Inria

Université de Lille

Compact Storage of Data Streams in Mobile Devices

(with Fast Linear Interpolation)

R. Raes, O. Ruas, A. Luxey-Bitri & R. Rouvoy

Project members

Rémy RAES
Inria
Univ. Lille

Olivier RUAS
Pathway

Adrien LUXEY-BITRI
Univ. Lille
Inria

Romain ROUVOY
Univ. Lille
Inria

01

Context

- Smartphones
- Data (stream) consumers
- Data (stream) producers

- Smartphones
- Data (stream) consumers
- Data (stream) producers

Ubiquitous devices & data

- Smartphones
- Data (stream) consumers
- Data (stream) producers

- Local processing is challenging
- Constrained devices
  - Communication bandwidth
  - Processing capacities
  - Storage capacity

- Local processing is challenging
- Constrained devices
  - Communication bandwidth
  - Processing capacities
  - Storage capacity

- Local processing is challenging
- Constrained devices
  - > Communication bandwidth
  - > Processing capacities
  - > Storage capacity

- Local processing is challenging
- Constrained devices
  - Communication bandwidth
  - Processing capacities
  - Storage capacity

- Local processing is challenging
- Constrained devices
  - > Communication bandwidth
  - > Processing capacities
  - > Storage capacity

02

Storage solutions

State-of-the-art databases

- *Relational databases* (RDBMS)
  - > No emphasis on read/write performances
  - > Not designed for storing time series
- *Time series databases* (TSDB)
  - > Destructive retention policy
  - > DB process unstable
- *Moving objects databases* (MOD)
  - > Require storing raw samples on devices

- *Relational databases* (RDBMS)
  - > No emphasis on read/write performances
  - > Not designed for storing time series
- *Time series databases* (TSDB)
  - > Destructive retention policy
  - > DB process unstable
- *Moving objects databases* (MOD)
  - > Require storing raw samples on devices

- *Relational databases* (RDBMS)
  - > No emphasis on read/write performances
  - > Not designed for storing time series
- *Time series databases* (TSDB)
  - > Destructive retention policy
  - > DB process unstable
- *Moving objects databases* (MOD)
  - > Require storing raw samples on devices

- *Relational databases* (RDBMS)
  - > No emphasis on read/write performances
  - > Not designed for storing time series
- *Time series databases* (TSDB)
  - > Destructive retention policy
  - > DB process unstable
- *Moving objects databases* (MOD)
  - > Require storing raw samples on devices

- *Relational databases* (RDBMS)
  - > No emphasis on read/write performances
  - > Not designed for storing time series
- *Time series databases* (TSDB)
  - > Destructive retention policy
  - > DB process unstable
- *Moving objects databases* (MOD)
  - > Require storing raw samples on devices

- *Relational databases* (RDBMS)
  - > No emphasis on read/write performances
  - > Not designed for storing time series
- *Time series databases* (TSDB)
  - > Destructive retention policy
  - > DB process unstable
- *Moving objects databases* (MOD)
  - > Require storing raw samples on devices

- *Relational databases* (RDBMS)
  - > No emphasis on read/write performances
  - > Not designed for storing time series
- *Time series databases* (TSDB)
  - > Destructive retention policy
  - > DB process unstable
- *Moving objects databases* (MOD)
  - > Require storing raw samples on devices

- *Relational databases* (RDBMS)
  - > No emphasis on read/write performances
  - > Not designed for storing time series
- *Time series databases* (TSDB)
  - > Destructive retention policy
  - > DB process unstable
- *Moving objects databases* (MOD)
  - > Require storing raw samples on devices

- *Relational databases* (RDBMS)
  - > No emphasis on read/write performances
  - > Not designed for storing time series
- *Time series databases* (TSDB)
  - > Destructive retention policy
  - > DB process unstable
- *Moving objects databases* (MOD)
  - > Require storing raw samples on devices

- SWAB
- Greycat
- ShrinkingCone

Figure: SWAB uses a sliding buffer to do bottom-up segmentation.

- SWAB
- Greycat
- ShrinkingCone

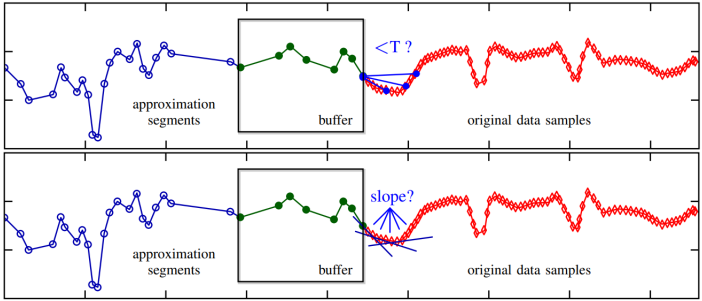


Figure: SWAB uses a sliding buffer to do bottom-up segmentation.

- SWAB
- Greycat
- ShrinkingCone

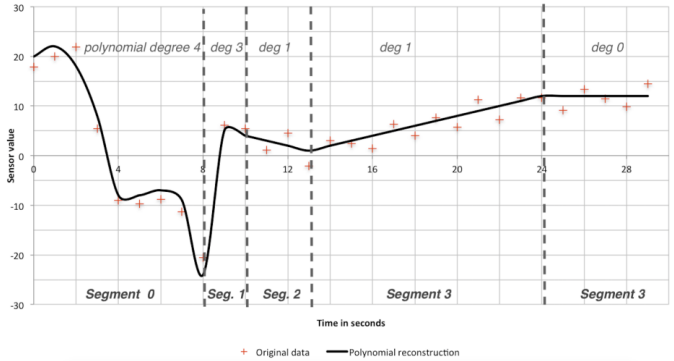


Figure: A segmentation and polynomial encoding of a continuous attribute with GREYCAT.

- SWAB
- Greycat
- ShrinkingCone

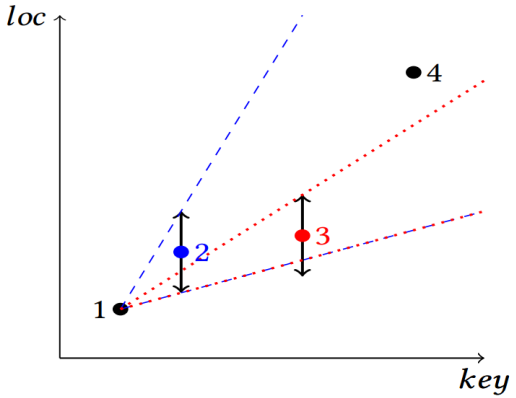


Figure: Successive acceptance spaces with SHRINKINGCONE.

Recap

- State-of-the-art databases
  - > Fail to store & index data streams (time series)
  - > Are not designed with I/O performance in mind
  - > Are incompatible with smartphones' constraints
- Data stream modeling techniques
  - > Require storing raw data samples
  - > Require analyzing data samples *a posteriori*
  - > Not available for smartphones

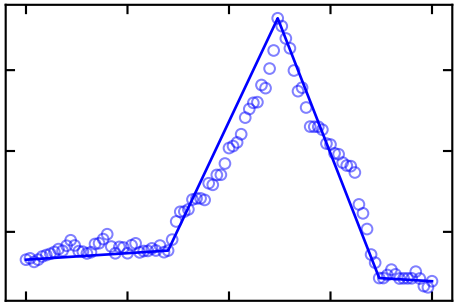
- State-of-the-art databases
  - > Fail to store & index data streams (time series)
  - > Are not designed with I/O performance in mind
  - > Are incompatible with smartphones' constraints
- Data stream modeling techniques
  - > Require storing raw data samples
  - > Require analyzing data samples *a posteriori*
  - > Not available for smartphones

- State-of-the-art databases
  - > Fail to store & index data streams (time series)
  - > Are not designed with I/O performance in mind
  - > Are incompatible with smartphones' constraints
- Data stream modeling techniques
  - > Require storing raw data samples
  - > Require analyzing data samples *a posteriori*
  - > Not available for smartphones

03

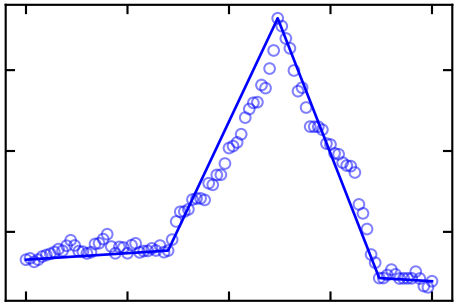
Fast Linear Interpolation: FLI

Fast linear interpolation



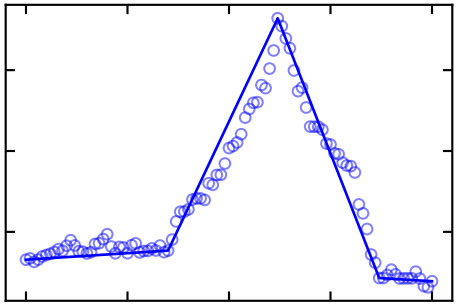
- Fast? Rather simple!
- (very graphic!)
- Relies on linear interpolation

Fast linear interpolation

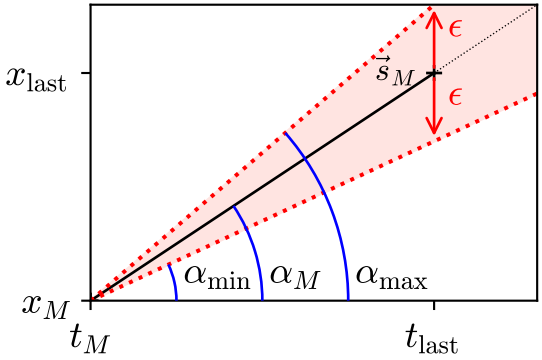


- Fast? Rather simple!
- (very graphic!)
- Relies on linear interpolation

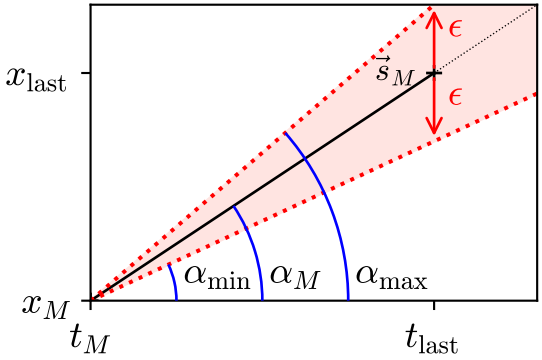
Fast linear interpolation



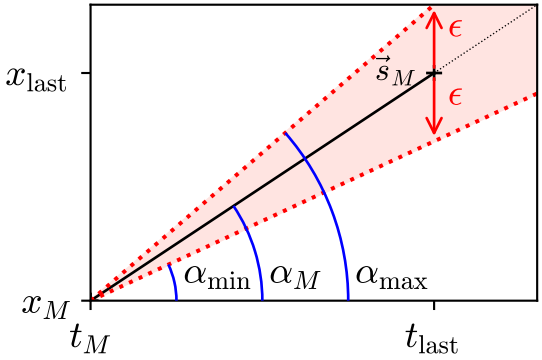
- Fast? Rather simple!
- (very graphic!)
- Relies on linear interpolation



- Parameter ϵ: tolerated error
- Persisted model: (t_M, x_M, α_M)
- In-memory: $(\alpha_{\min}, \alpha_{\max})$ cone

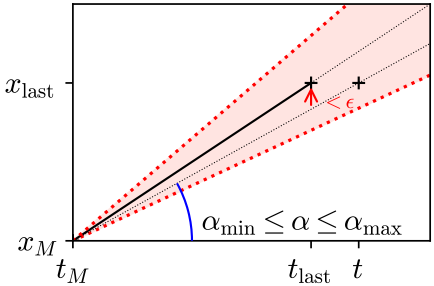


- Parameter ϵ: tolerated error
- Persisted model: (t_M, x_M, α_M)
- In-memory: $(\alpha_{\min}, \alpha_{\max})$ cone



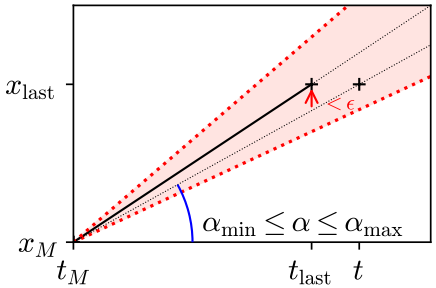
- Parameter ϵ: tolerated error
- Persisted model: (t_M, x_M, α_M)
- In-memory: $(\alpha_{\min}, \alpha_{\max})$ cone

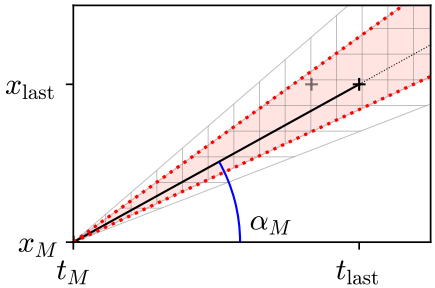
Data modeling: adding a fitting point



- New point fits current model
- Model is updated

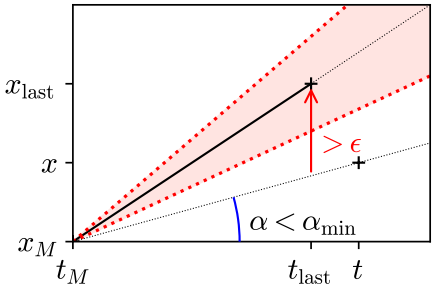
Data modeling: adding a fitting point





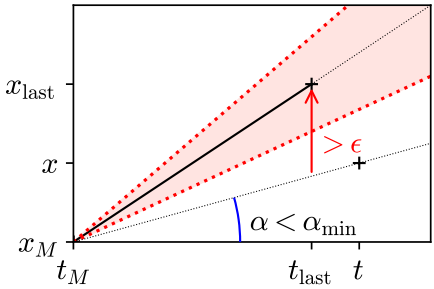
- New point fits current model
- Model is updated

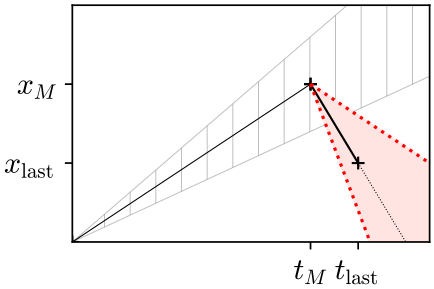
Data modeling: adding a non-fitting point



- New point does not fit current model
- Model is saved, and a new one is created

Data modeling: adding a non-fitting point





- New point does not fit current model
- Model is saved, and a new one is created

- The quality of the models directly depends on the selected ϵ

Figure: A variety of bad choices (overfitting and underfitting, respectively) for epsilon value.

- The quality of the models directly depends on the selected ϵ

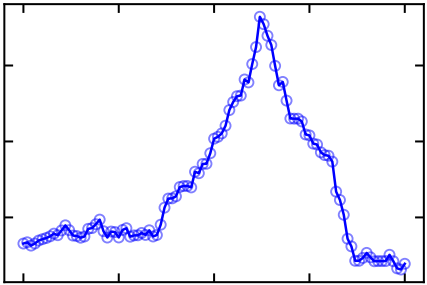


Figure: A variety of bad choices (overfitting and underfitting, respectively) for epsilon value.

- The quality of the models directly depends on the selected ϵ

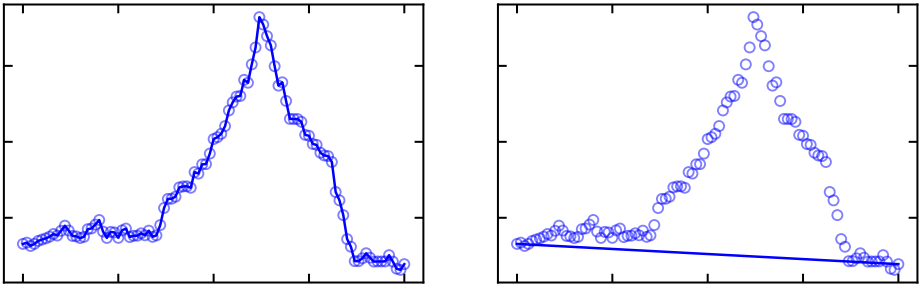


Figure: A variety of bad choices (overfitting and underfitting, respectively) for epsilon value.

04

Evaluation

- Location datasets
- Cabspotting
  - 536 taxis
  - San Francisco, CA, USA
  - 1 month
  - 11M GPS records
  - 388 MB
- Privamov
  - 100 users
  - Lyon, France
  - 15 months
  - 156M GPS records
  - 7.2 GB

- Location datasets
- Cabspotting
  - 536 taxis
  - San Francisco, CA, USA
  - 1 month
  - 11M GPS records
  - 388 MB
- Privamov
  - 100 users
  - Lyon, France
  - 15 months
  - 156M GPS records
  - 7.2 GB

- Location datasets
- CABSPOTTING
  - 536 taxis
  - San Francisco, CA, USA
  - 1 month
  - 11M GPS records
  - 388 MB
- PRIVAMOV
  - 100 users
  - Lyon, France
  - 15 months
  - 156M GPS records
  - 7.2 GB

- Selecting a good ϵ value requires data domain knowledge
- Drift between consecutive values (x_1, y_1) and (x_2, y_2): $|y_2 - y_1|/|x_2 - x_1|$.

Figure: CDF of latitude and longitude variations of successive locations in CABSPOTTING and PRIVAMOV.

- We used $\epsilon = 10^{-3}$ as a baseline value

- Selecting a good ϵ value requires data domain knowledge
- Drift between consecutive values (x_1, y_1) and (x_2, y_2): $|y_2 - y_1|/|x_2 - x_1|$.

Figure: CDF of latitude and longitude variations of successive locations in CABSPOTTING and PRIVAMOV.

- We used $\epsilon = 10^{-3}$ as a baseline value

- Selecting a good ϵ value requires data domain knowledge
- Drift between consecutive values (x_1, y_1) and (x_2, y_2): $|y_2 - y_1|/|x_2 - x_1|$.

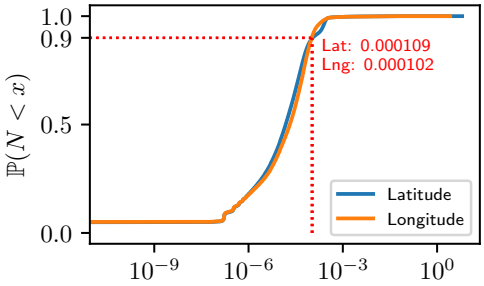


Figure: CDF of latitude and longitude variations of successive locations in CABSPOTTING and PRIVAMOV.

- We used $\epsilon = 10^{-3}$ as a baseline value

- Selecting a good ϵ value requires data domain knowledge
- Drift between consecutive values (x_1, y_1) and (x_2, y_2): $|y_2 - y_1|/|x_2 - x_1|$.

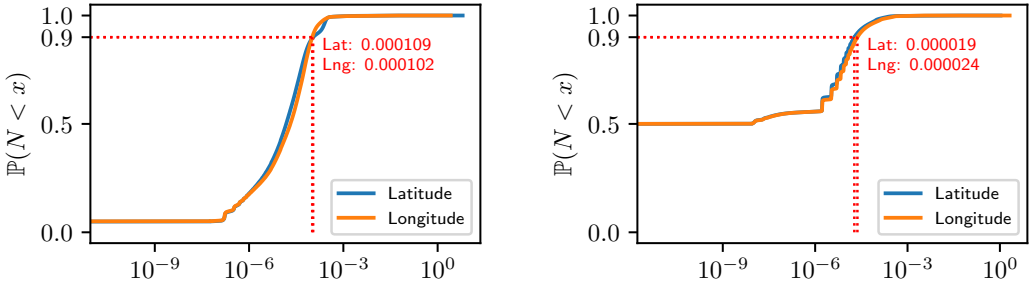


Figure: CDF of latitude and longitude variations of successive locations in CABSPOTTING and PRIVAMOV.

- We used $\epsilon = 10^{-3}$ as a baseline value

- Selecting a good ϵ value requires data domain knowledge
- Drift between consecutive values (x_1, y_1) and (x_2, y_2): $|y_2 - y_1|/|x_2 - x_1|$.

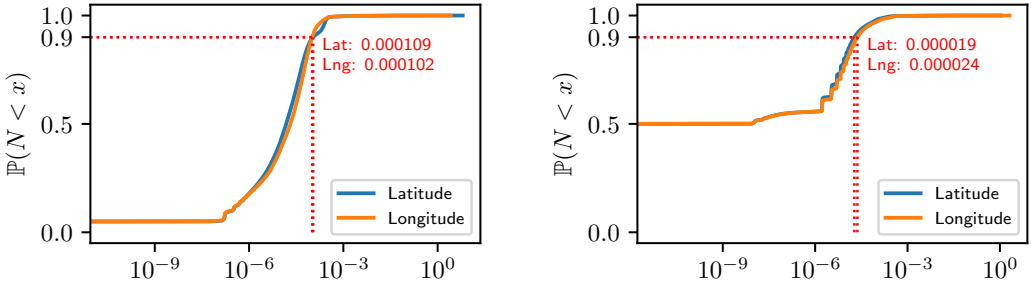


Figure: CDF of latitude and longitude variations of successive locations in CABSPOTTING and PRIVAMOV.

- We used $\epsilon = 10^{-3}$ as a baseline value

- When $\epsilon = 10^{-3}$
  - Latitude max error: $\approx 111\,m$
  - Longitude max error: $\approx 88\,m$ (Cabspotting) / $\approx 77\,m$ (Privamov)
- Controlling data accuracy
  - Aligning with sensor uncertainty
  - Adjusting privacy / utility trade-off

- When $\epsilon = 10^{-3}$
  - > Latitude max error: $\approx 111\,m$
  - > Longitude max error: $\approx 88\,m$ (CABSPOTTING) / $\approx 77\,m$ (PRIVAMOV)
- Controlling data accuracy
  - > Aligning with sensor uncertainty
  - > Adjusting privacy / utility trade-off

- When $\epsilon = 10^{-3}$
  - > Latitude max error: $\approx 111\,m$
  - > Longitude max error: $\approx 88\,m$ (CABSPOTTING) / $\approx 77\,m$ (PRIVAMOV)
- Controlling data accuracy
  - > Aligning with sensor uncertainty
  - > Adjusting privacy / utility trade-off

- When $\epsilon = 10^{-3}$
  - > Latitude max error: $\approx 111\,m$
  - > Longitude max error: $\approx 88\,m$ (CABSPOTTING) / $\approx 77\,m$ (PRIVAMOV)
- Controlling data accuracy
  - > Aligning with sensor uncertainty
  - > Adjusting privacy / utility trade-off

- When $\epsilon = 10^{-3}$
  - > Latitude max error: $\approx 111\,m$
  - > Longitude max error: $\approx 88\,m$ (CABSPOTTING) / $\approx 77\,m$ (PRIVAMOV)
- Controlling data accuracy
  - > Aligning with sensor uncertainty
  - > Adjusting privacy / utility trade-off

- When $\epsilon = 10^{-3}$
  - > Latitude max error: $\approx 111\,m$
  - > Longitude max error: $\approx 88\,m$ (CABSPOTTING) / $\approx 77\,m$ (PRIVAMOV)
- Controlling data accuracy
  - > Aligning with sensor uncertainty
  - > Adjusting privacy / utility trade-off

- Original size: 388 MB
- $\epsilon = 10^{-3} \Rightarrow$ 307 MB (22% gain)
- $\epsilon = 2 \times 10^{-3} \Rightarrow$ 202 MB (47.9% gain)

Memory benchmark: CABSPOTTING results

- Original size: 388 MB
- $\epsilon = 10^{-3} \Rightarrow$ 307 MB (22% gain)
- $\epsilon = 2 \times 10^{-3} \Rightarrow$ 202 MB (47.9% gain)

- Original size: 388 MB
- $\epsilon = 10^{-3} \Rightarrow$ 307 MB (22% gain)
- $\epsilon = 2 \times 10^{-3} \Rightarrow$ 202 MB (47.9% gain)

- Original size: 388 MB
- $\epsilon = 10^{-3} \Rightarrow$ 307 MB (22% gain)
- $\epsilon = 2 \times 10^{-3} \Rightarrow$ 202 MB (47.9% gain)

- Original size: 388 MB
- $\epsilon = 10^{-3} \Rightarrow$ 307 MB (22% gain)
- $\epsilon = 2 \times 10^{-3} \Rightarrow$ 202 MB (47.9% gain)

Memory benchmark: PRIVAMOV results

- $\epsilon = 10^{-3}$
- From 7.2 GB to 25 MB
  - > (data utility study to come later)
- Fits in memory! (compared to SQLite)

- $\epsilon = 10^{-3}$
- From 7.2 GB to 25 MB
  - > (data utility study to come later)
- Fits in memory! (compared to SQLite)

- $\epsilon = 10^{-3}$
- From 7.2 GB to 25 MB
  - > (data utility study to come later)
- Fits in memory! (compared to SQLite)

- $\epsilon = 10^{-3}$
- From 7.2 GB to 25 MB
  - $>$ (data utility study to come later)
- Fits in memory! (compared to SQLite)

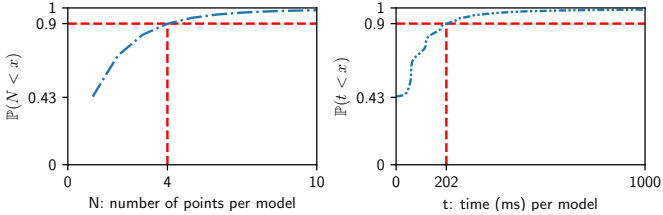


Figure: CABSPOTTING modeling

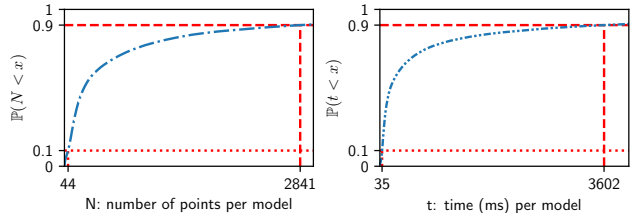


Figure: PRIVAMOV modeling

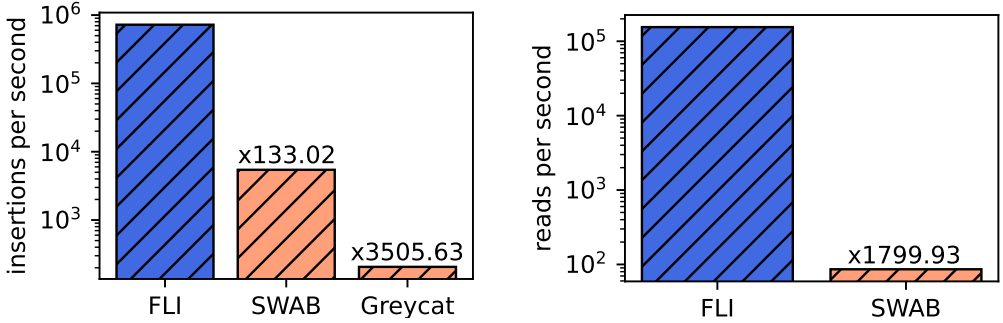


Figure: Throughput for insertions and reads using FLI, SWAB, and GREYCAT (log scale).

05

Conclusion

What's next?

- *In situ* computing: privacy protection
- Controlled ϵ compression (priority to newer data with size constraint)
- Integration with federated learning (pattern/anomaly detection)

What's next?

- *In situ* computing: privacy protection
- Controlled ϵ compression (priority to newer data with size constraint)
- Integration with federated learning (pattern/anomaly detection)

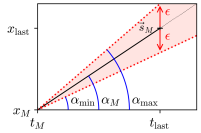
- *In situ* computing: privacy protection
- Controlled ϵ compression (priority to newer data with size constraint)
- Integration with federated learning (pattern/anomaly detection)

Take away

Edge services

- Local processing is challenging
- Constrained devices
  - > Communication bandwidth
  - > Processing capacities
  - > Storage capacity

Data modeling



- Parameter ϵ: tolerated error
- Persisted model: (t_M, x_M, α_M)
- In-memory: $(\alpha_{\min}, \alpha_{\max})$ cone

About the *epsilon* value

- Selecting a good ϵ value requires data domain knowledge
- Drift between consecutive values (x_1, y_1) and (x_2, y_2): $|y_2 - y_1|/|x_2 - x_1|$.

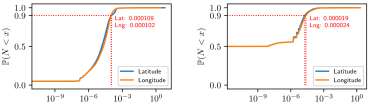


Figure: CDF of latitude and longitude variations of successive locations in CABSPOTTING and PRIVAMOV.

- We used $\epsilon = 10^{-3}$ as a baseline value

I/O throughput

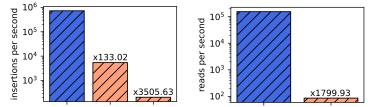


Figure: Throughput for insertions and reads using FLI, SWAB, and GREYCAT (log scale).

Inria

Université de Lille

Compact Storage of Data Streams in Mobile Devices

(with Fast Linear Interpolation)

R. Raes, O. Ruas, A. Luxey-Bitri & R. Rouvoy

- Latitude
  - > Tolerated error: 10^{-3} *deg* ≈ 111 *m*
  - > Median error: 5.33×10^{-5}
  - > RMSE: 3.72×10^{-4}
- Longitude
  - > Tolerated error: 10^{-3} *deg* ≈ 88 *m*
  - > Median error: 2.81×10^{-5}
  - > RMSE: 3.44×10^{-4}
- Privacy utility

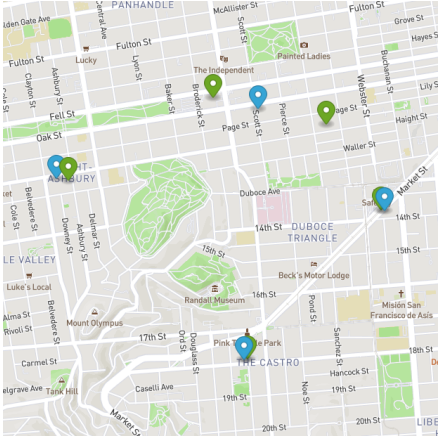


Figure: Points of Interest computed using raw data and FLI-modeled data.

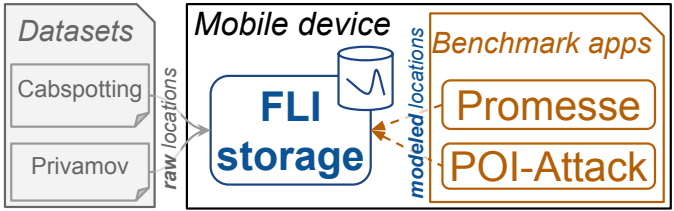


Figure: FLI enables big data processing directly on smartphones.

Algorithm 1 FLAIR insertion using parameter $\epsilon \in \mathbb{R}^{+*}$

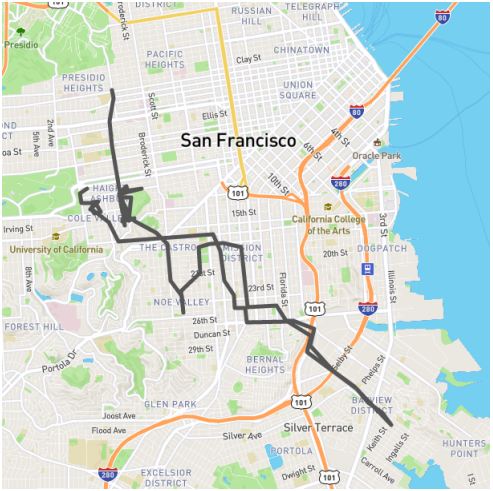
Before: $M; (x_0, x_{t-1}) \in \mathbb{R}^{2+}; (y_0, y_{t-1}, A_0, A_{\min}, A_{\max}) \in \mathbb{R}^5$

function INSERT($x_t \in \mathbb{R}^+, y_t \in \mathbb{R}$)
  $(x_t^\Delta, y_t^\Delta) \leftarrow (x_t - x_0, y_t - y_0)$ ▷ Compute A_t
  $A_t \leftarrow y_t^\Delta / x_t^\Delta$
  if $A_{\min} \leq A_t \leq A_{\max}$ **then**
    $A_0 \leftarrow A_t$ ▷ Update model
    $A_{\min} \leftarrow \max\left(A_{\min}, \frac{y_t^\Delta - \epsilon}{x_t^\Delta}\right)$
    $A_{\max} \leftarrow \min\left(A_{\max}, \frac{y_t^\Delta + \epsilon}{x_t^\Delta}\right)$
  else
    $\mathcal{M}$.insert(x_0, y_0, A_0) ▷ Persist model
    $(x_0, y_0) \leftarrow (x_{t-1}, y_{t-1})$ ▷ Build new model
    $(x_t^\Delta, y_t^\Delta) \leftarrow (x_t - x_0, y_t - y_0)$
    $A_0 \leftarrow y_t^\Delta / x_t^\Delta$
    $A_{\min} \leftarrow \left(y_t^\Delta - \epsilon\right) / x_t^\Delta$
    $A_{\max} \leftarrow \left(y_t^\Delta + \epsilon\right) / x_t^\Delta$
  end if
  $(x_{t-1}, y_{t-1}) \leftarrow (x_t, y_t)$ ▷ Update penultimate
end function

Algorithm 2 FLAIR approximate read

Before: Current model (x_0, y_0, A_0);
Memory $\mathcal{M}$ containing previous models

function READ($x \in \mathbb{R}^+$)
  if $x_0 \leq x$ **then**
    return $A_0 \times (x - x_0) + y_0$
  end if
  Select i s.t. $(x_i, y_i, A_i) \in \mathcal{M} \wedge x_i \leq x < x_{i+1}$
  return $A_i \times (x - x_i) + y_i$
end function



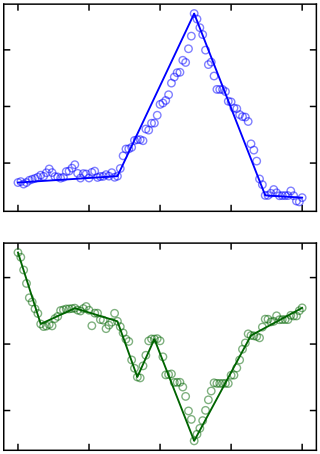


Figure: Modeled latitude & longitude.

Image credits

- Olivier Ruas, Rémy Raes, Adrien Luxey-Bitri & Romain Rouvoy
- © SQLite, TimescaleDB & InfluxDB
- © Freepik