Progressive Multidimensional Projections

A Process Model based on Vector Quantization

E. Ventocilla, R. M. Martins, F. Paulovich, M. Riveiro
Progressive Multidimensional Projections

A Process Model based on Vector Quantization

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

Presenter
<table>
<thead>
<tr>
<th>Problem</th>
<th>Contributions</th>
<th>Related work</th>
<th>Design requirements</th>
<th>Process model</th>
<th>Prototype</th>
<th>Discussion</th>
</tr>
</thead>
</table>

Problem Contributions Related work Design requirements Process model Prototype Discussion

0 → N

dimensionality

# 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.
<table>
<thead>
<tr>
<th>Problem</th>
<th>Contributions</th>
<th>Related work</th>
<th>Design requirements</th>
<th>Process model</th>
<th>Prototype</th>
<th>Discussion</th>
</tr>
</thead>
</table>

**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
<table>
<thead>
<tr>
<th>Time to visual feedback</th>
<th>Improving performance</th>
<th>Progressive training</th>
</tr>
</thead>
</table>

**Visual cluttering**
Overlapping elements in the visualization.

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

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

**Visual cluttering**
Overlapping elements in the visualization

**View interactivity**
Capabilities that a view has for user interactions with fast visual feedback, e.g., brushing and linking
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.
- **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.
Problem Contributions Related work Design requirements Process model Prototype Discussion

**iVQ**

Objects: $X$, $x_t$

States: $E^s$

Parameters: $P^s$

Functions: $S$

**MDP**

States: $E^v$

Parameters: $P^u$, $P^r$

Functions: $U$, $R$

Objects: $V_t$, $A_t$, $A_{t-1}$, $M_t$, $M_{t-1}$, $P^o$, $O$

**MLVis 2020.** E. Ventocilla & R. M. Martins & F. Paulovich & M. Riveiro / Progressive Multidimensional Projections: A Process Model
Problem Contributions Related work Design requirements Process model Prototype Discussion

Sample size, replacement, frequency

Number of units, learning rate, cooling factor

Iterations per sample, distance metric, perplexity

Plot size, color map

Parameters $P^s$

Functions $S$

Objects $X \rightarrow x_t$

States $E^s$

Sample size, replacement, frequency

Number of units, learning rate, cooling factor

Iterations per sample, distance metric, perplexity

Plot size, color map

Parameters $P^o$

Functions $O$

Objects $E^o$

States $E^v$

Sample size, replacement, frequency

Number of units, learning rate, cooling factor

Iterations per sample, distance metric, perplexity

Plot size, color map

Parameters $P^u$

Functions $U$

Objects $E^u$

States $E^v$

Sample size, replacement, frequency

Number of units, learning rate, cooling factor

Iterations per sample, distance metric, perplexity

Plot size, color map

Parameters $P^r$

Functions $R$

Objects $E^r$

States $E^v$

Sample size, replacement, frequency

Number of units, learning rate, cooling factor

Iterations per sample, distance metric, perplexity

Plot size, color map
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

Apache Spark's DataFrame

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

Other pre-processing for

Other pre-processing for

Apache Spark's DataFrame

Problem Contributions Related work Design requirements Process model Prototype Discussion

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 interactivity 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