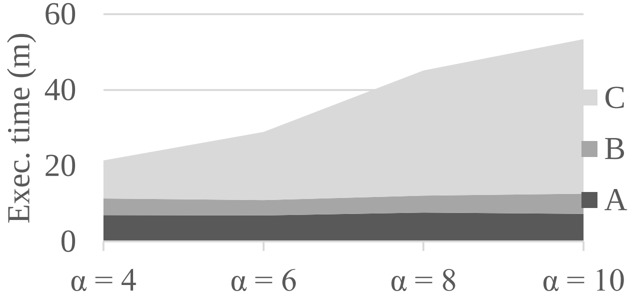
τ affects A (higher shuffle)

γ affects B (higher shuffle)

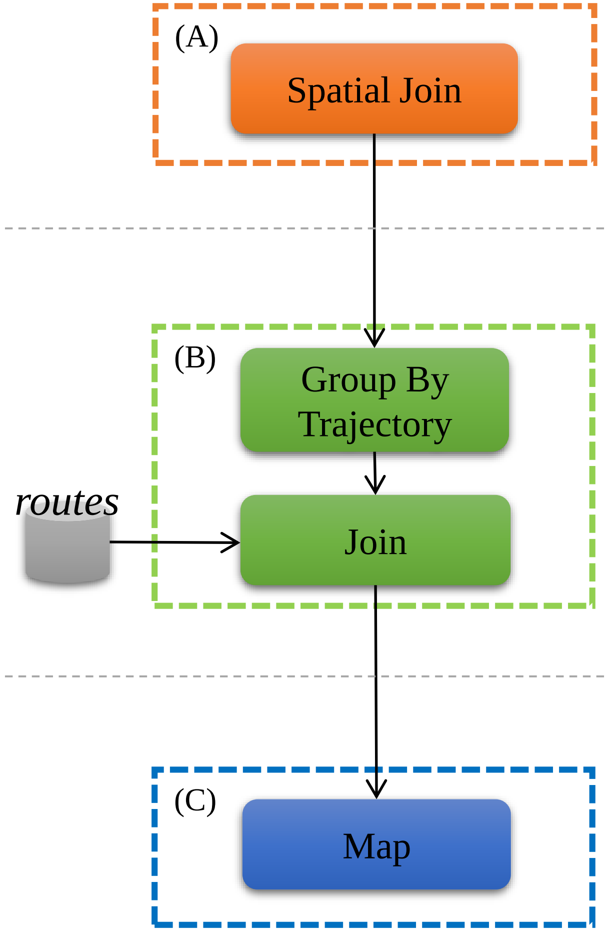


ϑ affects *routes* (more joins)





α affects C (bigger Viterbi's search space)



Conclusion

Contribution:

- Distributed Map-Matching on Spark
- Accuracy equivalent to sequential HMM [4]
  - Overcome limitations for highly fragmented urban networks
  - Map aiding embedded in the Viterbi computation

Future directions:

- Matching should adapt to high variance in:
  - Means of transportation (e.g., car, walk, bicycle)
  - Sampling rates of GPS points (e.g., from seconds to minutes)
- And scale up to huge road networks
  - From Milan to the entire region Lombardia

https://github.com/big-unibo/map-matching

References

[1] Xiaofang Zhou and Lei Li. Spatiotemporal data: Trajectories. In Encyclopedia of Big Data Technologies. Springer, 2019.

[2] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate GPS trajectories. In 17th ACM SIGSPATIAL Int. Symp. on Advances in Geographic Inform. Syst., ACM-GIS 2009, Nov. 4-6, 2009, Seattle, Washington, USA, Proc., pages 352–361, 2009.

[3] Paul Newson and John Krumm. Hidden markov Map-Matching through noise and sparseness. In 17th ACM SIGSPATIAL Int. Symp. on Advances in Geographic Inform. Syst., ACM-GIS 2009, Nov. 4-6, 2009, Seattle, Washington, USA, Proc., pages 336–343, 2009.

[4] An Luo, Shenghua Chen, and Bin Xv. Enhanced map-matching algorithm with a hidden markov model for mobile phone positioning. ISPRS Int. J. Geo-Inform., 6(11):327, 2017.

[5] Antonio M. R. Almeida, Maria I. V. Lima, José Antonio Fernandes de Macedo, and Javam C. Machado. DMM: A distributed map-matching algorithm using the mapreduce paradigm. In 19th IEEE Int. Conf. on Intelligent Transportation Syst., ITSC 2016.

[6] Ayman Zeidan, Eemil Lagerspetz, Kai Zhao, Petteri Nurmi, Sasu Tarkoma, and Huy T. Vo. Geomatch: Efficient large-scale Map-Matching on apache spark. In IEEE Int. Conf. on Big Data, Big Data 2018, Seattle, WA, USA, Dec. 10-13, 2018, pages 384–391, 2018.

[7] Douglas Alves Peixoto, Hung Quoc Viet Nguyen, Bolong Zheng, and Xiaofang Zhou. A framework for parallel map-matching at scale using spark. Distributed and Parallel Databases, pages 1– 24, 2018.

Thank you

Questions?

BUSINESS
INTELLIGENCE
GROUP