M2 DATASCALE: DATA MANAGEMENT IN LARGE-SCALE DISTRIBUTED SYSTEMS

Investigating feature selection techniques to improve data mining tasks

Project supervised by

Dr Zaineb GARCIA

zaineb.chelly-dagdia@uvsq.fr

UVSQ — Paris Saclay University

Context

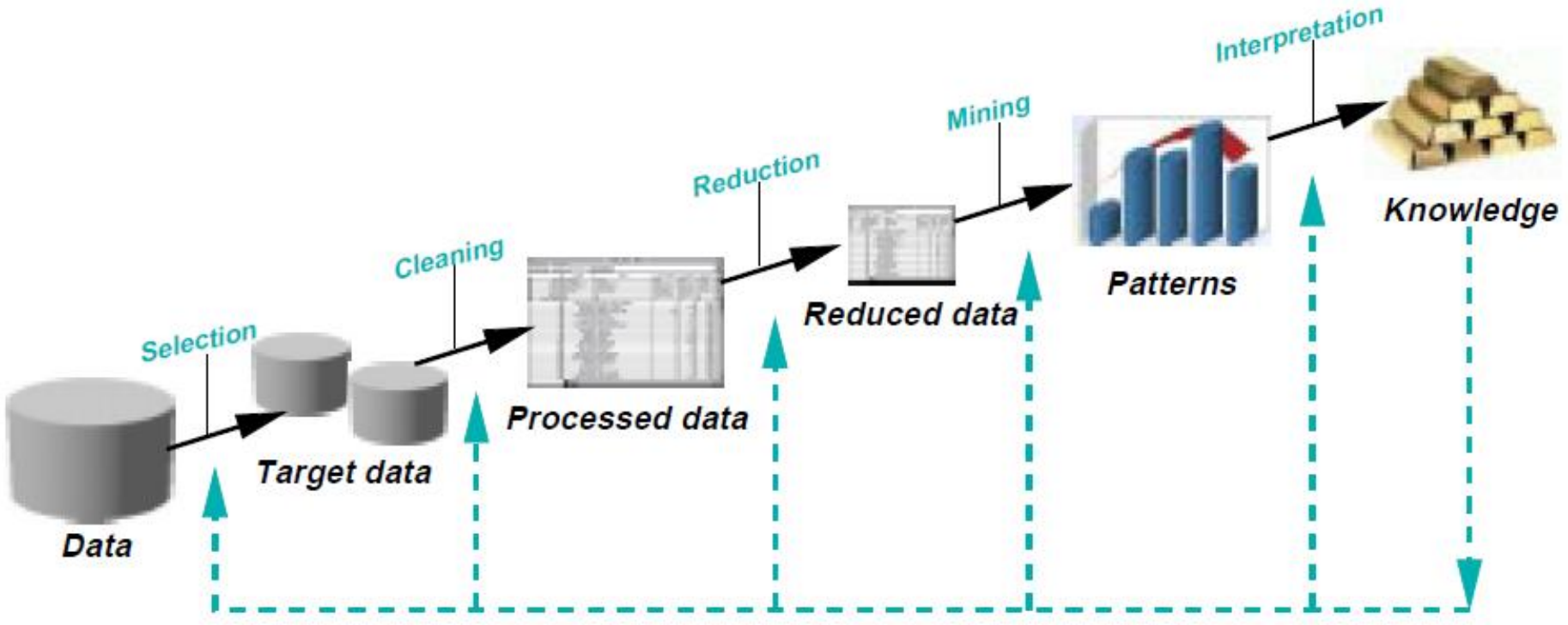

Figure: The KDD process

Context

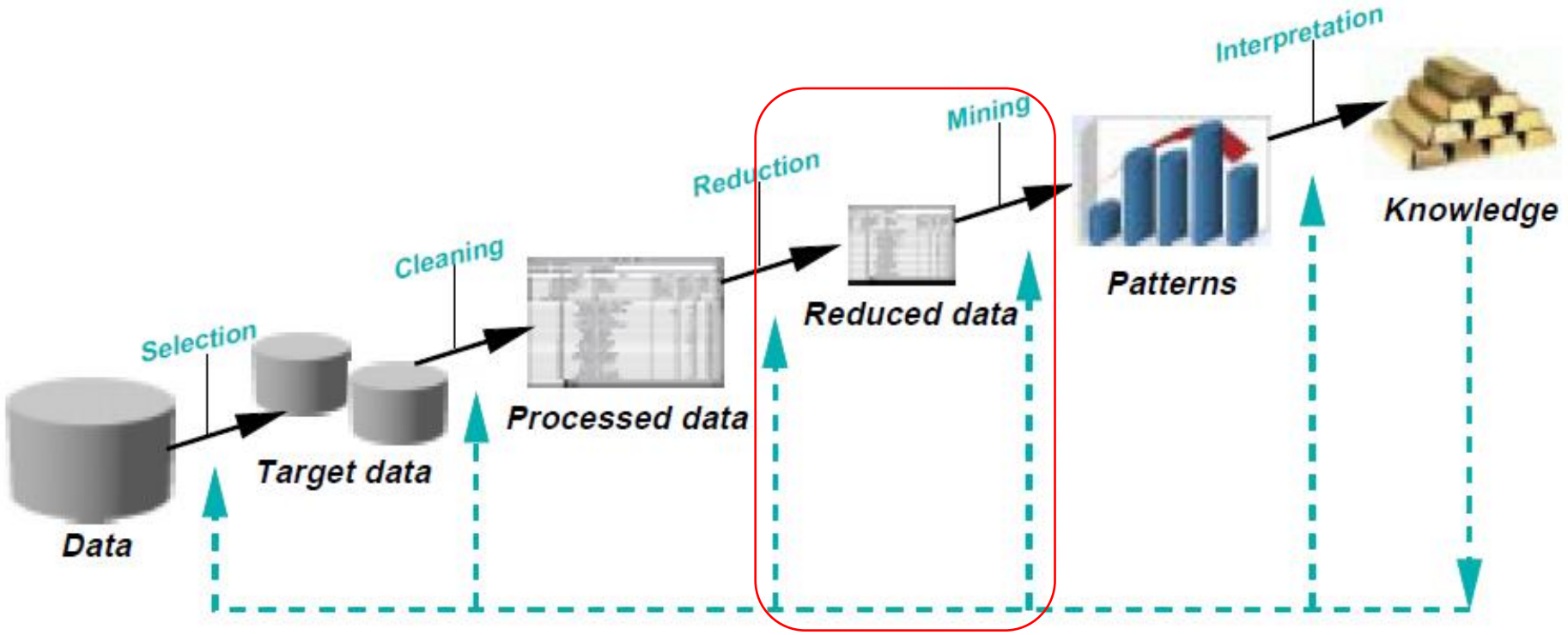

Figure: The KDD process

→ **Reducing the data : Source of significant data loss**

→ **Use of suitable techniques**

Taxonomy of dimensionality reduction techniques

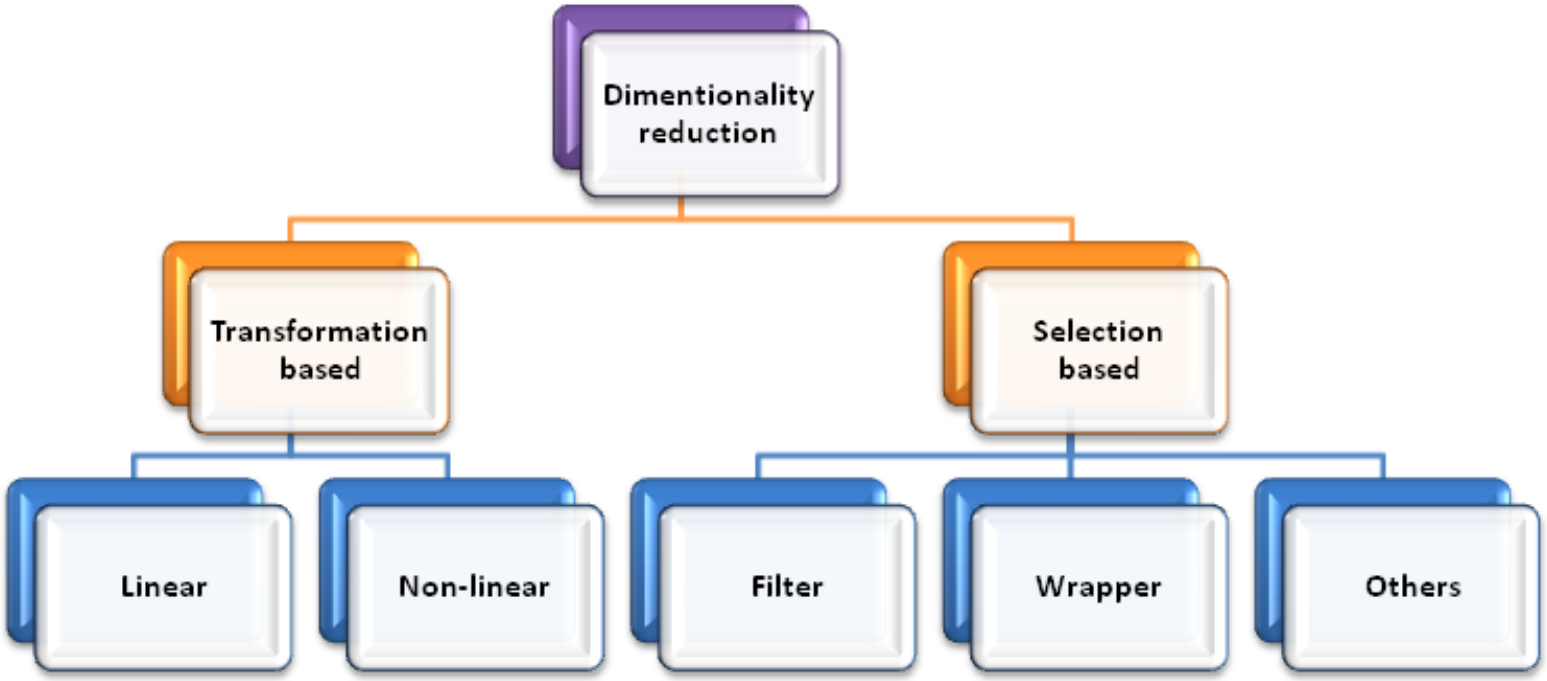

Figure: Classification of dimensionality reduction techniques

Feature selection

- **Feature selection methods are particularly desirable as these facilitate the interpretability of the resulting knowledge**

We need a technique that:

- Analyzes the facts hidden in data;
- Does not need any additional information about the data such as thresholds or expert knowledge on a particular domain;
- Finds a minimal knowledge representation;

→ **Rough Set Theory**

Rough Set Theory

- Rough set theory was developed by Zdzislaw Pawlak

- Rough sets constitutes a sound basis for data mining as a tool to:
  - ✓ **Feature selection;**
  - ✓ Discretization;
  - ✓ Decision rule generation;
  - ✓ Classification ;

Project tasks

❖ **Objective:** **Investigate several rough set based feature selection techniques**

Tasks:

1) Understand the in-depth functioning of the QuickReduct Algorithm as well as its technical details
2) Implement the QuickReduct Algorithm
3) Test the code using a sample dataset to validate the algorithm's implementation
4) Investigate other rough set based algorithms for comparison purposes
5) Test the relevance of the selected features on a set of classification algorithms using UCI machine learning datasets
6) Present visually the results
7) Identify some limitations of the used algorithm
8) Propose some improvements of the algorithm
9) Write the technical report
10) Present the conducted work

Algorithm 3.1 The QuickReduct Algorithm

```
C: the set of all conditional features;
D: the set of decision features;
R ← {};
do
   T ← R
   ∀ x ∈ (C - R);
   if γ_{R∪{x}}(D) > γ_T(D);
    T ← R ∪ {x};
   end if
   R ← T;
until γ_R(D) == γ_C(D)
return R
```