Int. Conf. on Pattern Recognition (ICPR) 2020

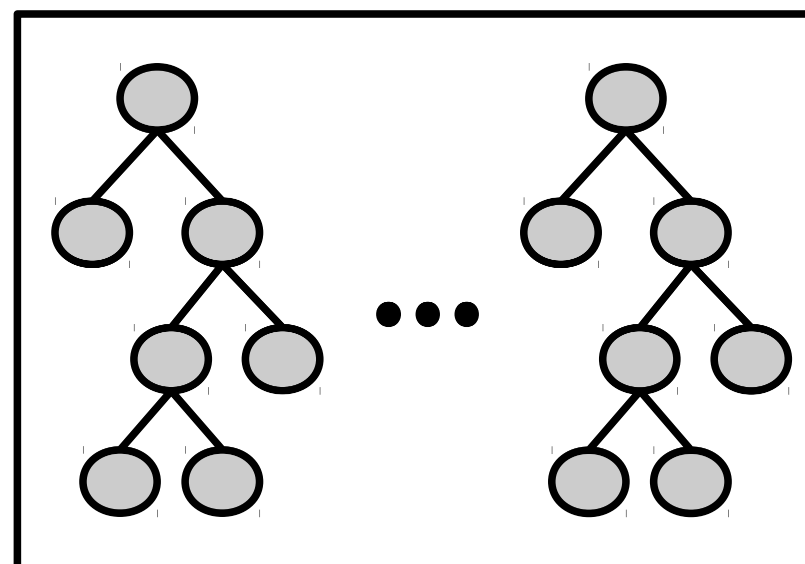
On learning Random Forests for Random Forest-clustering

Manuele Bicego, Francisco Escolano
University of Verona (Italy), Universidad de Alicante (Spain)
manuele.bicego@univr.it, sco@dccia.ua.es

Random Forest Clustering

- Random Forests: powerful and interpretable tools based on aggregation of decision trees
- Main exploitation: classification and regression

In other scenarios, such as **clustering**, they have been less investigated



Random Forest Clustering

- Main RF-clustering approaches:
  - Methods based on direct exploitation of RF (or RF-like schemes) to get clustering
  - Methods which exploit description capabilities of RF to derive a dissimilarity measure (to be used with standard distance-based clustering methods)

Random Forest Clustering

- Main RF-clustering approaches:
  - Methods based on direct exploitation of RF (or RF-like schemes) to get clustering
  - Methods which exploit description capabilities of RF to derive a dissimilarity measure (to be used with standard distance-based clustering methods)

The general scheme

Objects to be clustered

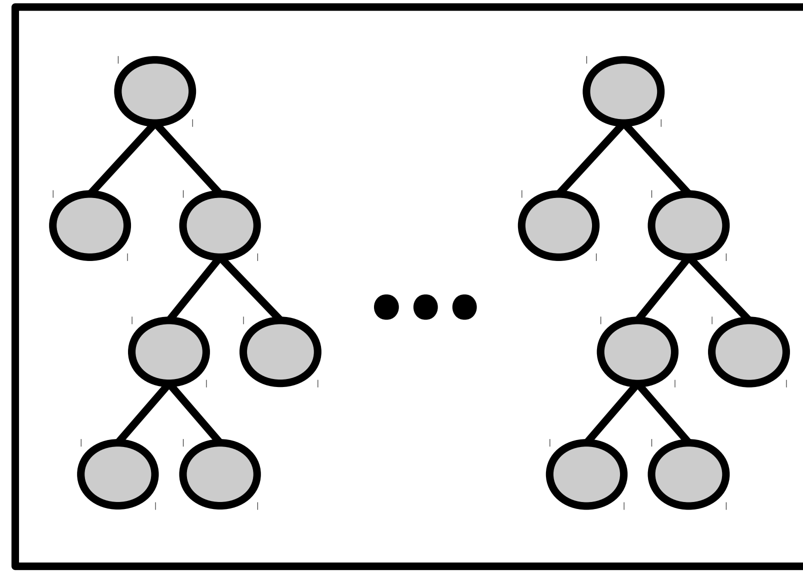
$$\boxed{\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{d1} & \cdots & x_{dN} \end{bmatrix}}$$

The general scheme

Objects to be clustered

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{d1} & \cdots & x_{dN} \end{bmatrix}$$

Random Forest

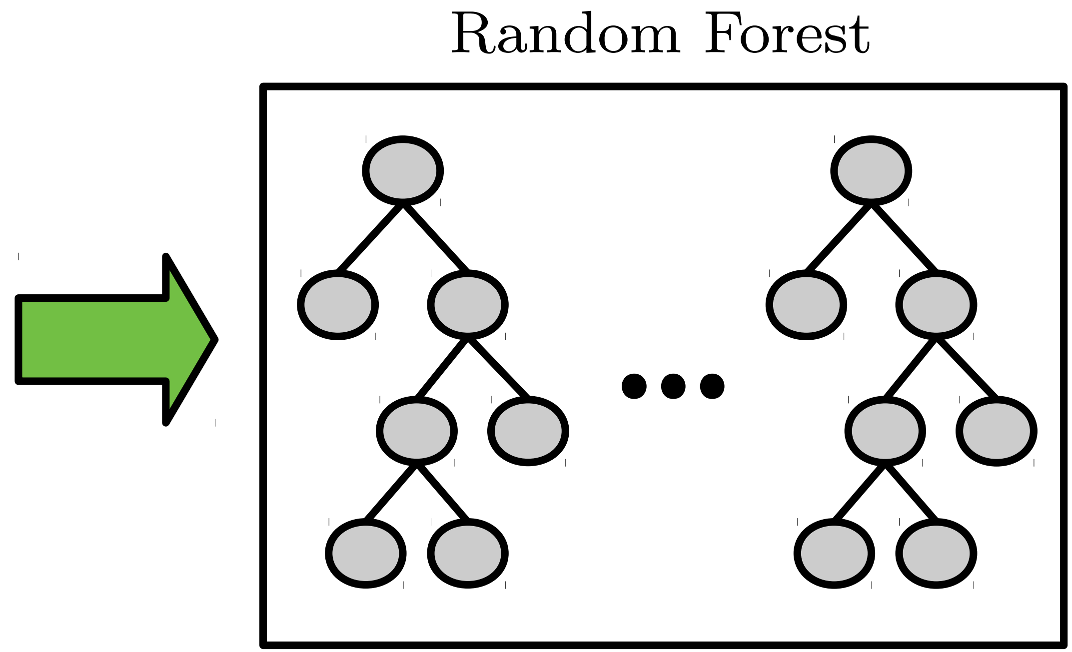


<u>STEP 1</u>: Build a Random Forest on **X**

The general scheme

Objects to be clustered

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{d1} & \cdots & x_{dN} \end{bmatrix}$$



STEP 1: Build a Random Forest on **X**

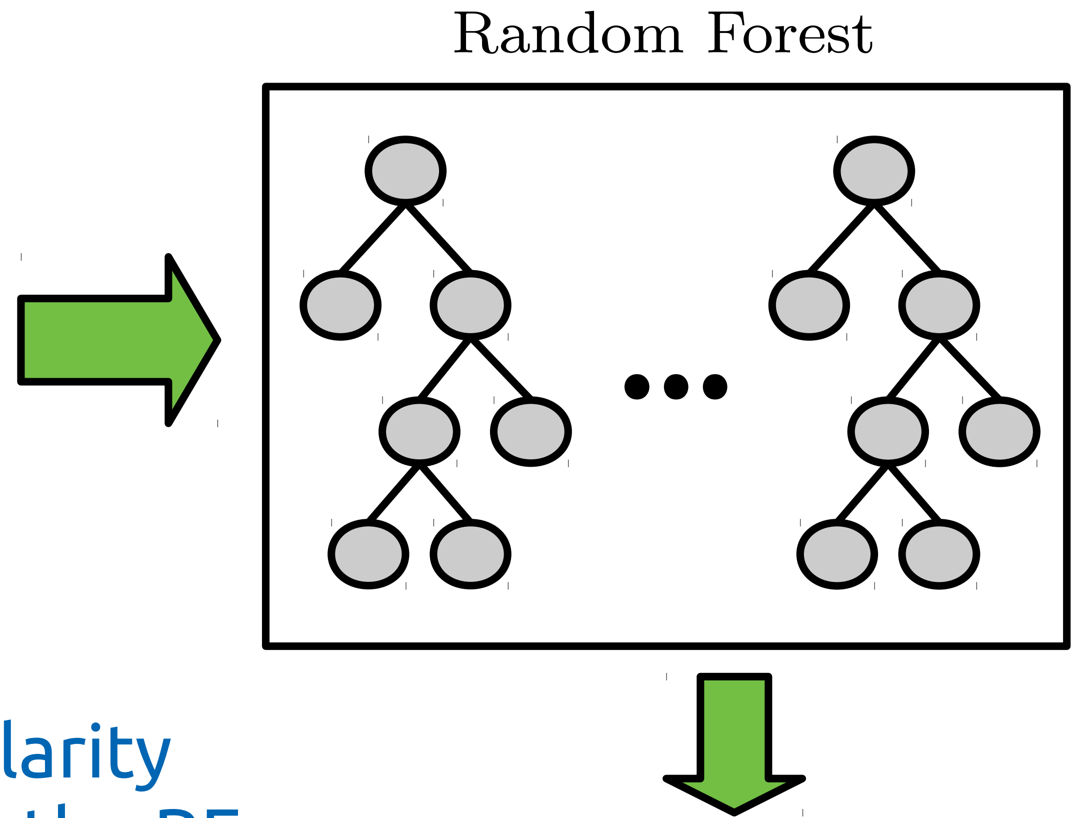
Problem: no labels!

Classic approach:
Generate negative class + classification Random Forest

The general scheme

Objects to be clustered

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{d1} & \cdots & x_{dN} \end{bmatrix}$$



STEP 2: Extract a dissimilarity between points **through** the RF

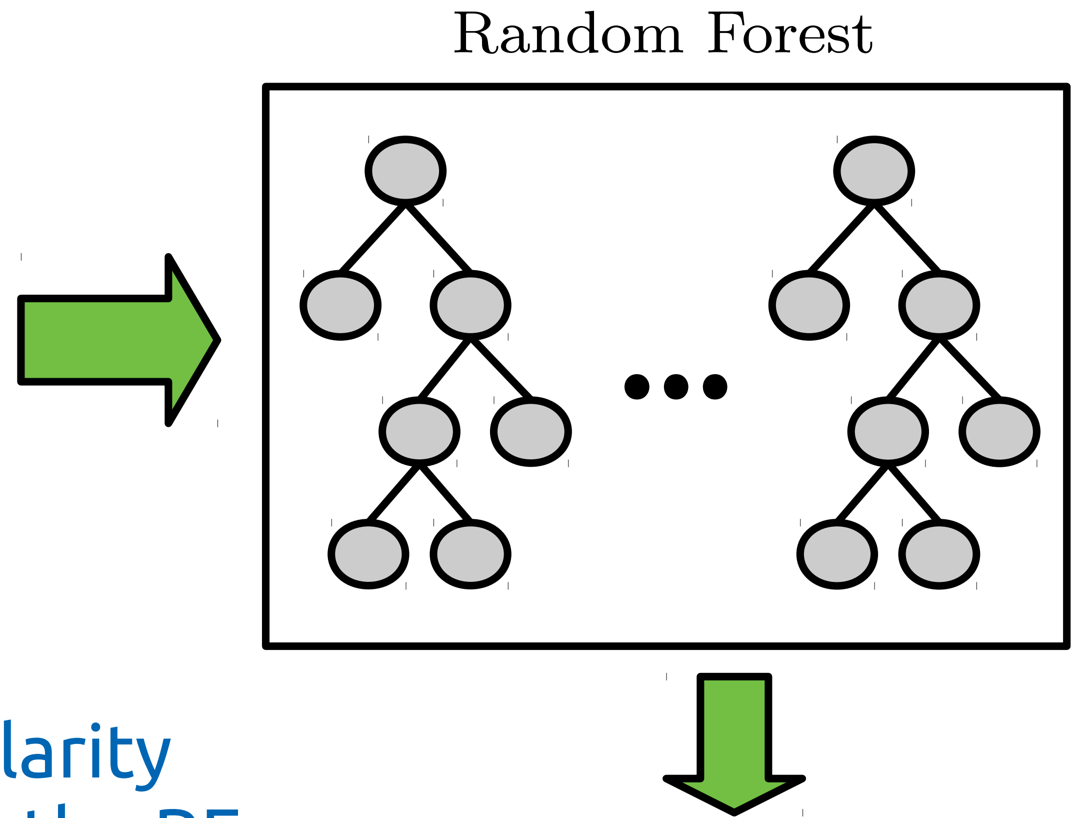
Dissimilarity Matrix

$$D = [dis(\mathbf{x}_i, \mathbf{x}_j)]$$

The general scheme

Objects to be clustered

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{d1} & \cdots & x_{dN} \end{bmatrix}$$



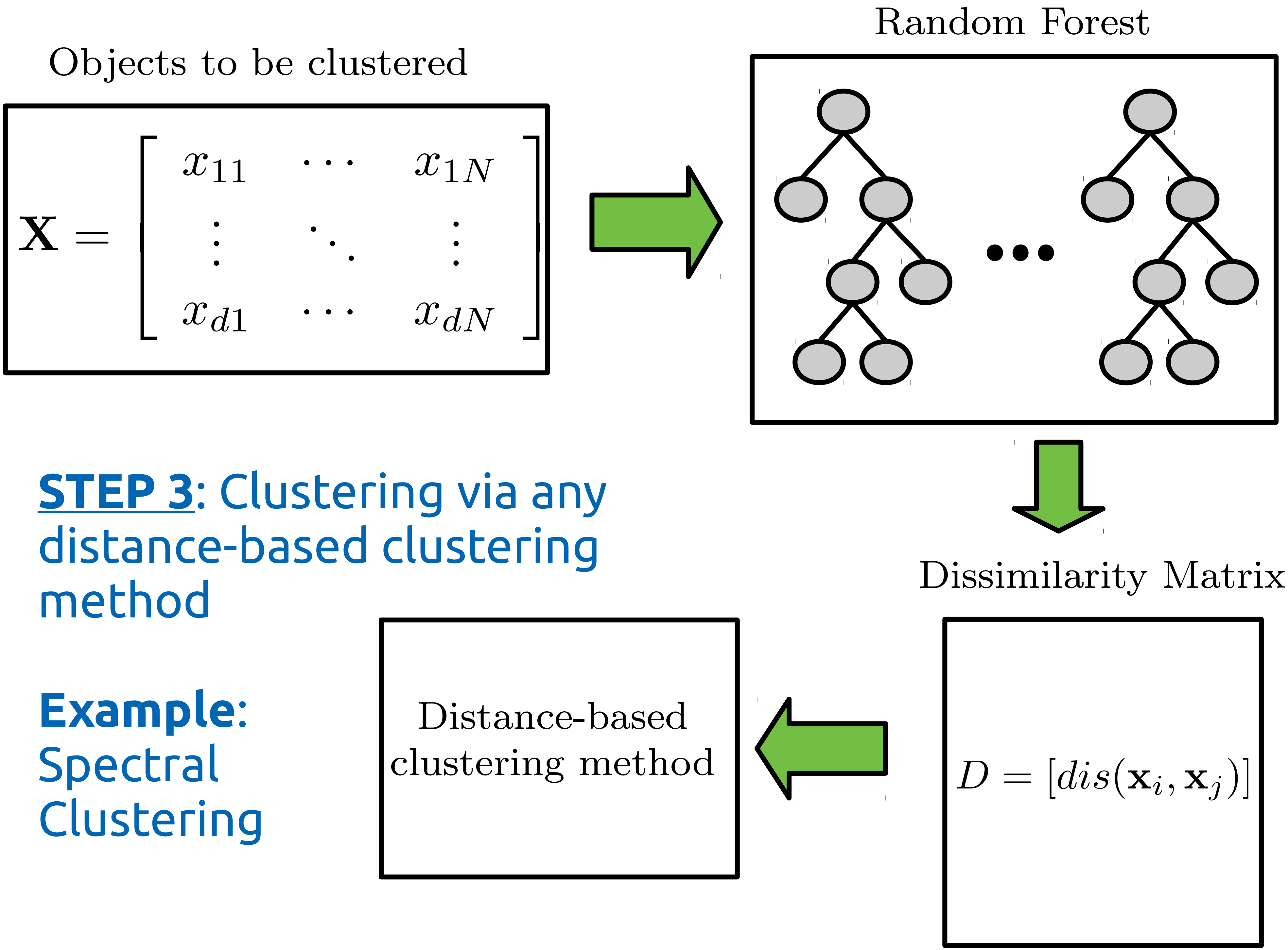
<u>STEP 2</u>: Extract a dissimilarity between points **through** the RF

Example: two points are similar if, in the different RF trees, they fall **very often** in the same leave (similar answers to tests)

Dissimilarity Matrix

$$D = [dis(\mathbf{x}_i, \mathbf{x}_j)]$$

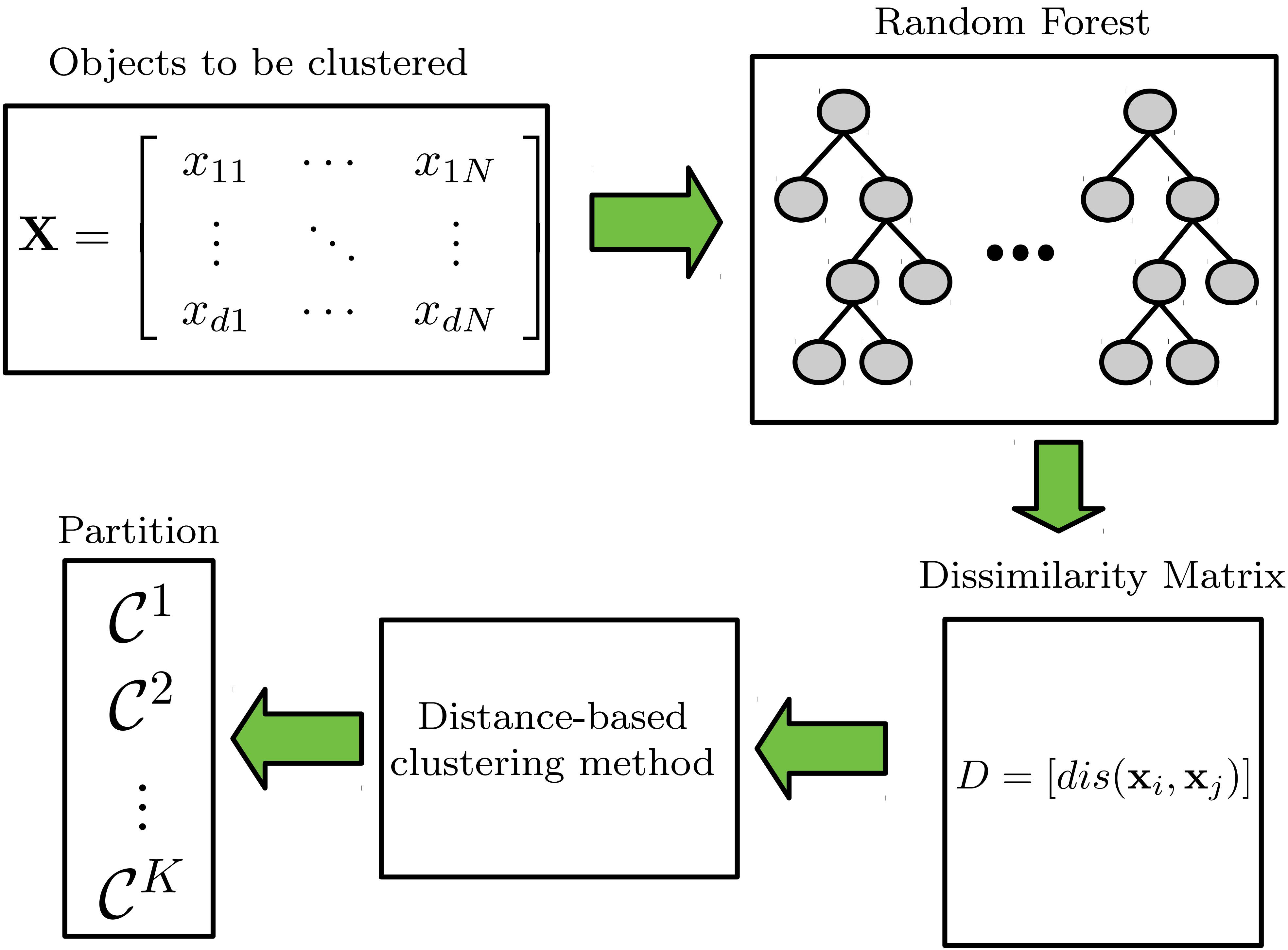
The general scheme

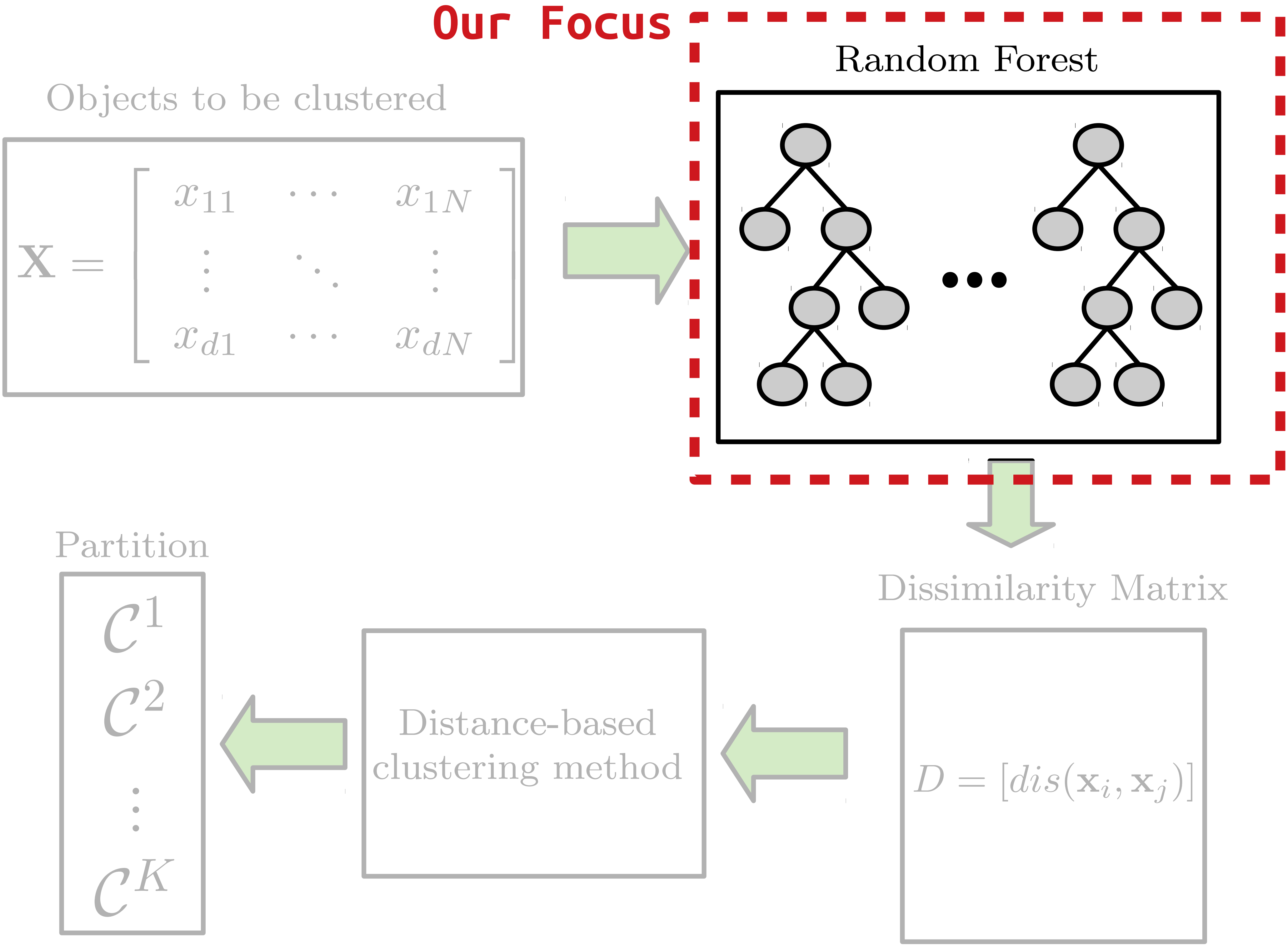


STEP 3: Clustering via any distance-based clustering method

Example: Spectral Clustering

The general scheme





Our Focus
Random Forest
Objects to be clustered
$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{d1} & \cdots & x_{dN} \end{bmatrix}$
Partition
$\mathcal{C}^1$
$\mathcal{C}^2$
$\vdots$
$\mathcal{C}^K$
Distance-based clustering method
Dissimilarity Matrix
$D = [dis(\mathbf{x}_i, \mathbf{x}_j)]$

Learning Random Forests

- **STEP1** (Learning of RF) has received poor attention by researchers (main efforts are on **STEP 2**)
  - Most of the cases: generation of a synthetic negative class plus training of a standard classification RF
  - Few others: use of completely randomized RF (as in Extremely Randomized Trees [Geurts et al, ML06])

- Our position: **this step is crucial!**

Learning Random Forests

- **Our contributions**:
  - We propose two novel solutions for learning RF in RF-clustering
  - We perform a thorough experimental evaluation to show that a proper learning of RF is fundamental in RF clustering
  - We derive a set of guidelines to suggest the proper learning depending on the given dataset

Contribution 1: novel learning schemes

- **Gaussian Density Random Forests**
  - Random Forests designed for density estimation (Criminisi et al 2012) but never used for RF-clustering
  - Trees are built so that in each node the **Gaussian entropy** is maximized
    - Assumption: data in each node follow a Gaussian distribution

Contribution 1: novel learning schemes

- **Rényi Random Forests**

  - Novel Random Forests we introduce in this paper
  - Trees are built so that in each node the **Renyi entropy** is maximized
    - The Renyi entropy is estimated using **non parametric** bypass entropy estimator
    - Appropriate when the Gaussianity assumption is too strict

All details are in the paper!

Contribution 2: thorough experimental evaluation

- We employed 8 standard UCI-ML datasets
- We analyse different options for all the steps
  - **STEP 1**: 4 learning strategies (ClassRF, RandRF, GaussRF and RenyiRF), with different parametrizations
  - **STEP 2**: 4 different distances
    - Shi: [Shi et al 2006]
    - Zhu2, Zhu3: [Zhu et al, CVPR14]
    - Ting: [Ting et al, KDD16]
  - **STEP 3**: 3 different distance-based methods
    - Spectral clustering, Affinity Propagation, Hierarchical clustering (Ward-Link)

Results

- All the numbers are in the paper!
- Main findings:
  - The classic learning scheme is hardly the best solution (in less than 2% of the cases)
  - Random Forests based on data entropy (Gaussian or Rényi) seem to be very promising
  - Also random training works adequately well, especially for high dimensional datasets

Contribution 3: guidelines for RF-clustering

- We provide suggestions for **all STEPS** of RF-clustering (details in the paper)
- For the learning:
  - If the problem is highly dimensional → use the Random-RF scheme;
  - If the problem is low dimensional:
    - Train with Gauss-RF strategy
    - Check the Gaussianity of the resulting clusters (e.g. with Royston's test);
    - If all clusters are non-Gaussian, discard the trained RF and train a RF with the Rényi-RF strategy

Conclusions

- The proper learning of Random Forests for RF-clustering is crucial
- Methods based on data entropy are adequate for low dimensional datasets
- Methods based on random mechanisms can work very well, especially in high dimensional spaces

Thanks for your attention!