A Pipeline for Tailored Sampling for Progressive Visual Analytics

Marius Hogräfer, Jakob Burkhardt, Hans-Jörg Schulz

Aarhus University, Denmark

Rome, 13th June 2022

AARHUS UNIVERSITY

Contributions

1 A technique

- Propose a tailorable sampling pipeline for PVA

2 A demonstrative use case

- Tailor the progressive sampling

3 A tool

- ProSample, which allows comparing two pipelines side-by-side

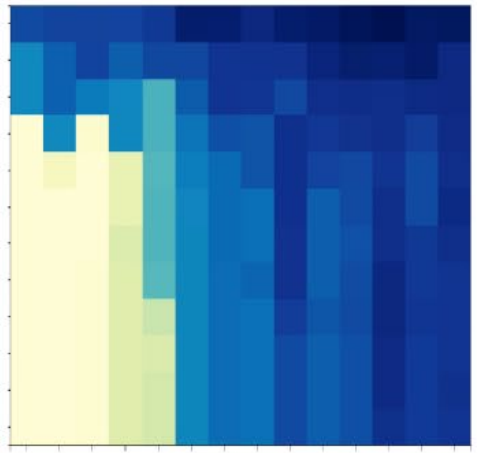
Primer on Progressive Visual Analytics

- Analysis on large data takes too long → **Not interactive!**
- Split the data into smaller chunks
- Enable interactive analysis on **early, partial** results

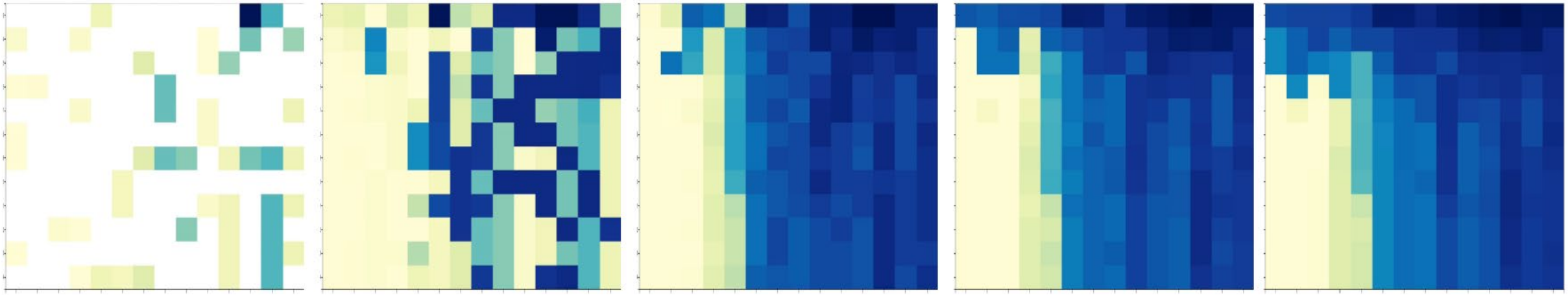
- Analyst gets to see the data they are interested much earlier
- Bring the Human "back into the loop"

Standard approach:

Launch analysis… 1h later (or more!)



PVA approach:

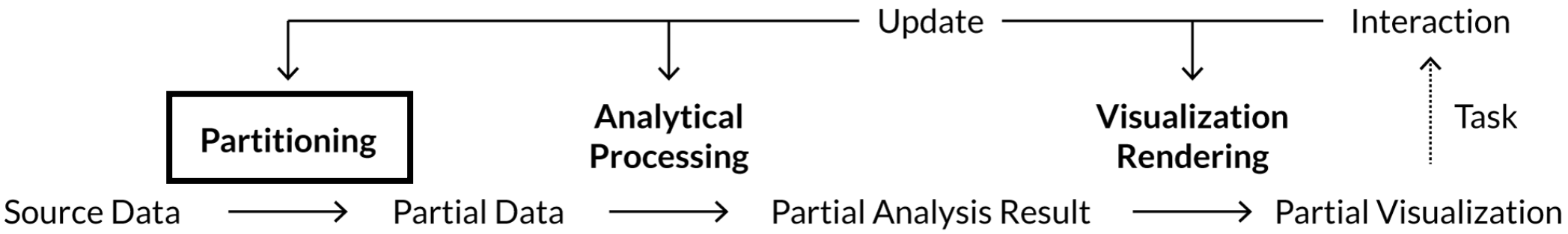


1 sec **10** sec **30** sec **5** min **…** **1** h

Sampling in PVA

- Fundamental to PVA: **Data arrives in chunks**

- The first step in the process:



Adapted from [Li+Ma, 2020]

Challenge

In-progress visualization should be **representative** of the final result

- What makes a chunk representative?
- What order should the data arrive in?

Depends on the analysis scenario!
(see Related Work)

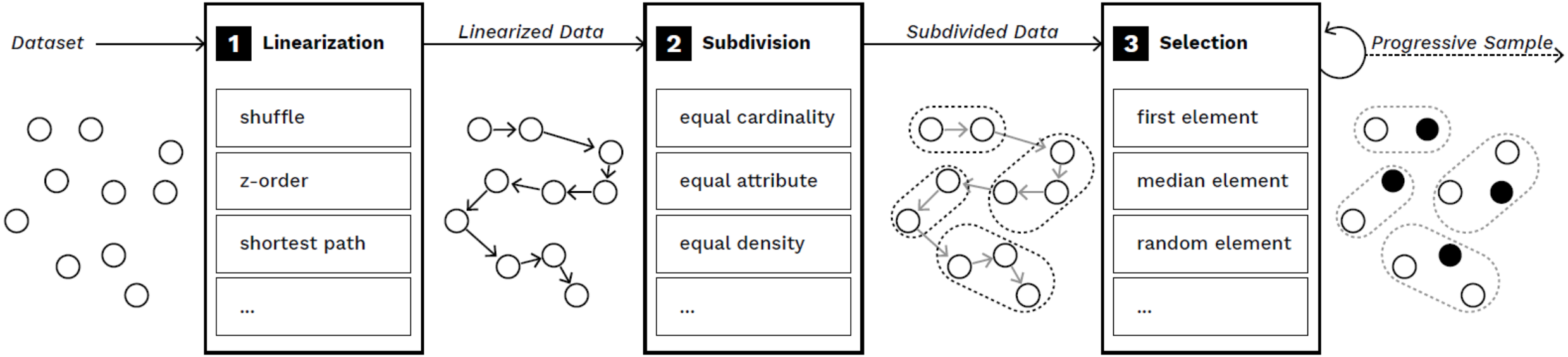
Background

- What to do, when no dedicated algorithm exists?

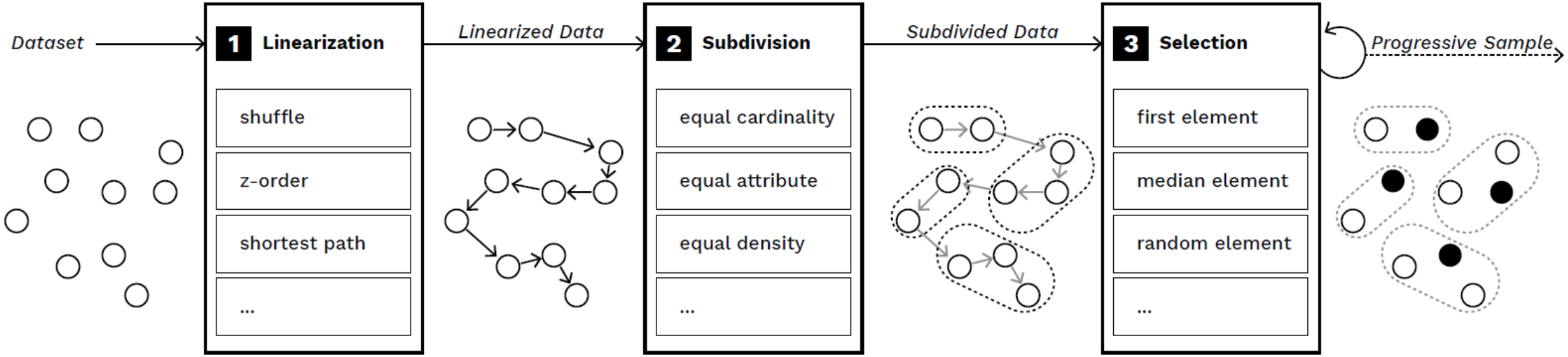
- Fallback is random sampling ("one-size-fits-all")
  - Can produce visual artifacts on some visualizations [Zheng et al. 2017]
  - Poor fit for tasks like outlier detection [Chen et al. 2022]

Our idea: break up sampling process to modularize it
→ allows to **tailor** it to analysis scenarios

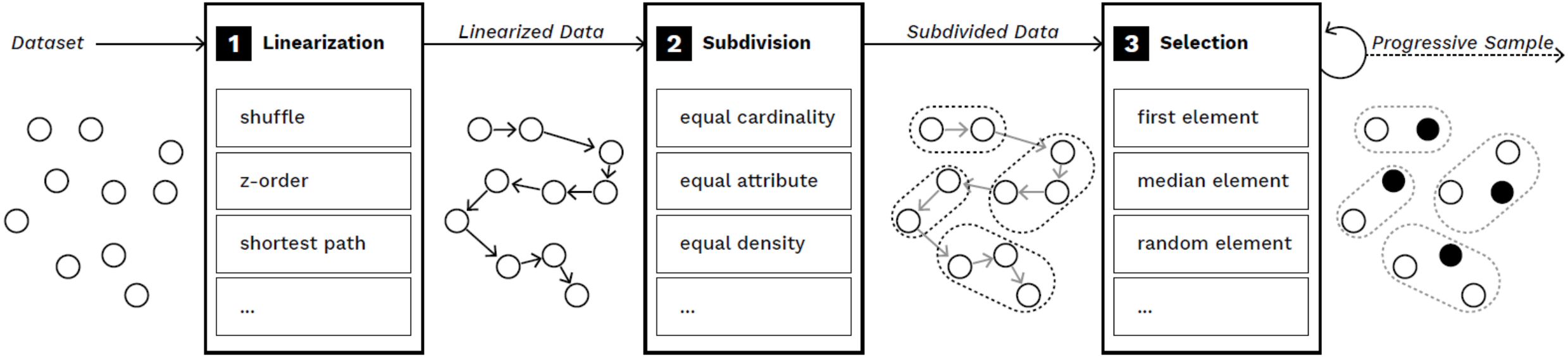
A tailorable Sampling Pipeline for PVA



A tailorable Sampling Pipeline for PVA



How does the pipeline work?



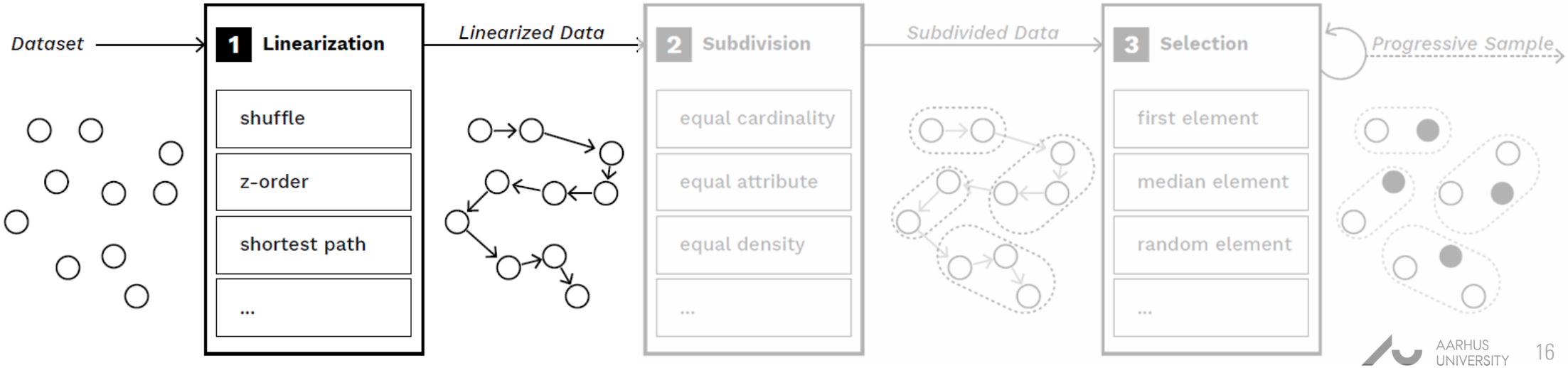
Harmonize input data into linear list

Divide the list into smaller list

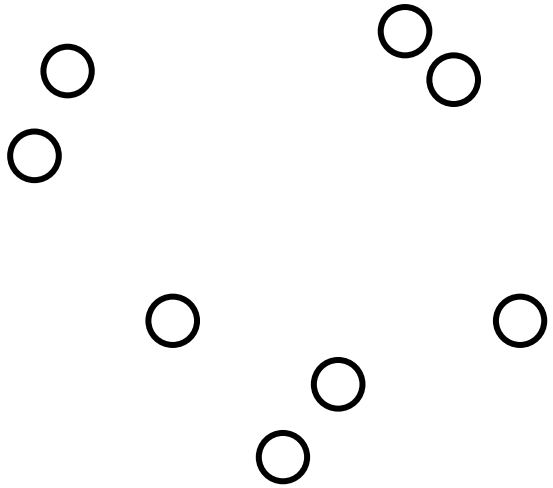
Select items from each list

Linearization

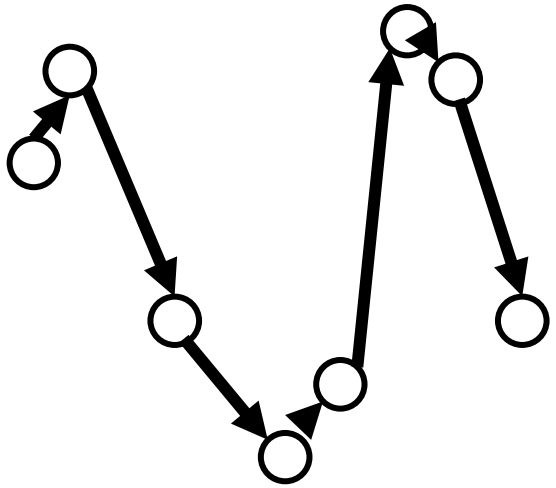
- Input: Dataset
- Output: List
- Tailor to **characteristics of the dataset**
- Based on data structure



Linearization - Example

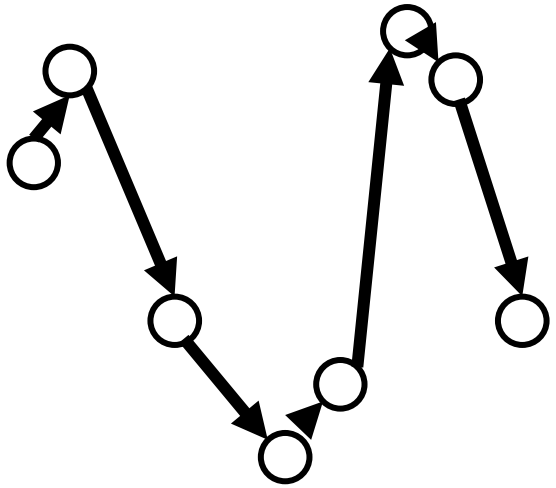


Linearization - Example

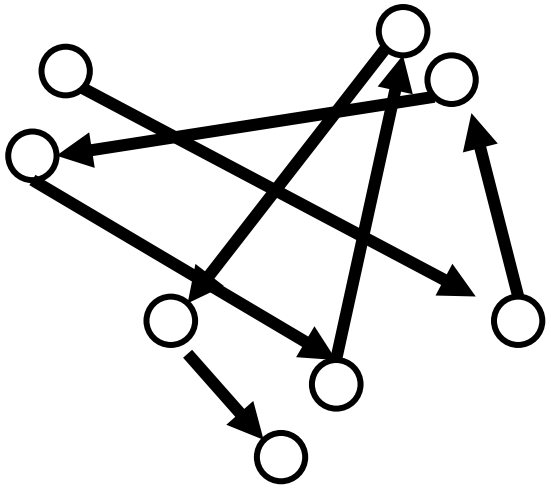


Sort by attribute

Linearization - Example

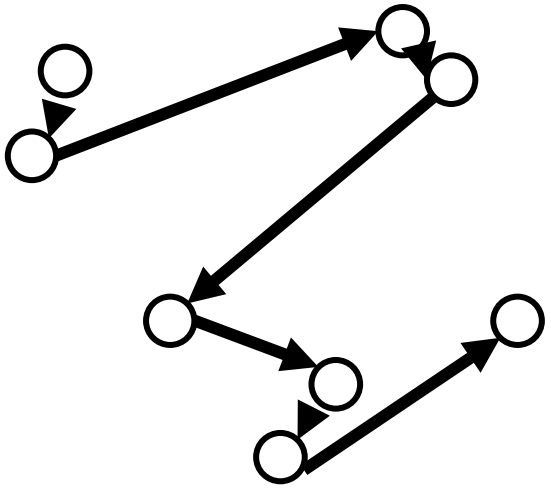


Sort by attribute



Shuffle

[Badam et al. 2013]

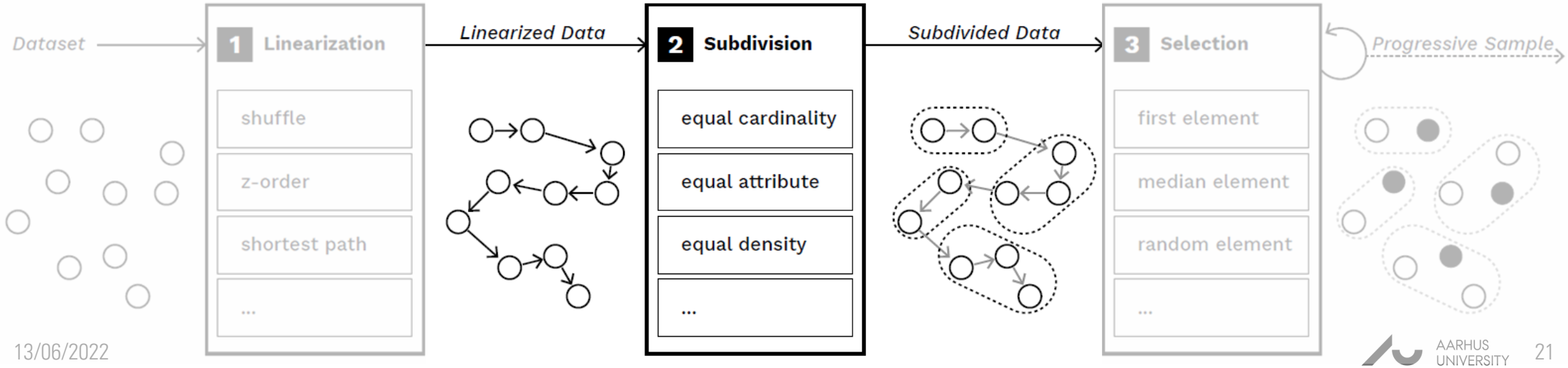


Z-order

[Zhou et al. 2020]

Subdivision

- Input: List of data items
- Output: Groups of lists of data items
- Tailor to the **analysis task**
- Based on data attributes

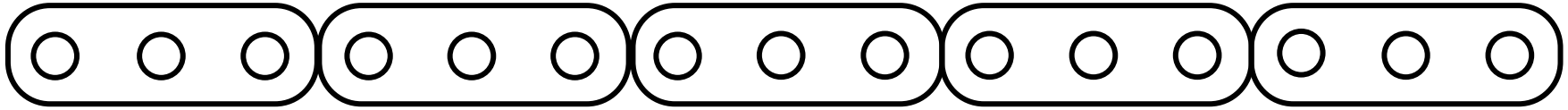


Subdivision - Example

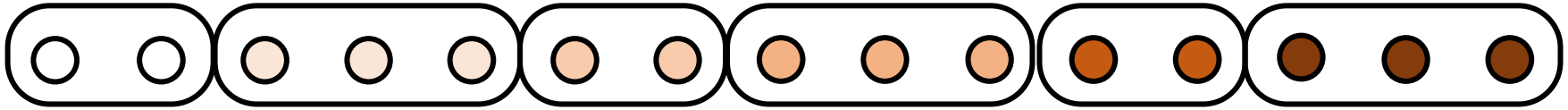
Linear. data: ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

Subdivision - Example

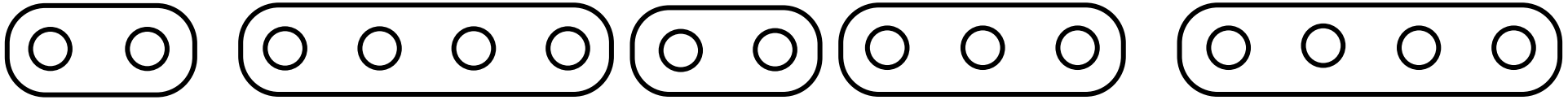
Cardinality:



Attribute:

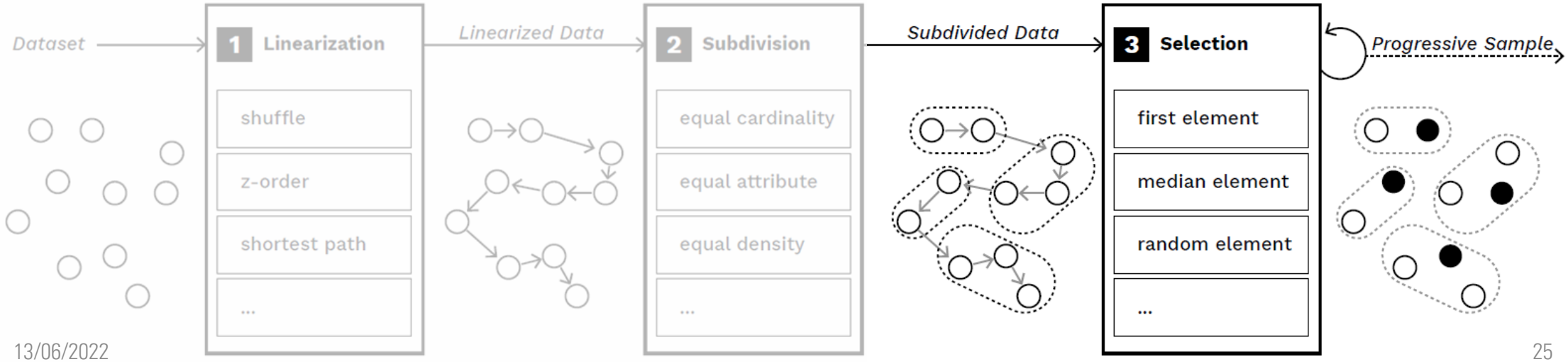


Distance:



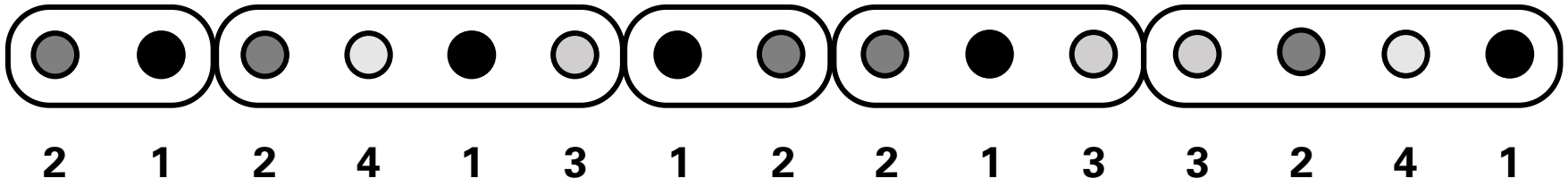
Selection

- Input: Groups of data lists
- Output: Partitions of the dataset
- Tailor to the **user interest** [Micallef et al. 2019]
- Based on the desired order of the data



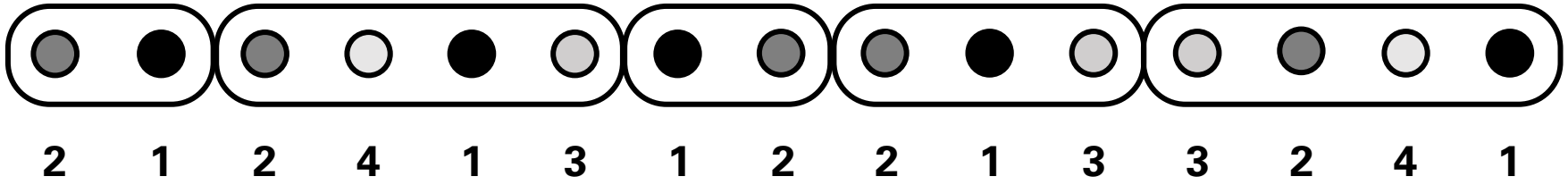
Selection – Example

Random:

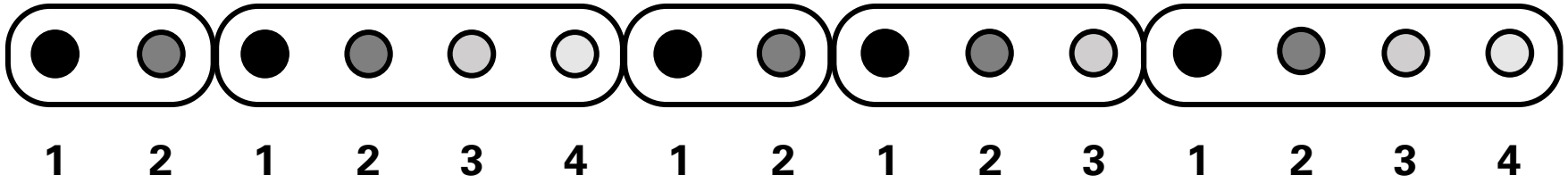


Selection – Example

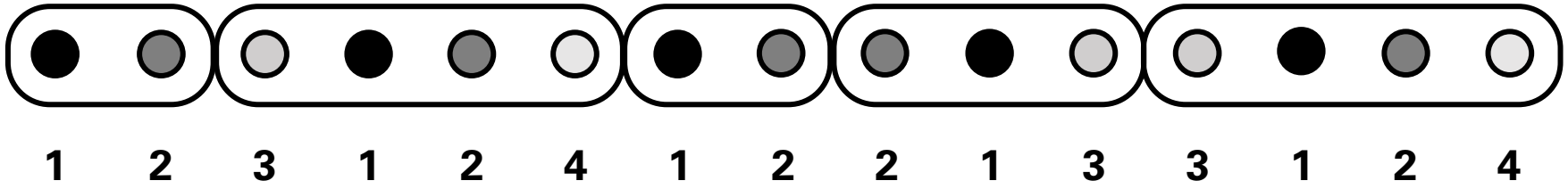
Random:



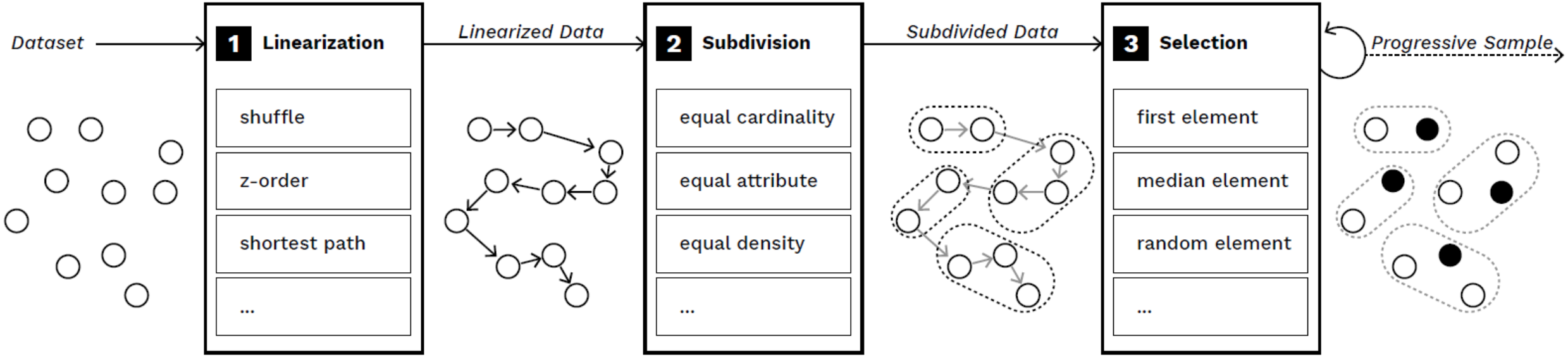
First:



Median:

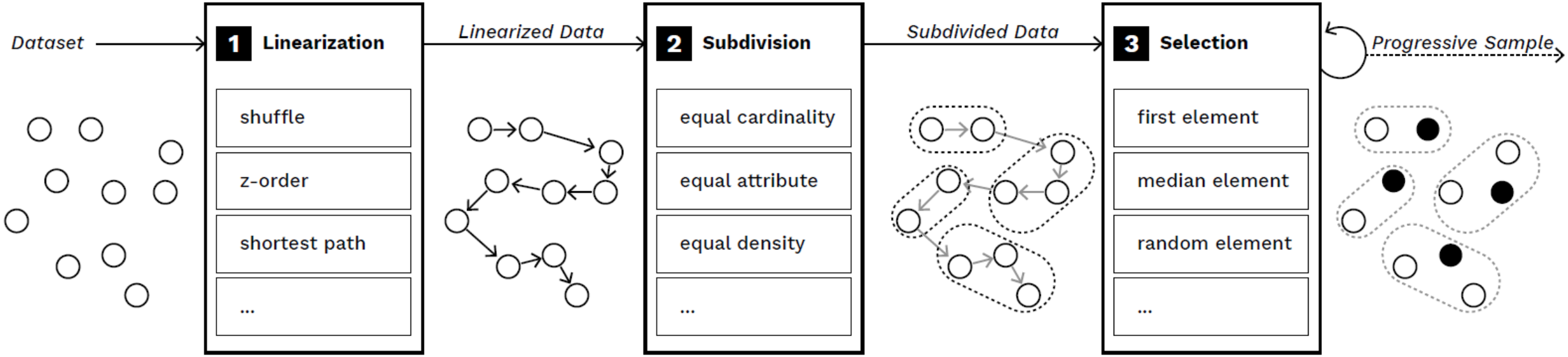


Tailoring the Sampling



Tailor for	Data	Analysis task	User Interest
based on	Data structure	Data attributes	Order of the data

Tailoring the Sampling



Tailor for	Data	Analysis task	User Interest
based on	Data structure	Data attributes	Order of the data

Using the pipeline

Geospatial dataset ("mountain peaks" from OpenStreetMaps)

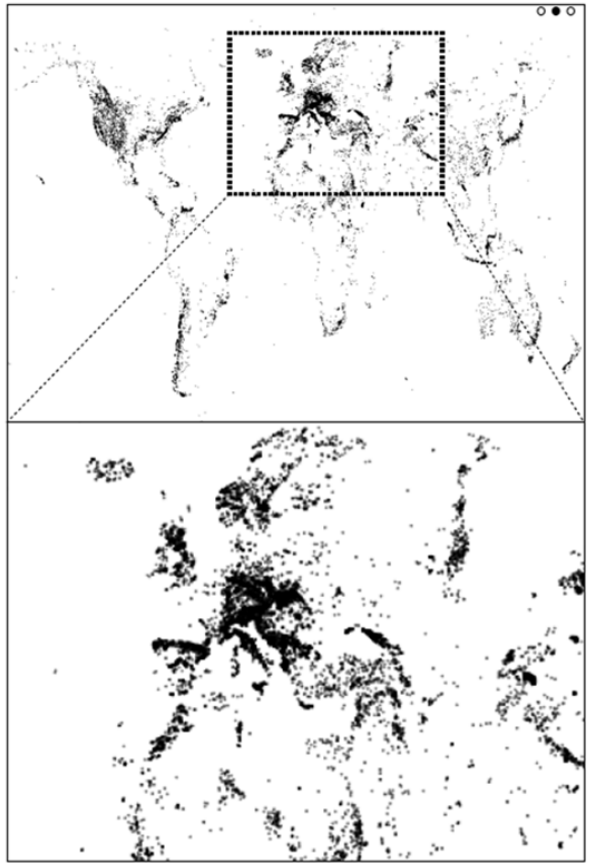
Longitude, latitude, altitude of each peak.

Explore altitude distribution in the dataset.

Explore the data

Overview
~25k items

Details
~25k items



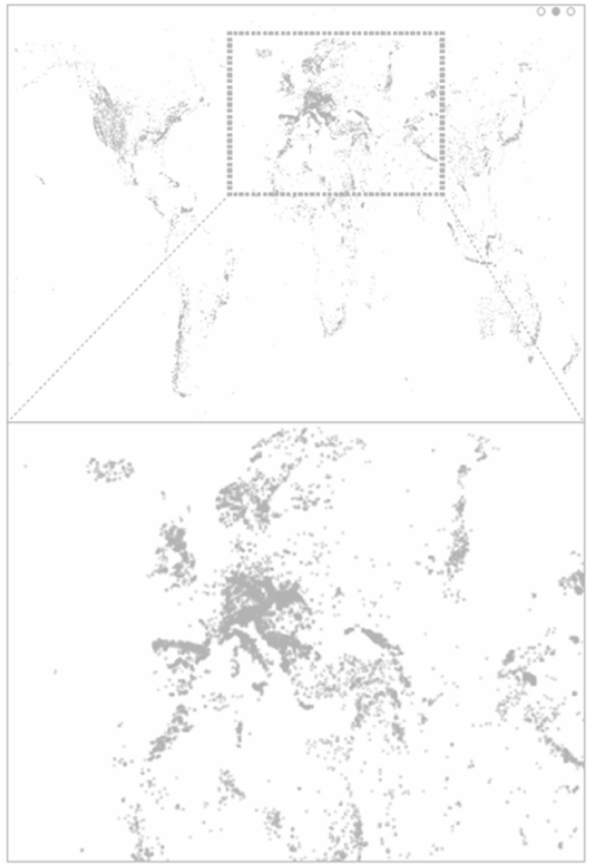
- Develop a "feel" for the dataset
- Make little assumptions about the data in the beginning

Linearization	Subdivision	Selection
random	Equal cardinality	random

Tailor for Data

Overview
~25k items

Details
~25k items



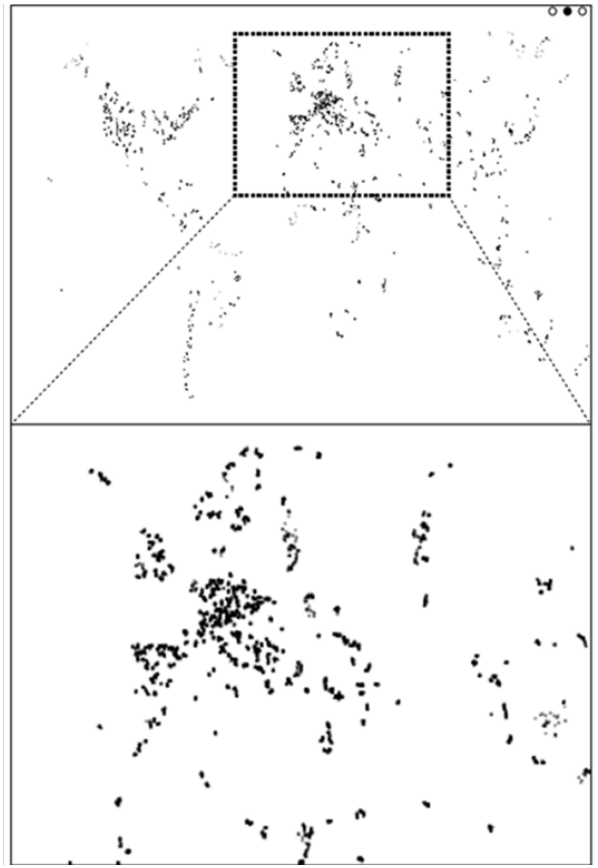
- Develop a "feel" for the dataset
- Make little assumptions about the data in the beginning
- Preserve density+outliers
  → tailor via **Linearization**

Linearization		Subdivision		Selection
z-order	▶	Equal cardinality	▶	random

Explore average altitude

Overview
~25k items

Details
~25k items



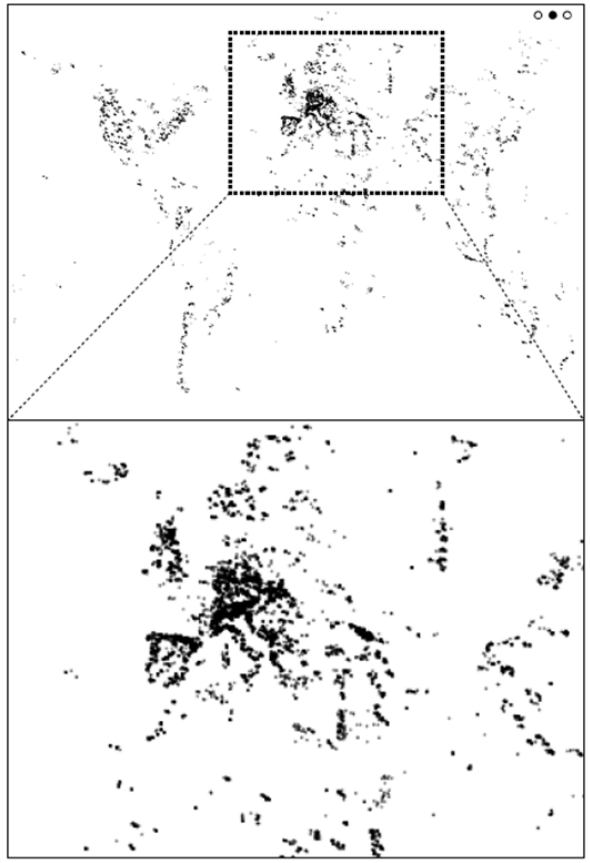
- What is the average altitude in a region?
- Change in user interest
  → tailor via **Selection**

Linearization	Subdivision	Selection
z-order	Equal cardinality	**median**

Maintain outliers

Overview
~25k items

Details
~25k items



- Get distribution of altitude
- Change in user interest
  → Tailor via **Selection**

Linearization	Subdivision	Selection
z-order	Equal cardinality	**random**

Pipeline

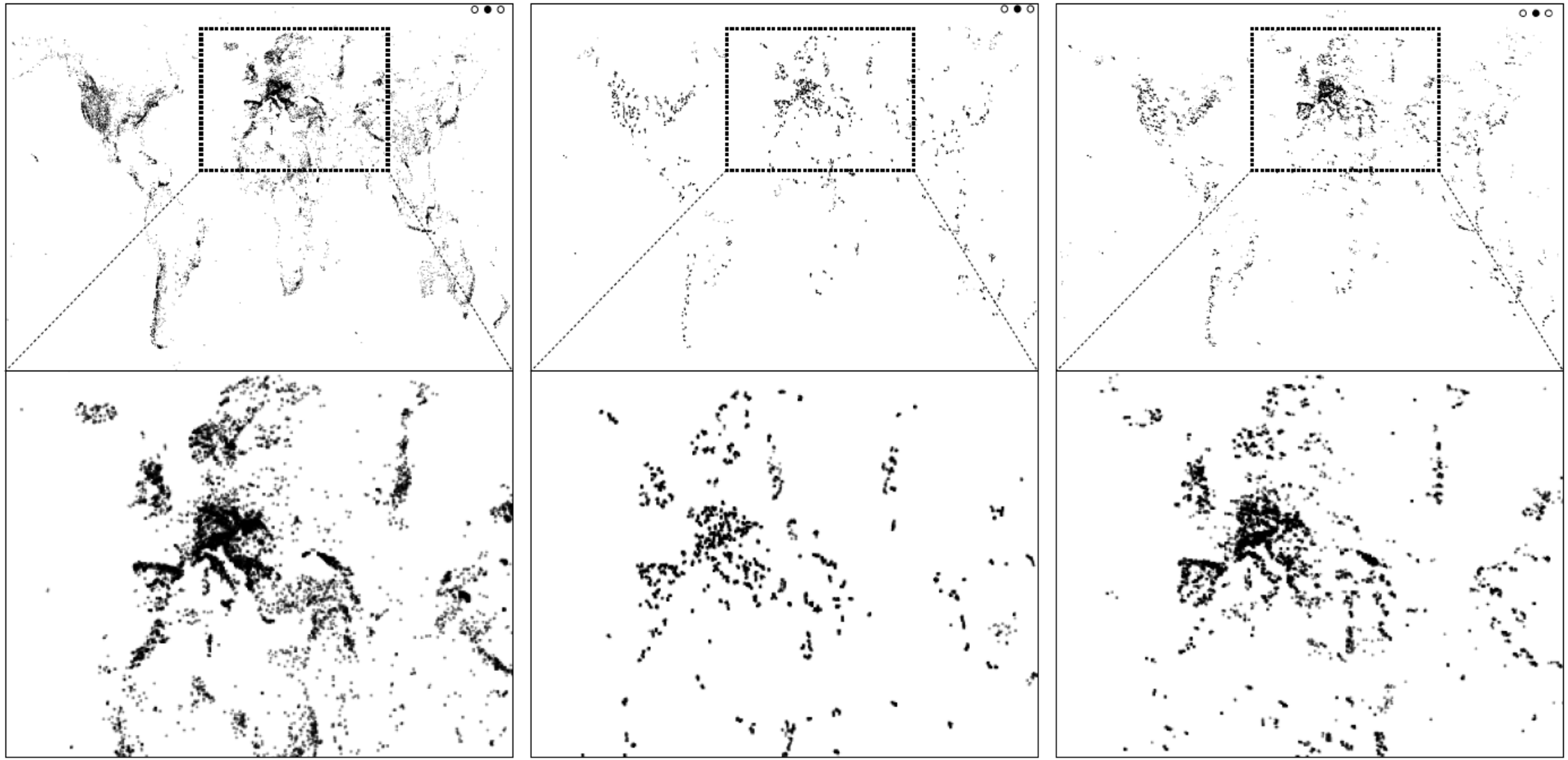
random → equal cardinality → random

z-order → equal cardinality → median

z-order → equal cardinality → random

Overview
~25k items

Details
~25k items



Overview | **Median Altitude** | **Altitude distribution**

Comparing Pipelines with ProSample

Future Work

- Develop actionable **guidelines** for pipeline configurations
  - Based on runtime, utility, parameters …

- Explore **open questions**:
  - Where in the PVA process can we position which sampling pipeline?
  - How to combine with steering approaches? How can we prioritize certain data in that pipeline? (Are the two are complementary to each other?)

Recap

A technique

- Proposed a tailorable sampling pipeline for PVA

A demonstrative use case

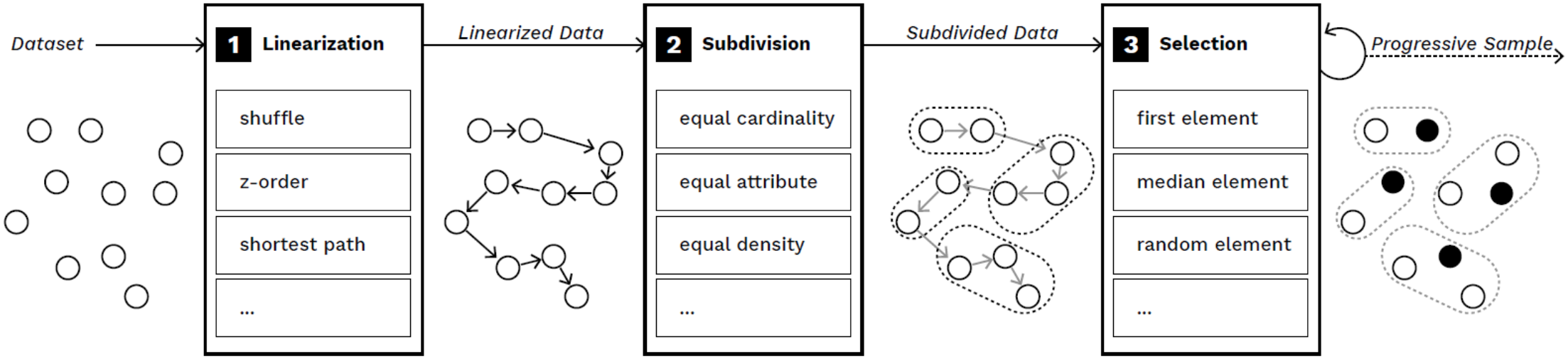
- Showed how to tailor the sampling to three scenarios

A tool

- Introduced ProSample

A Pipeline for Tailored Sampling in Progressive Visual Analytics

Marius Hogräfer, Jakob Burkhardt, Hans-Jörg Schulz



Try ProSample: **https://vis-au.github.io/prosample**

AARHUS UNIVERSITY