Information Retention in Trimmed Datasets

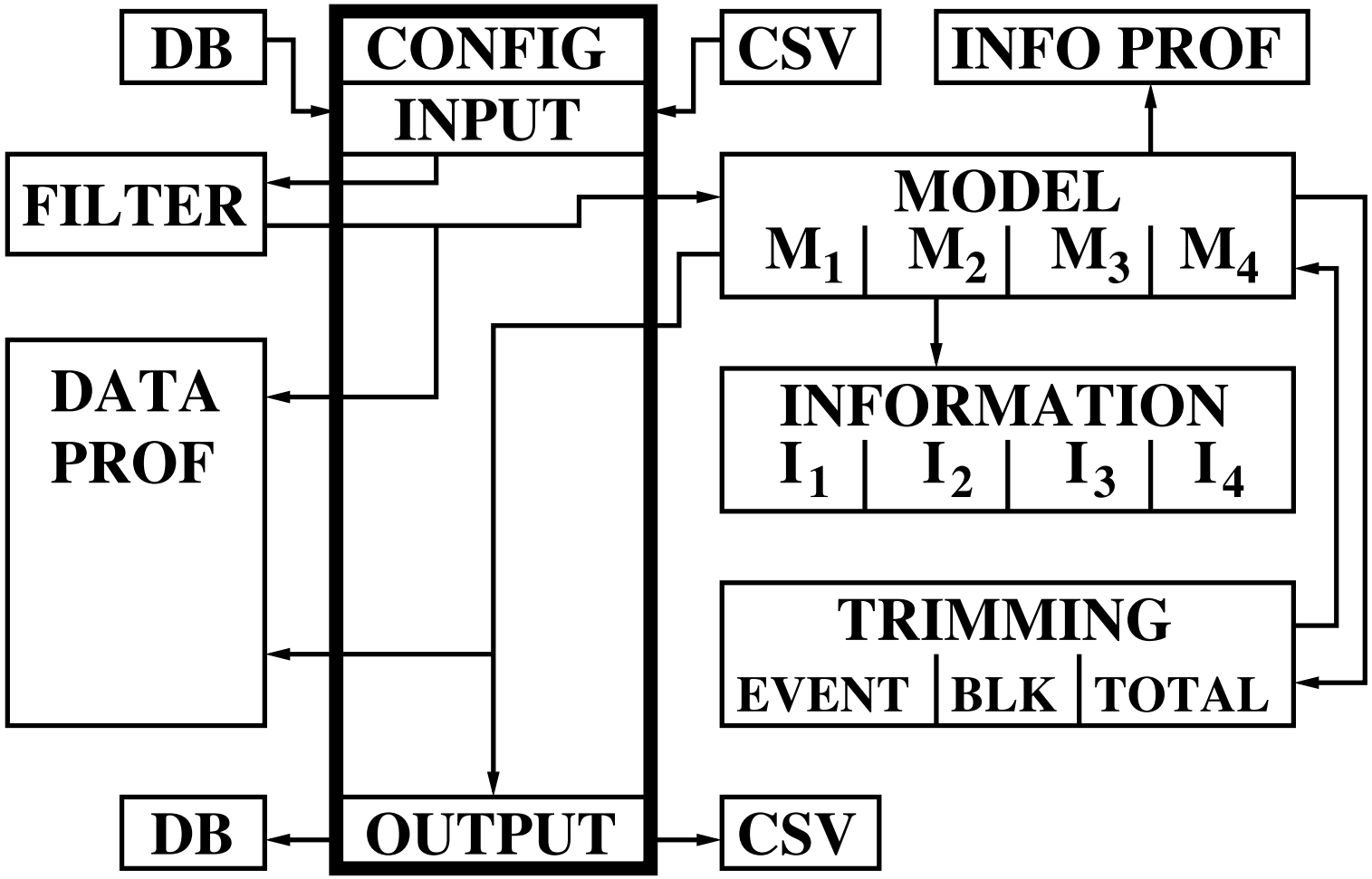
F. B. Manolache[1] Zhilan Wang[1] Xinran Su[1] O. Rusu[2]

[1]Carnegie Mellon University Pittsburgh, PA, USA

[2]Alexandru Ioan Cuza University Iasi, Romania

24-th RoEduNet International Conference, Chisinau, 2025

- Architecture -



- Features -

- reverse trimming: -r flag
- database interaction: *db* input/output command line argument
- timestamp column: -t command line argument
- amount of trimming: -n absolute value, -c percentage
- specifics: -m model, -s strategy;
- trimming efficiency analysis: -e flag
- exclude classes: -x column list
- preprocessing filters: -f filter list
- duplicate rows elimination as step 0 of trimming: -d 1 (rows) or 2 (events)
- anomaly thresholds for profiling tests: in the *profile* section of the configuration file.

- Features -

- reverse trimming: -r flag
- database interaction: *db* input/output command line argument
- timestamp column: -t command line argument
- amount of trimming: -n absolute value, -c percentage
- specifics: -m model, -s strategy;
- trimming efficiency analysis: -e flag
- exclude classes: -x column list
- preprocessing filters: -f filter list
- duplicate rows elimination as step 0 of trimming: -d 1 (rows) or 2 (events)
- anomaly thresholds for profiling tests: in the *profile* section of the configuration file.

- Features -

- reverse trimming: -r flag
- database interaction: *db* input/output command line argument
- timestamp column: -t command line argument
- amount of trimming: -n absolute value, -c percentage
- specifics: -m model, -s strategy;
- trimming efficiency analysis: -e flag
- exclude classes: -x column list
- preprocessing filters: -f filter list
- duplicate rows elimination as step 0 of trimming: -d 1 (rows) or 2 (events)
- anomaly thresholds for profiling tests: in the *profile* section of the configuration file.

- Features -

- reverse trimming: -r flag
- database interaction: *db* input/output command line argument
- timestamp column: -t command line argument
- amount of trimming: -n absolute value, -c percentage
- specifics: -m model, -s strategy;
- trimming efficiency analysis: -e flag
- exclude classes: -x column list
- preprocessing filters: -f filter list
- duplicate rows elimination as step 0 of trimming: -d 1 (rows) or 2 (events)
- anomaly thresholds for profiling tests: in the *profile* section of the configuration file.

- Features -

- reverse trimming: -r flag
- database interaction: *db* input/output command line argument
- timestamp column: -t command line argument
- amount of trimming: -n absolute value, -c percentage
- specifics: -m model, -s strategy;
- trimming efficiency analysis: -e flag
- exclude classes: -x column list
- preprocessing filters: -f filter list
- duplicate rows elimination as step 0 of trimming: -d 1 (rows) or 2 (events)
- anomaly thresholds for profiling tests: in the *profile* section of the configuration file.

- Features -

- reverse trimming: -r flag
- database interaction: *db* input/output command line argument
- timestamp column: -t command line argument
- amount of trimming: -n absolute value, -c percentage
- specifics: -m model, -s strategy;
- trimming efficiency analysis: -e flag
- exclude classes: -x column list
- preprocessing filters: -f filter list
- duplicate rows elimination as step 0 of trimming: -d 1 (rows) or 2 (events)
- anomaly thresholds for profiling tests: in the *profile* section of the configuration file.

- Features -

- reverse trimming: -r flag
- database interaction: *db* input/output command line argument
- timestamp column: -t command line argument
- amount of trimming: -n absolute value, -c percentage
- specifics: -m model, -s strategy;
- trimming efficiency analysis: -e flag
- exclude classes: -x column list
- preprocessing filters: -f filter list
- duplicate rows elimination as step 0 of trimming: -d 1 (rows) or 2 (events)
- anomaly thresholds for profiling tests: in the *profile* section of the configuration file.

- Features -

- reverse trimming: -r flag
- database interaction: *db* input/output command line argument
- timestamp column: -t command line argument
- amount of trimming: -n absolute value, -c percentage
- specifics: -m model, -s strategy;
- trimming efficiency analysis: -e flag
- exclude classes: -x column list
- preprocessing filters: -f filter list
- duplicate rows elimination as step 0 of trimming: -d 1 (rows) or 2 (events)
- anomaly thresholds for profiling tests: in the *profile* section of the configuration file.

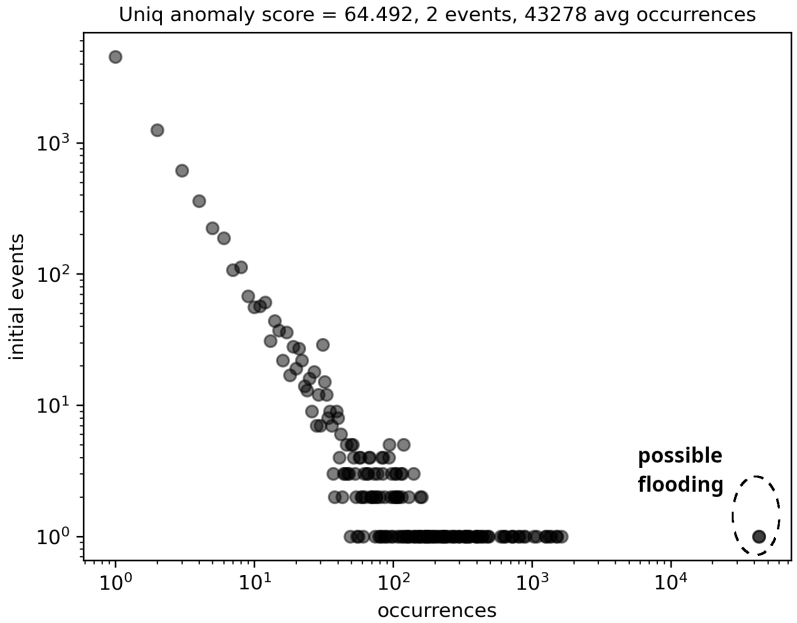
- Features -

- reverse trimming: -r flag
- database interaction: *db* input/output command line argument
- timestamp column: -t command line argument
- amount of trimming: -n absolute value, -c percentage
- specifics: -m model, -s strategy;
- trimming efficiency analysis: -e flag
- exclude classes: -x column list
- preprocessing filters: -f filter list
- duplicate rows elimination as step 0 of trimming: -d 1 (rows) or 2 (events)
- anomaly thresholds for profiling tests: in the *profile* section of the configuration file.

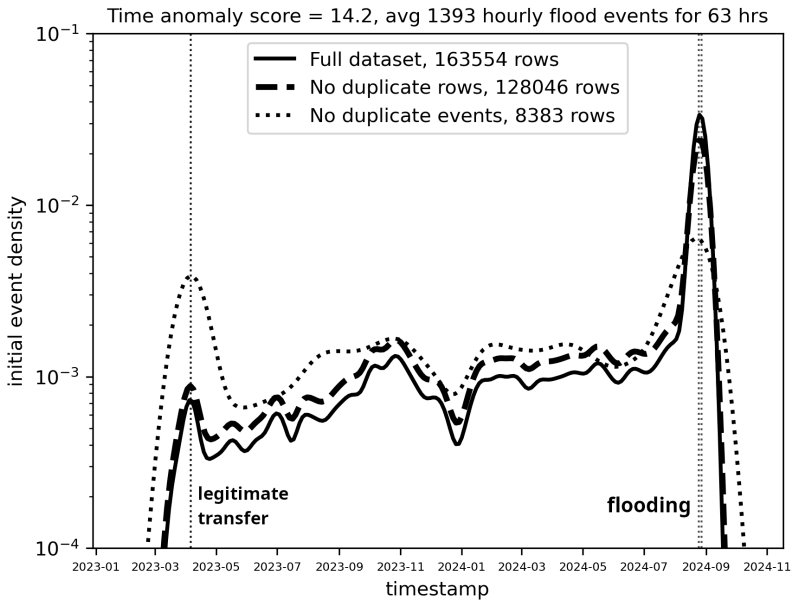
- Features -

- reverse trimming: -r flag
- database interaction: *db* input/output command line argument
- timestamp column: -t command line argument
- amount of trimming: -n absolute value, -c percentage
- specifics: -m model, -s strategy;
- trimming efficiency analysis: -e flag
- exclude classes: -x column list
- preprocessing filters: -f filter list
- duplicate rows elimination as step 0 of trimming: -d 1 (rows) or 2 (events)
- anomaly thresholds for profiling tests: in the *profile* section of the configuration file.

- Profiling -



(a) Event uniqueness distribution with hints of event flooding and a high anomaly score



(b) Timestamp distribution with hints of event flooding for the real flood but also for a spike in legitimate activity

- Computation Speed -

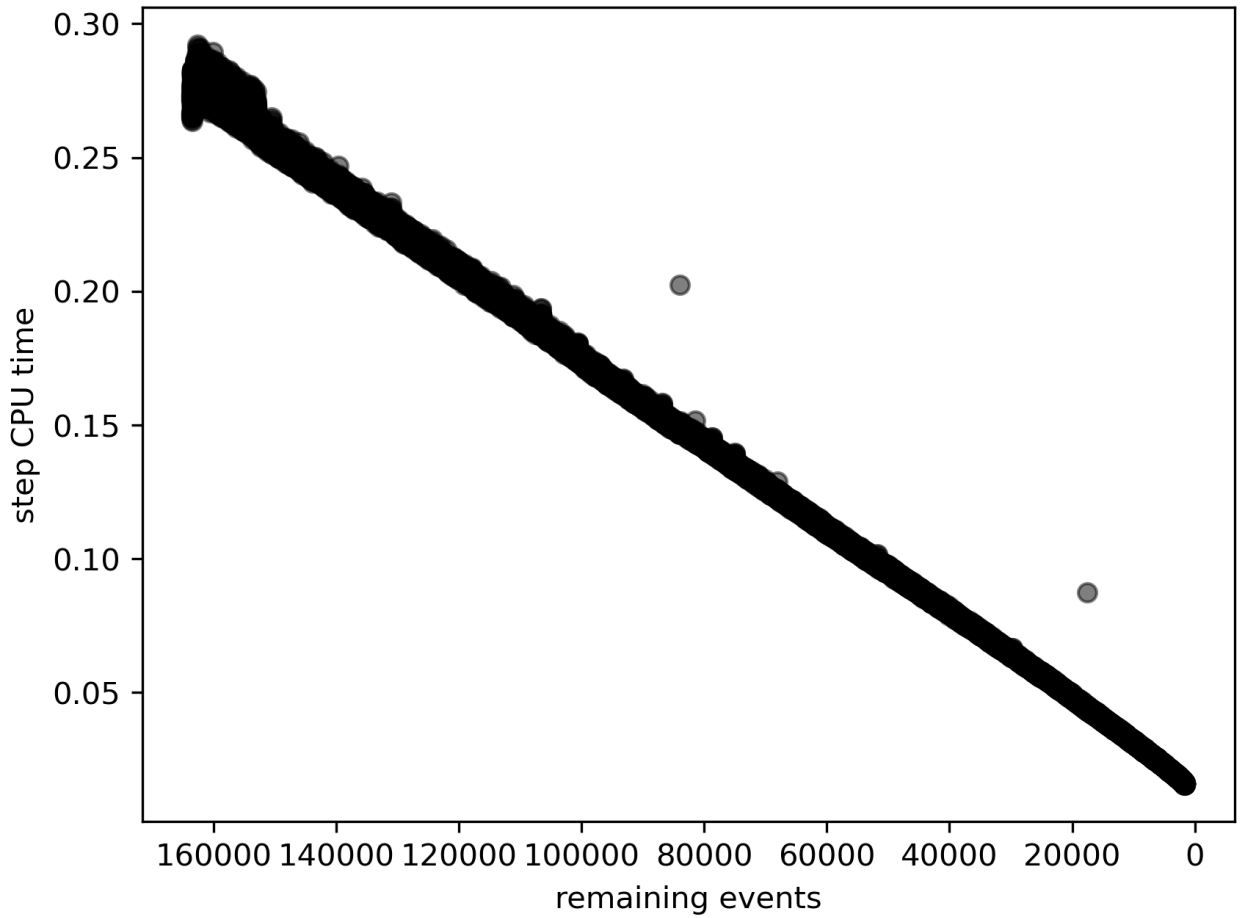
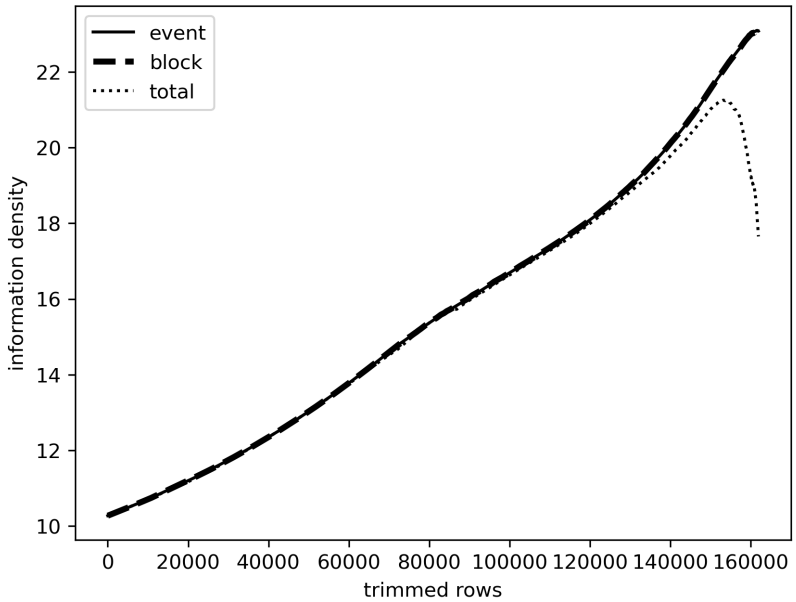
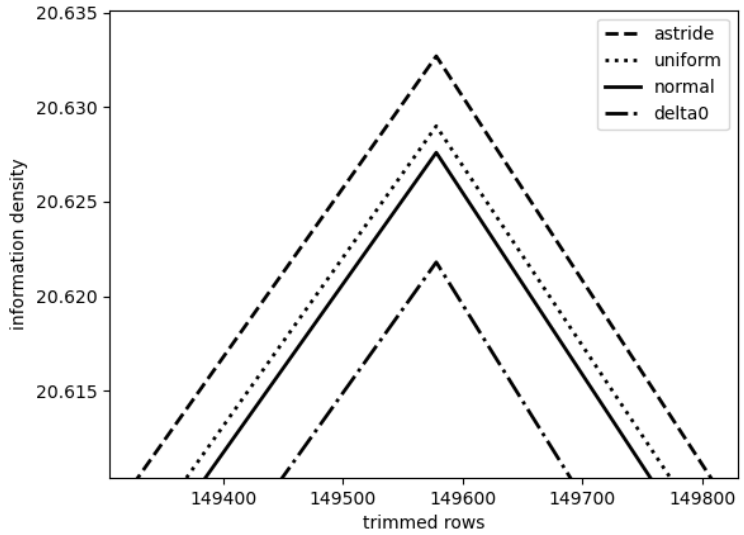


Figure: Step CPU time by the number of remaining events for IPM, event strategy, astride procedure.

- Information Density -



(a) Information density for the astride procedure, calculated by the event, block, and total strategies using IPM



(b) Information density around the maximum value, calculated by astride, uniform, normal, and delta0 procedures for the total strategy using CSM

- Trim Level Control -

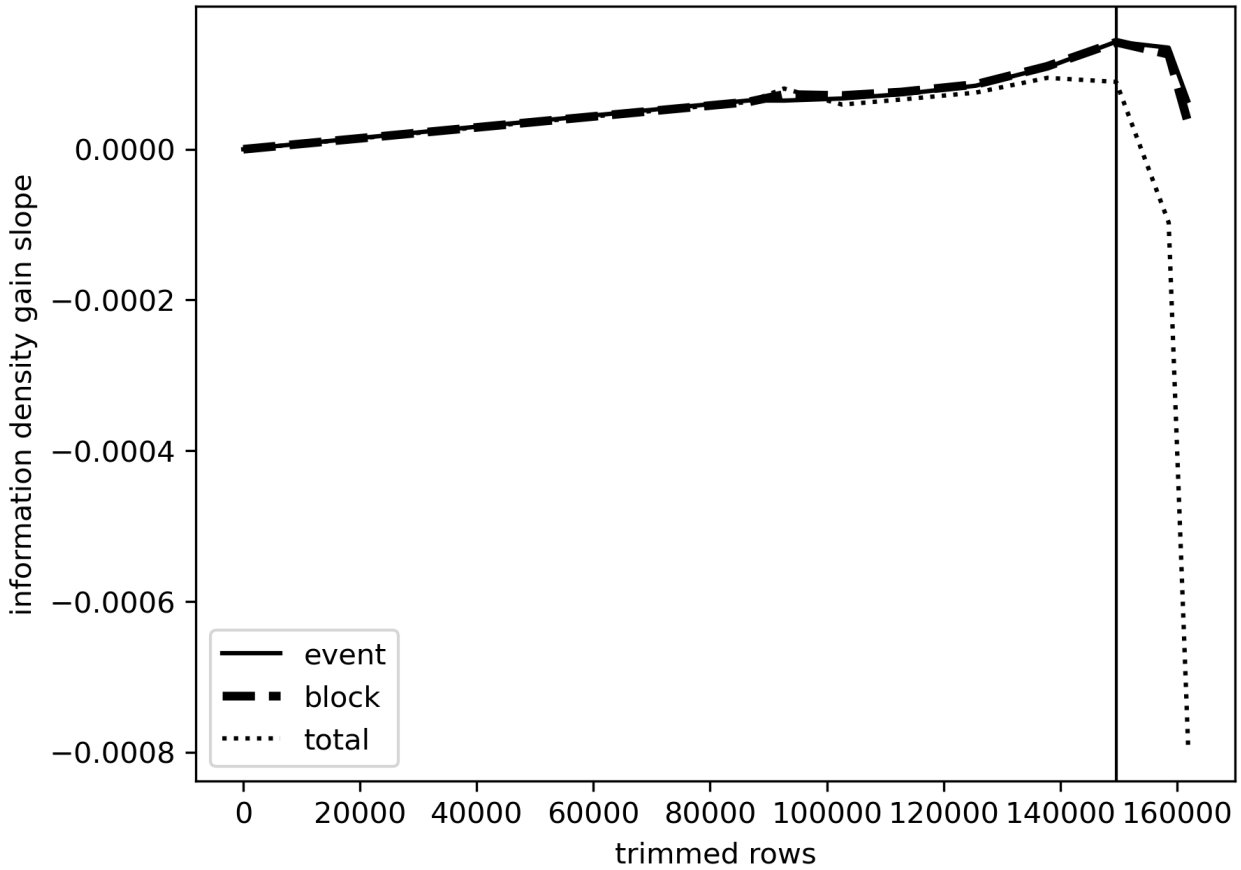


Figure: Information density gain slope for the astride procedure, calculated by the event, block, and total strategies using CSM

- Error Control -

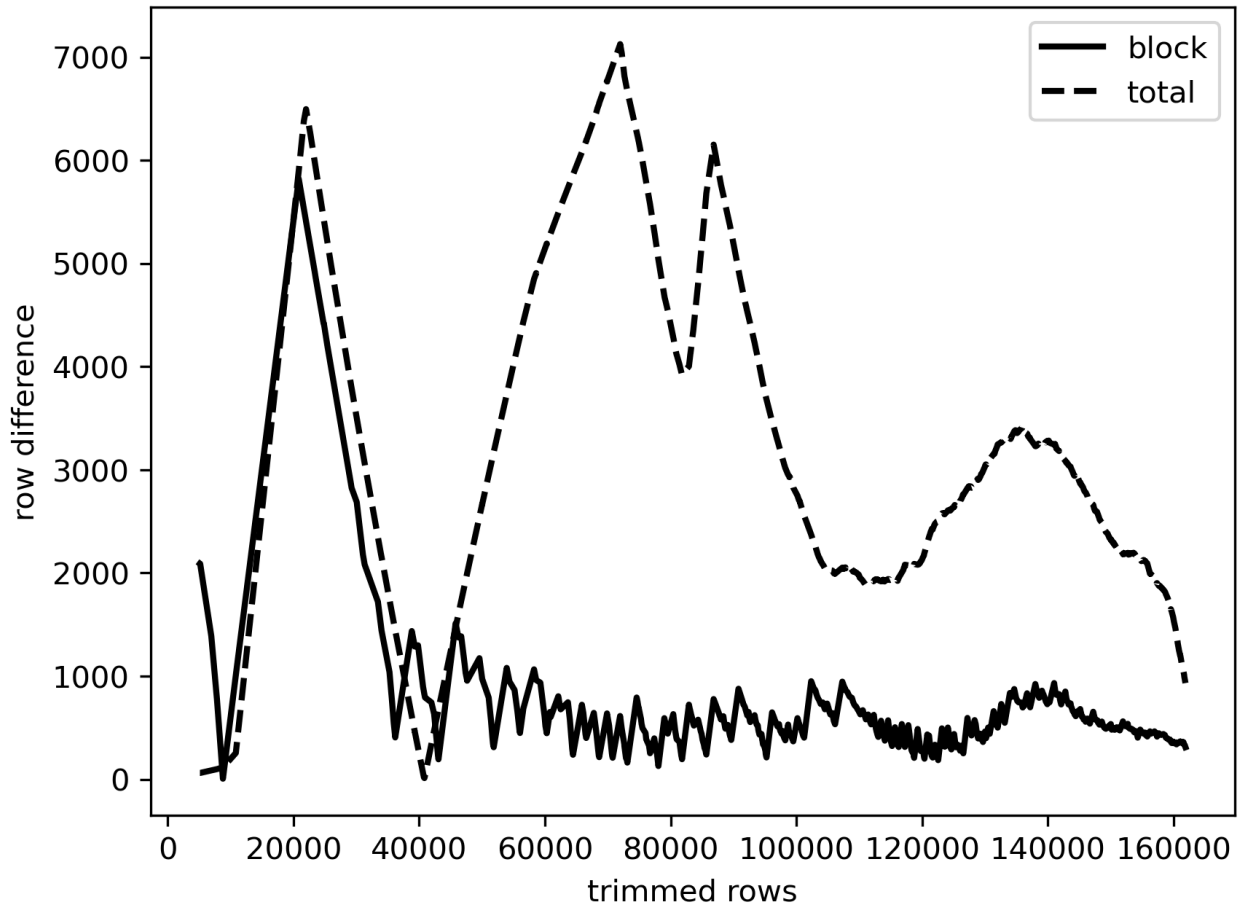


Figure: Number of different rows after block and total strategies compared to event trimming for the astride procedure using IPM

Conclusions

- Modular software for trimming was developed
- Dataset profiling and result visualizing tests were implemented
- Computation speed and error was compared to a conventional greedy algorithm
- The block strategy yields low errors, while speeding up significantly the computation
- Optimum trimming is identified as the inflection point of the information density function as the trimming level is increased.

Conclusions

- Modular software for trimming was developed
- Dataset profiling and result visualizing tests were implemented
- Computation speed and error was compared to a conventional greedy algorithm
- The block strategy yields low errors, while speeding up significantly the computation
- Optimum trimming is identified as the inflection point of the information density function as the trimming level is increased.

Conclusions

- Modular software for trimming was developed
- Dataset profiling and result visualizing tests were implemented
- Computation speed and error was compared to a conventional greedy algorithm
- The block strategy yields low errors, while speeding up significantly the computation
- Optimum trimming is identified as the inflection point of the information density function as the trimming level is increased.

Conclusions

- Modular software for trimming was developed
- Dataset profiling and result visualizing tests were implemented
- Computation speed and error was compared to a conventional greedy algorithm
- The block strategy yields low errors, while speeding up significantly the computation
- Optimum trimming is identified as the inflection point of the information density function as the trimming level is increased.

Conclusions

- Modular software for trimming was developed
- Dataset profiling and result visualizing tests were implemented
- Computation speed and error was compared to a conventional greedy algorithm
- The block strategy yields low errors, while speeding up significantly the computation
- Optimum trimming is identified as the inflection point of the information density function as the trimming level is increased.

Questions

Thank You!

Questions?

Questions

Thank You!

Questions?