Progressive Multidimensional Projections

A Process Model based on Vector Quantization

E. Ventocilla, R. M. Martins, F. Paulovich, M. Riveiro

UNIVERSITY OF SKÖVDE

Linnéuniversitetet
Linnaeus University

DALHOUSIE UNIVERSITY

JÖNKÖPING UNIVERSITY

Progressive Multidimensional Projections

A Process Model based on Vector Quantization

Elio Ventocilla, R. M. Martins, F. Paulovich, M. Riveiro

Presenter

UNIVERSITY OF SKÖVDE

Linnéuniversitetet
Linnaeus University

DALHOUSIE UNIVERSITY

JÖNKÖPING UNIVERSITY

# of features

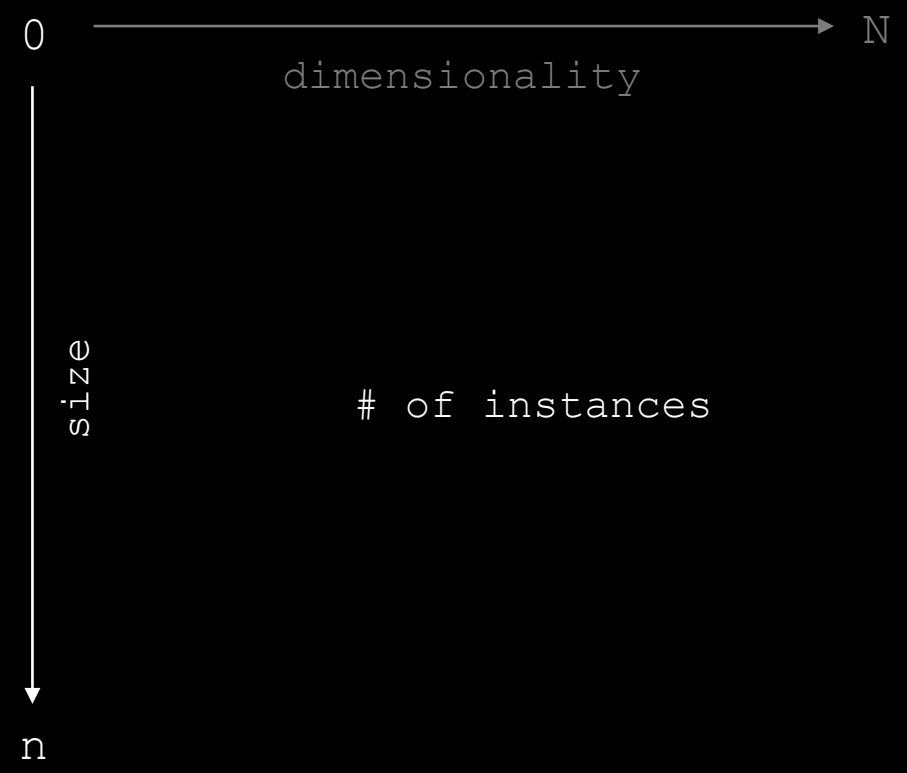
# of instances

The curse of dimensionality [AHK01]

- More sparsity
- Less meaningful distance relations
- Less meaningful data structure visualizations

Lots of related work, e.g.:

- MDS (1958)
- PCA (1986)
- LLE (2000)
- Isomap (2003)
- LSP (2008)
- t-SNE (2008)
- LAMP (2011)
- UMAP (2018)



Three main (usability) challenges

Time to visual feedback
How long it takes for a system to provide visual feedback to the user.

Visual cluttering
Overlapping elements in the visualization

View interactiveness
Capabilities that a view has for user interactions with fast visual feedback, e.g., brushing and linking

We propose

Using **incremental Vector Quantization (iVQ)** techniques, as a pre-step to **multidimensional projections (MDP)**

iVQ, techniques that:

- compress data to a given number of prototypes
- work on batches of data
- continuously improves its representation of the distance relations of the original data

MDPs:

- DR, e.g. PCA, MDS, t-SNE, UMAP
- Clustering, e.g., SOM, Ward, OPTICS, GNG.

Design requirements

An extended list of design requirements for P-MDP that addresses the outlined usability constraints.

Process model

A process model that enables P-MDPs for large datasets through iVQ, and outlines which elements

(a) enable different types of user involvements

(b) address the design requirements

Prototype

that scales to distributed datasets, and illustrates the flexibility of the model as well as its validity in terms of the design requirements

Time to visual feedback

How long it takes for a system to provide visual feedback to the user.

Visual cluttering

Overlapping elements in the visualization

View interactiveness

Capabilities that a view has for user interactions with fast visual feedback, e.g., brushing and linking

Improving performance

E.g., FastMap (1995), Glimmer (2008), BH t-SNE (2014), t-SNE-CUDA (2019), GPGPU (2020).

Progressive training

Stolper et al. (2014), Mühlbacher et al. (2014), Schulz et al. (2016), Fekete et al. (2016)

Time to visual feedback

How long it takes for a system to provide visual feedback to the user.

Visual cluttering

Overlapping elements in the visualization

View interactiveness

Capabilities that a view has for user interactions with fast visual feedback, e.g., brushing and linking

Visual techniques

- Opacity and contours, e.g., Pezzoti et al. (2017)
- Edge bundling, e.g., (Zhou et al., 2008; Liu et al., 2017)
- Surfaces, e.g., (Poco et al., 2012)

Compression

Only a limited amount of elements are visually encoded by using, e.g.:

- CCA (Demartines, 1997)
- SOM (Riveiro et al., 2008)
- HiPP (Paulovich et al, 2008)
- HSNE (Höllt, 2019)
- GNG (Ventocilla et al., 2019)

Time to visual feedback

How long it takes for a system to provide visual feedback to the user.

Visual cluttering

Overlapping elements in the visualization

View interactiveness

Capabilities that a view has for user interactions with fast visual feedback, e.g., brushing and linking

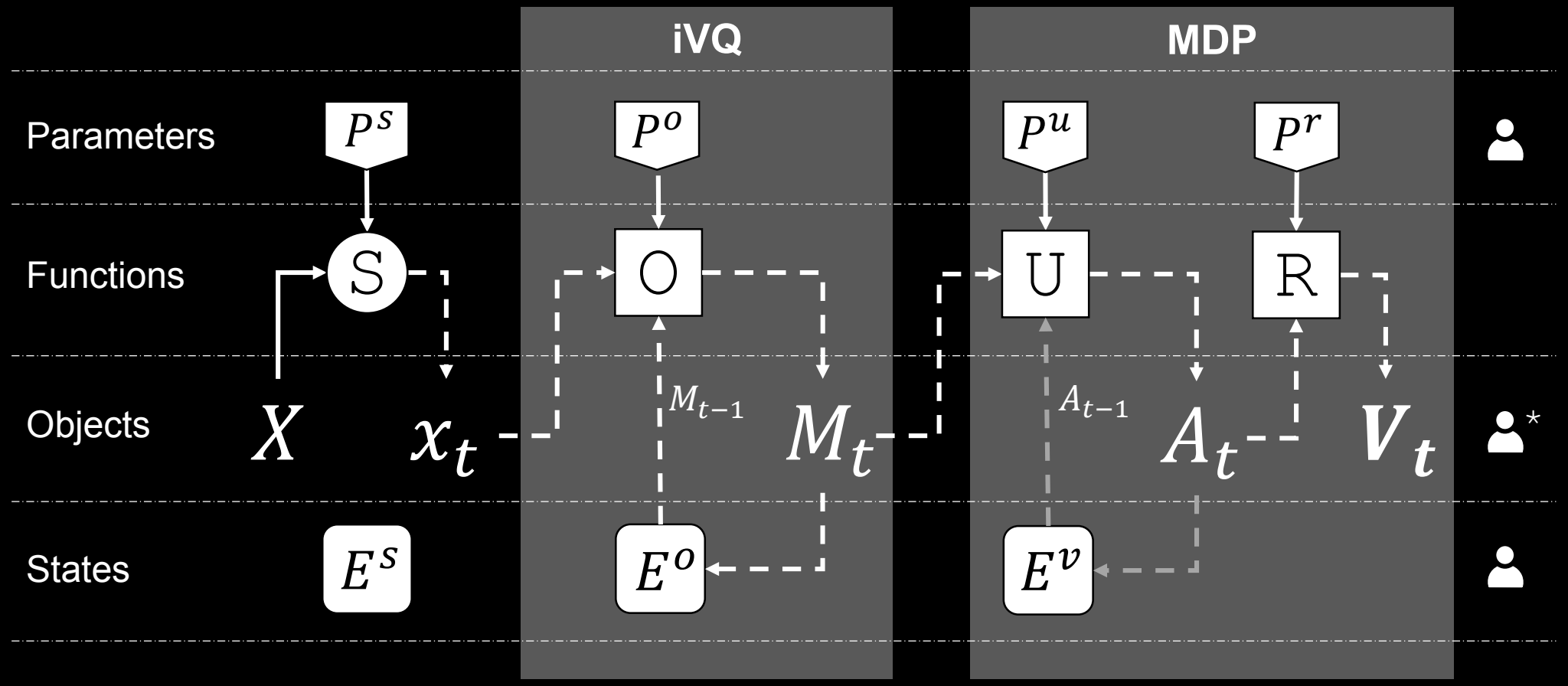
Interactions with visual feedback at continuity-preserving latency (i.e., < .1 seconds)

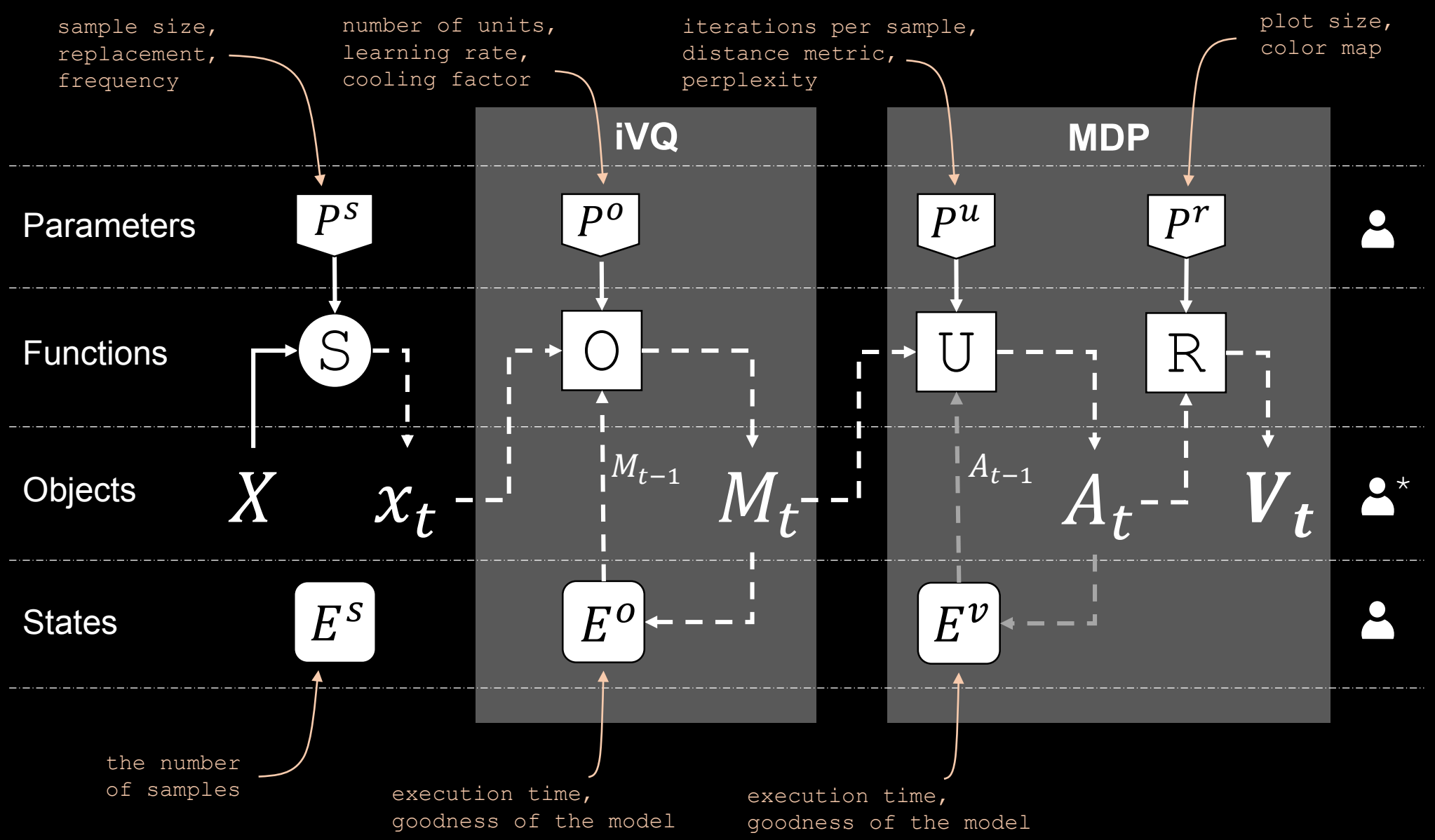
Progressive MDPs should (Fekete et al., 2016):

- **F1**. provide increasingly meaningful partial results.
- **F2**. provide feedback about the state of the computation.
- **F3**. provide control over the process.
- **F4**. guarantee feedback time constraints (mainly to attention preserving latency, i.e., **< 10s**).
- **F5**. allow manipulation of progressive values.
- **F6**. allow steering through parameters or other computation components.
- **F7**. allow performing exploratory, analytical computations.

and should also

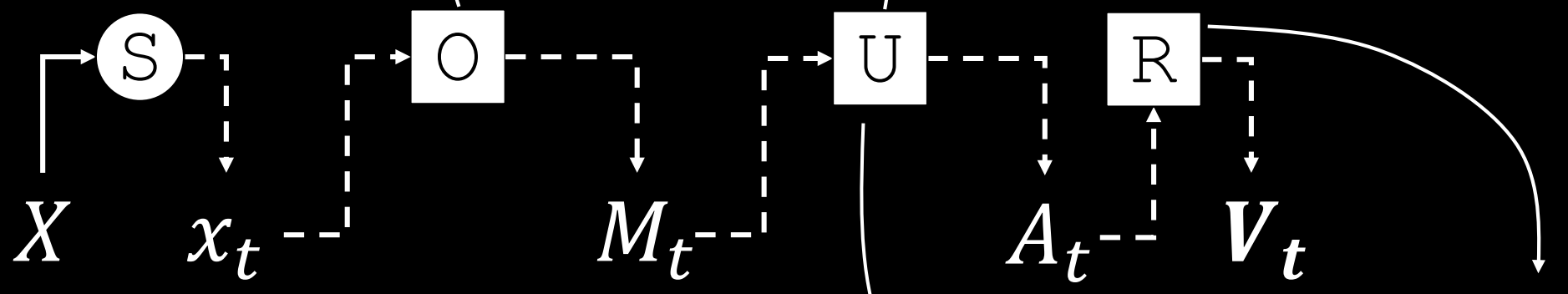
- **F8**. provide an overview of the data structure, while avoiding visual clutter.
- **F9**. maintain view interactiveness at a continuity preserving latency (< .1 seconds).
- **F10**. allow users to navigate across different levels of detail.


iVQ
MDP
Parameters
Functions
Objects
States
P^s
P^o
P^u
P^r
S
O
U
R
X
x_t
M_{t-1}
M_t
A_{t-1}
A_t
V_t
E^s
E^o
E^v


sample size, replacement, frequency
number of units, learning rate, cooling factor
iterations per sample, distance metric, perplexity
plot size, color map
iVQ
MDP
Parameters
Functions
Objects
States
P^s
P^o
P^u
P^r
S
O
U
R
X
x_t
M_{t-1}
M_t
A_{t-1}
A_t
V_t
E^s
E^o
E^v
the number of samples
execution time, goodness of the model
execution time, goodness of the model

Neural based: SOM, GNG
Partition-based: Mini batch K-Means
Hierarchy-based: BIRCH
Density-based: D-Stream

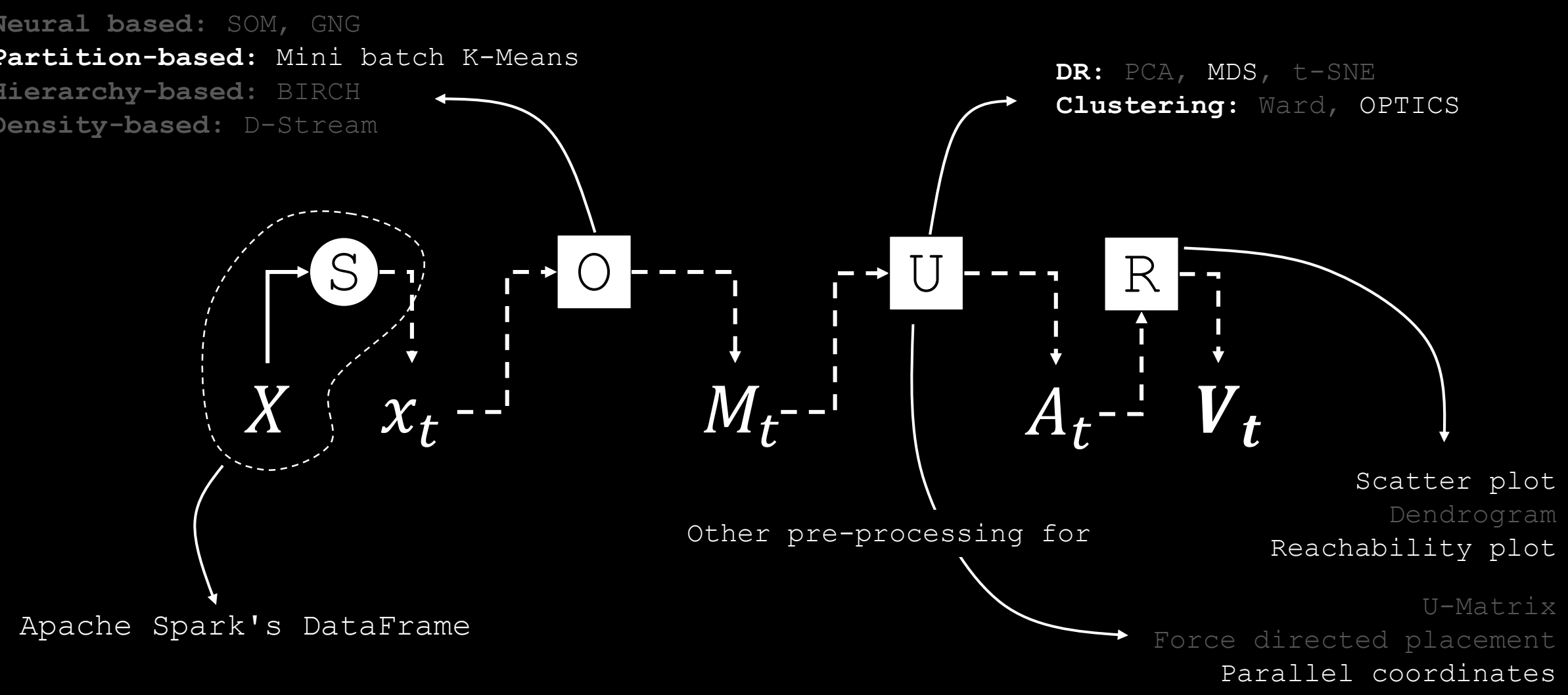
DR: PCA, MDS, t-SNE
Clustering: Ward, OPTICS



Other pre-processing for

Scatter plot
Dendrogram
Reachability plot

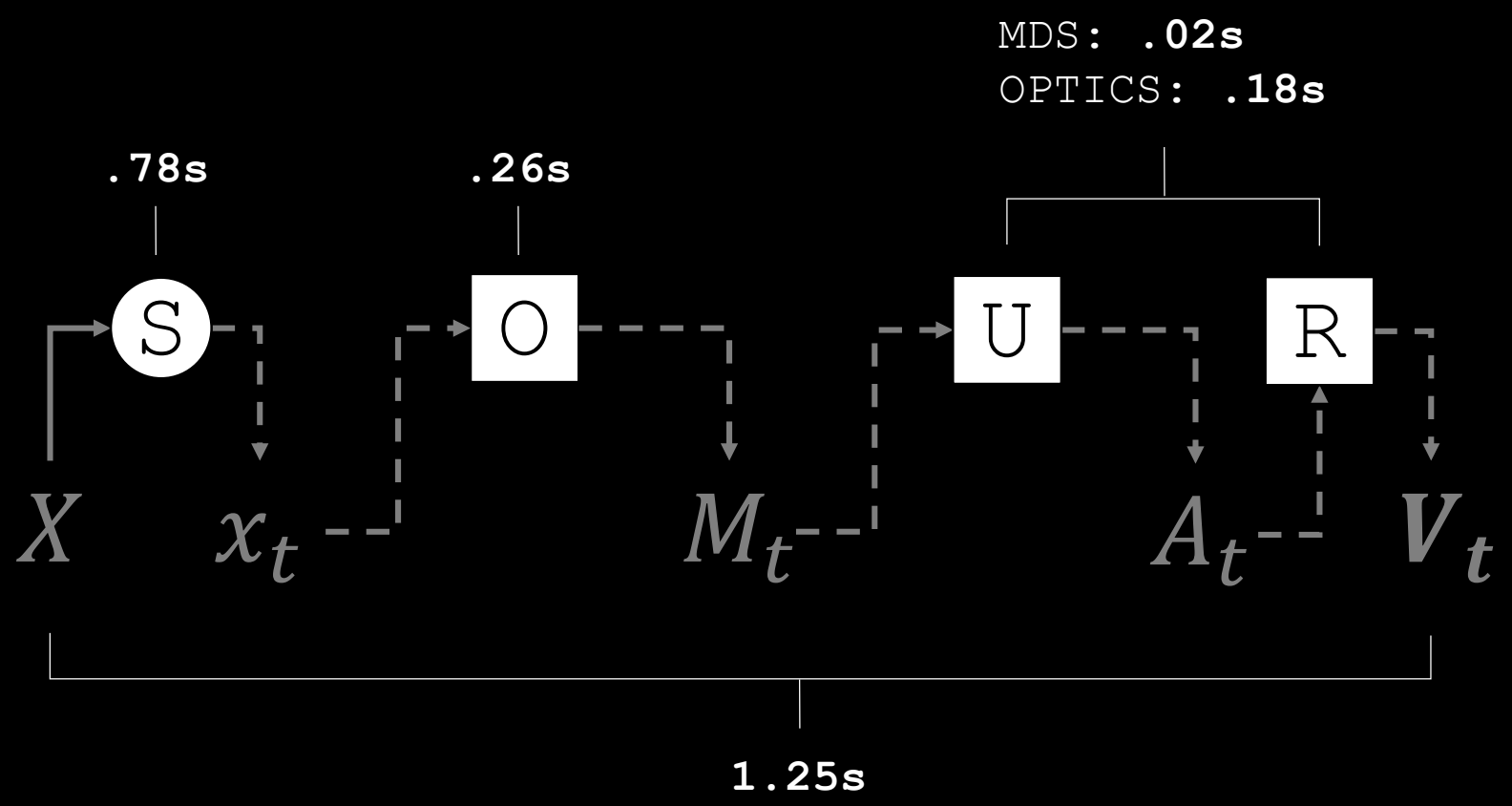
U-Matrix
Force directed placement
Parallel coordinates


Neural based: SOM, GNG
Partition-based: Mini batch K-Means
Hierarchy-based: BIRCH
Density-based: D-Stream
DR: PCA, MDS, t-SNE
Clustering: Ward, OPTICS
S
O
U
R
X
x_t
M_t
A_t
V_t
Scatter plot
Dendrogram
Reachability plot
Other pre-processing for
U-Matrix
Force directed placement
Parallel coordinates
Apache Spark's DataFrame


Reset
Run
Sampler
Avg time: 0.97s | samples: 3
Quantizer
Avg time: 1.11s | inertia: 296
Avg time (all): 3.04s
OPTICS
Avg time: 0.28s
MDS
Avg time: 0.06s | stress: 5582
00049497 S00049659 S00049732 S00050876 S00049882 S00049816 S00049969 S00049824 S00049963 S00049822 S00049489
0.9784 0.96761 0.97846 0.97461 0.94606 0.97308 0.93585 0.95191 0.95595 0.93995 0.9381
0.0262 0.01863 0.02335 0.02983 0.0302 0.02163 0.03044 0.02482 0.01898 0.02623 0.02136

Visual feedback
within attention preserving latency (< 10s)

- Samples size = 500
- VQ iterations/sample = 10
- MDP iterations/sample = 100



Visual clutter

provide an overview of the data structure, while avoiding visual clutter

View interactiveness

within continuity preserving latency (< 0.1s)

- Closer to 1s
- Due to how Python Dash handles interaction events

Other considerations

- Handling view interactions while training.
- Providing context while zooming in and out.
- Ensuring time constraints.
- Convergence?
- iVQ versus DR vantage points.
- More parameters to tune.

Future work

Extend model to account for

- Zooming in and out.
- User control through view interactions
- Streaming
- Explanatory techniques, e.g., LIME, BRL

Thank you

Elio Ventocilla, R. M. Martins, F. Paulovich, M. Riveiro

Presenter

UNIVERSITY OF SKÖVDE

Linnéuniversitetet
Linnaeus University

DALHOUSIE UNIVERSITY

JÖNKÖPING UNIVERSITY