Information Models for Large Table Trimming

F. B. Manolache[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

- Motivation -

Trimming large datasets:

- Slow web backends that deal with live data aquisition
- ML sampling of training data
- anomaly detection/filtering

Requirements:

- Reduce the number of rows as much as possible
- Keep as much information as possible

Result: best compromise between size and retained information

- Motivation -

Trimming large datasets:

- Slow web backends that deal with live data aquisition
- ML sampling of training data
- anomaly detection/filtering

Requirements:

- Reduce the number of rows as much as possible
- Keep as much information as possible

Result: best compromise between size and retained information

- Motivation -

Trimming large datasets:

- Slow web backends that deal with live data aquisition
- ML sampling of training data
- anomaly detection/filtering

Requirements:

- Reduce the number of rows as much as possible
- Keep as much information as possible

Result: best compromise between size and retained information

- Motivation -

Trimming large datasets:

- Slow web backends that deal with live data aquisition
- ML sampling of training data
- anomaly detection/filtering

Requirements:

- Reduce the number of rows as much as possible
- Keep as much information as possible

Result: best compromise between size and retained information

- Motivation -

Trimming large datasets:

- Slow web backends that deal with live data aquisition
- ML sampling of training data
- anomaly detection/filtering

Requirements:

- Reduce the number of rows as much as possible
- Keep as much information as possible

Result: best compromise between size and retained information

- Motivation -

Trimming large datasets:

- Slow web backends that deal with live data aquisition
- ML sampling of training data
- anomaly detection/filtering

Requirements:

- Reduce the number of rows as much as possible
- Keep as much information as possible

Result: best compromise between size and retained information

- Concepts -

Model:

- Information function: what we value
- Trimming procedure: order of discarding

Dataset:

- Classes: include data with the same meaning (table columns)
- Entries: one piece of data (table cell)
- Timestamps: mark the order (and frequency) of rows
- Events: data captured at the same time, hence placed in the same row (table row = timestamp + event)
- Correlations: common patterns in entries → cross-referencing

Time symmetry: attention factor

- neutral: older and newer events are equally important
- elevated: older events are less important than newer events
- low: older events are more important than newer events

- Concepts -

Model:

- Information function: what we value
- Trimming procedure: order of discarding

Dataset:

- Classes: include data with the same meaning (table columns)
- Entries: one piece of data (table cell)
- Timestamps: mark the order (and frequency) of rows
- Events: data captured at the same time, hence placed in the same row (table row = timestamp + event)
- Correlations: common patterns in entries → cross-referencing

Time symmetry: attention factor

- neutral: older and newer events are equally important
- elevated: older events are less important than newer events
- low: older events are more important than newer events

- Concepts -

Model:

- Information function: what we value
- Trimming procedure: order of discarding

Dataset:

- Classes: include data with the same meaning (table columns)
- Entries: one piece of data (table cell)
- Timestamps: mark the order (and frequency) of rows
- Events: data captured at the same time, hence placed in the same row (table row = timestamp + event)
- Correlations: common patterns in entries → cross-referencing

Time symmetry: attention factor

- neutral: older and newer events are equally important
- elevated: older events are less important than newer events
- low: older events are more important than newer events

- Concepts -

Model:

- Information function: what we value
- Trimming procedure: order of discarding

Dataset:

- Classes: include data with the same meaning (table columns)
- Entries: one piece of data (table cell)
- Timestamps: mark the order (and frequency) of rows
- Events: data captured at the same time, hence placed in the same row (table row = timestamp + event)
- Correlations: common patterns in entries → cross-referencing

Time symmetry: attention factor

- neutral: older and newer events are equally important
- elevated: older events are less important than newer events
- low: older events are more important than newer events

- Concepts -

Model:

- Information function: what we value
- Trimming procedure: order of discarding

Dataset:

- Classes: include data with the same meaning (table columns)
- Entries: one piece of data (table cell)
- Timestamps: mark the order (and frequency) of rows
- Events: data captured at the same time, hence placed in the same row (table row = timestamp + event)
- Correlations: common patterns in entries → cross-referencing

Time symmetry: attention factor

- neutral: older and newer events are equally important
- elevated: older events are less important than newer events
- low: older events are more important than newer events

- Concepts -

Model:

- Information function: what we value
- Trimming procedure: order of discarding

Dataset:

- Classes: include data with the same meaning (table columns)
- Entries: one piece of data (table cell)
- Timestamps: mark the order (and frequency) of rows
- Events: data captured at the same time, hence placed in the same row (table row = timestamp + event)
- Correlations: common patterns in entries → cross-referencing

Time symmetry: attention factor

- neutral: older and newer events are equally important
- elevated: older events are less important than newer events
- low: older events are more important than newer events

- Concepts -

Model:

- Information function: what we value
- Trimming procedure: order of discarding

Dataset:

- Classes: include data with the same meaning (table columns)
- Entries: one piece of data (table cell)
- Timestamps: mark the order (and frequency) of rows
- Events: data captured at the same time, hence placed in the same row (table row = timestamp + event)
- Correlations: common patterns in entries → cross-referencing

Time symmetry: attention factor

- neutral: older and newer events are equally important
- elevated: older events are less important than newer events
- low: older events are more important than newer events

- Concepts -

Model:

- Information function: what we value
- Trimming procedure: order of discarding

Dataset:

- Classes: include data with the same meaning (table columns)
- Entries: one piece of data (table cell)
- Timestamps: mark the order (and frequency) of rows
- Events: data captured at the same time, hence placed in the same row (table row = timestamp + event)
- Correlations: common patterns in entries $\rightarrow$ cross-referencing

Time symmetry: attention factor

- neutral: older and newer events are equally important
- elevated: older events are less important than newer events
- low: older events are more important than newer events

- Concepts -

Model:

- Information function: what we value
- Trimming procedure: order of discarding

Dataset:

- Classes: include data with the same meaning (table columns)
- Entries: one piece of data (table cell)
- Timestamps: mark the order (and frequency) of rows
- Events: data captured at the same time, hence placed in the same row (table row = timestamp + event)
- Correlations: common patterns in entries $\rightarrow$ cross-referencing

Time symmetry: attention factor

- neutral: older and newer events are equally important
- elevated: older events are less important than newer events
- low: older events are more important than newer events

- Concepts -

Model:

- Information function: what we value
- Trimming procedure: order of discarding

Dataset:

- Classes: include data with the same meaning (table columns)
- Entries: one piece of data (table cell)
- Timestamps: mark the order (and frequency) of rows
- Events: data captured at the same time, hence placed in the same row (table row = timestamp + event)
- Correlations: common patterns in entries → cross-referencing

Time symmetry: attention factor

- neutral: older and newer events are equally important
- elevated: older events are less important than newer events
- low: older events are more important than newer events

- Assumptions -

Information function:

- non-negative value for any amount of data
- adding events does not decrease the information; trimming events does not increase the information
- trimming the lowest information event results in minimum decrease the dataset information

Conclusion: total information is a sum-like function of event information

Cross-referencing: binary match (identical/different)

- Assumptions -

Information function:

- non-negative value for any amount of data
- adding events does not decrease the information; trimming events does not increase the information
- trimming the lowest information event results in minimum decrease the dataset information

Conclusion: total information is a sum-like function of event information

Cross-referencing: binary match (identical/different)

- Assumptions -

Information function:

- non-negative value for any amount of data
- adding events does not decrease the information; trimming events does not increase the information
- trimming the lowest information event results in minimum decrease the dataset information

Conclusion: total information is a sum-like function of event information

Cross-referencing: binary match (identical/different)

- Assumptions -

Information function:

- non-negative value for any amount of data
- adding events does not decrease the information; trimming events does not increase the information
- trimming the lowest information event results in minimum decrease the dataset information

Conclusion: total information is a sum-like function of event information

Cross-referencing: binary match (identical/different)

- Assumptions -

Information function:

- non-negative value for any amount of data
- adding events does not decrease the information; trimming events does not increase the information
- trimming the lowest information event results in minimum decrease the dataset information

Conclusion: total information is a sum-like function of event information

Cross-referencing: binary match (identical/different)

- Strategies -

Combinatorial subset optimization problem:

- exact solution is **impractically slow** for large datasets
- practical solution: greedy algorithms and derived refinements

Greedy algorithm accelerators: **trimming strategy**:

- event trimming: discards the event with lowest information in each iteration, then recalculates the contributions; slow; good memory
- block trimming: discards some of the events with low information in each iteration, then recalculates the contributions; faster; minimal memory loss
- total trimming: discards all events with the lowest information in one pass; very fast; no memory

- Strategies -

Combinatorial subset optimization problem:

- exact solution is **impractically slow** for large datasets
- practical solution: greedy algorithms and derived refinements

Greedy algorithm accelerators: **trimming strategy**:

- event trimming: discards the event with lowest information in each iteration, then recalculates the contributions; slow; good memory
- block trimming: discards some of the events with low information in each iteration, then recalculates the contributions; faster; minimal memory loss
- total trimming: discards all events with the lowest information in one pass; very fast; no memory

- Strategies -

Combinatorial subset optimization problem:

- exact solution is **impractically slow** for large datasets
- practical solution: greedy algorithms and derived refinements

Greedy algorithm accelerators: **trimming strategy**:

- event trimming: discards the event with lowest information in each iteration, then recalculates the contributions; slow; good memory
- block trimming: discards some of the events with low information in each iteration, then recalculates the contributions; faster; minimal memory loss
- total trimming: discards all events with the lowest information in one pass; very fast; no memory

- Strategies -

Combinatorial subset optimization problem:

- exact solution is **impractically slow** for large datasets
- practical solution: greedy algorithms and derived refinements

Greedy algorithm accelerators: **trimming strategy**:

- event trimming: discards the event with lowest information in each iteration, then recalculates the contributions; slow; good memory
- block trimming: discards some of the events with low information in each iteration, then recalculates the contributions; faster; minimal memory loss
- total trimming: discards all events with the lowest information in one pass; very fast; no memory

- Strategies -

Combinatorial subset optimization problem:

- exact solution is **impractically slow** for large datasets
- practical solution: greedy algorithms and derived refinements

Greedy algorithm accelerators: **trimming strategy**:

- event trimming: discards the event with lowest information in each iteration, then recalculates the contributions; slow; good memory
- block trimming: discards some of the events with low information in each iteration, then recalculates the contributions; faster; minimal memory loss
- total trimming: discards all events with the lowest information in one pass; very fast; no memory

- The Independent Event Models -

- Trimming one or multiple events does not change the order of the information contribution of the other events
- Due to introduction of procedures, information contribution of each event $= 1$
- Block trimming strategy is all it takes

- The Independent Event Models -

- Trimming one or multiple events does not change the order of the information contribution of the other events
- Due to introduction of procedures, information contribution of each event $= 1$
- Block trimming strategy is all it takes

- The Independent Event Models -

- Trimming one or multiple events does not change the order of the information contribution of the other events
- Due to introduction of procedures, information contribution of each event $= 1$
- Block trimming strategy is all it takes

- The Event Statistical Model -

- Based on Shannon information applied to entire events
- Assumes that the data is ergodic (all the events are equally important), so the timestamp column is ignored by the information function
- Ergodicity can be broken by the procedure
- Block trimming can be used without introducing errors

- The Event Statistical Model -

- Based on Shannon information applied to entire events
- Assumes that the data is ergodic (all the events are equally important), so the timestamp column is ignored by the information function
- Ergodicity can be broken by the procedure
- Block trimming can be used without introducing errors

- The Event Statistical Model -

- Based on Shannon information applied to entire events
- Assumes that the data is ergodic (all the events are equally important), so the timestamp column is ignored by the information function
- Ergodicity can be broken by the procedure
- Block trimming can be used without introducing errors

- The Event Statistical Model -

- Based on Shannon information applied to entire events
- Assumes that the data is ergodic (all the events are equally important), so the timestamp column is ignored by the information function
- Ergodicity can be broken by the procedure
- Block trimming can be used without introducing errors

- The Class Statistical Model -

- Shannon information is applied at entry (table cell) level
- Cross-referencing weights are introduced to Shannon information to emphasize entries in one class (column) appearing also in other classes
- Event information is a sum of entry contributions
- A block trimming strategy is developed based on KMeans clustering low information contribution events in blocks and eliminating them in one iteration
- Timestamp information still not captured by the information function

- The Class Statistical Model -

- Shannon information is applied at entry (table cell) level
- Cross-referencing weights are introduced to Shannon information to emphasize entries in one class (column) appearing also in other classes
- Event information is a sum of entry contributions
- A block trimming strategy is developed based on KMeans clustering low information contribution events in blocks and eliminating them in one iteration
- Timestamp information still not captured by the information function

- The Class Statistical Model -

- Shannon information is applied at entry (table cell) level
- Cross-referencing weights are introduced to Shannon information to emphasize entries in one class (column) appearing also in other classes
- Event information is a sum of entry contributions
- A block trimming strategy is developed based on KMeans clustering low information contribution events in blocks and eliminating them in one iteration
- Timestamp information still not captured by the information function

- The Class Statistical Model -

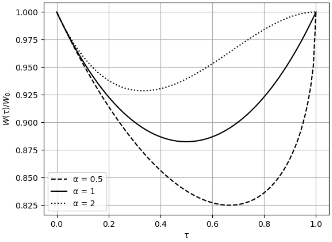
- Shannon information is applied at entry (table cell) level
- Cross-referencing weights are introduced to Shannon information to emphasize entries in one class (column) appearing also in other classes
- Event information is a sum of entry contributions
- A block trimming strategy is developed based on KMeans clustering low information contribution events in blocks and eliminating them in one iteration
- Timestamp information still not captured by the information function

- The Class Statistical Model -

- Shannon information is applied at entry (table cell) level
- Cross-referencing weights are introduced to Shannon information to emphasize entries in one class (column) appearing also in other classes
- Event information is a sum of entry contributions
- A block trimming strategy is developed based on KMeans clustering low information contribution events in blocks and eliminating them in one iteration
- Timestamp information still not captured by the information function

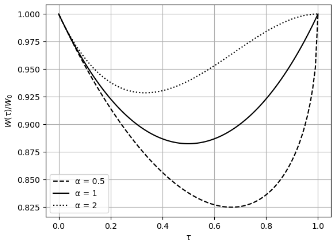
- The Ideal perception Model -

- Development of the Class Statistical Model with additional weights to account for timestamp distribution of repeated appearences of the same value in a class
- Timestamp distribution weights are calculated to mimic human perception (forgetting, memory reinforcement by repetition)



- The Ideal perception Model -

- Development of the Class Statistical Model with additional weights to account for timestamp distribution of repeated appearences of the same value in a class
- Timestamp distribution weights are calculated to mimic human perception (forgetting, memory reinforcement by repetition)



Conclusions

- Trimming should capture the relevant information in the dataset via a model.
- Compromise between the amount of trimming and the amount of information loss should be achieved.
- Information models structure should mimic the data consumption purpose.
- A computation speed-up can be obtained by using various trimming strategies with the greedy algorithm.

Conclusions

- Trimming should capture the relevant information in the dataset via a model.
- Compromise between the amount of trimming and the amount of information loss should be achieved.
- Information models structure should mimic the data consumption purpose.
- A computation speed-up can be obtained by using various trimming strategies with the greedy algorithm.

Conclusions

- Trimming should capture the relevant information in the dataset via a model.
- Compromise between the amount of trimming and the amount of information loss should be achieved.
- Information models structure should mimic the data consumption purpose.
- A computation speed-up can be obtained by using various trimming strategies with the greedy algorithm.

Conclusions

- Trimming should capture the relevant information in the dataset via a model.
- Compromise between the amount of trimming and the amount of information loss should be achieved.
- Information models structure should mimic the data consumption purpose.
- A computation speed-up can be obtained by using various trimming strategies with the greedy algorithm.

Questions

Thank You!

Questions?

Questions

Thank You!

Questions?