PhD In Information and Communication Technologies

Curriculum: Information and Communication Engineering

Department of Information Engineering, Electronic and Telecommunications (DIET)

SAPIENZA
UNIVERSITÀ DI ROMA

Distributed Algorithms for Data Mining and Knowledge Discovery

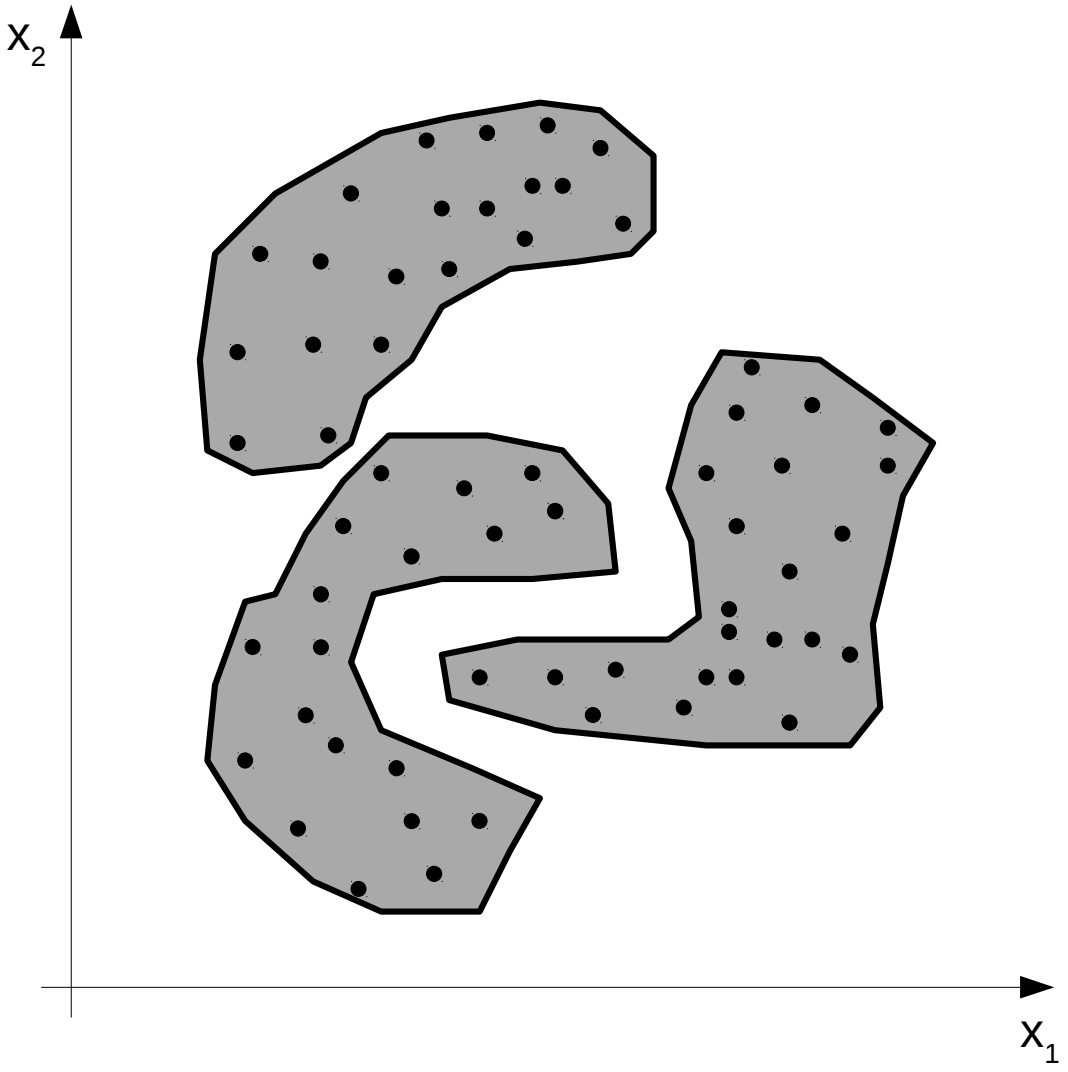
Supervisor:
Prof. Antonello Rizzi
Hosting professor:
Prof. Salima Hassas

Candidate:
Mauro Giampieri

Clustering and Classification background

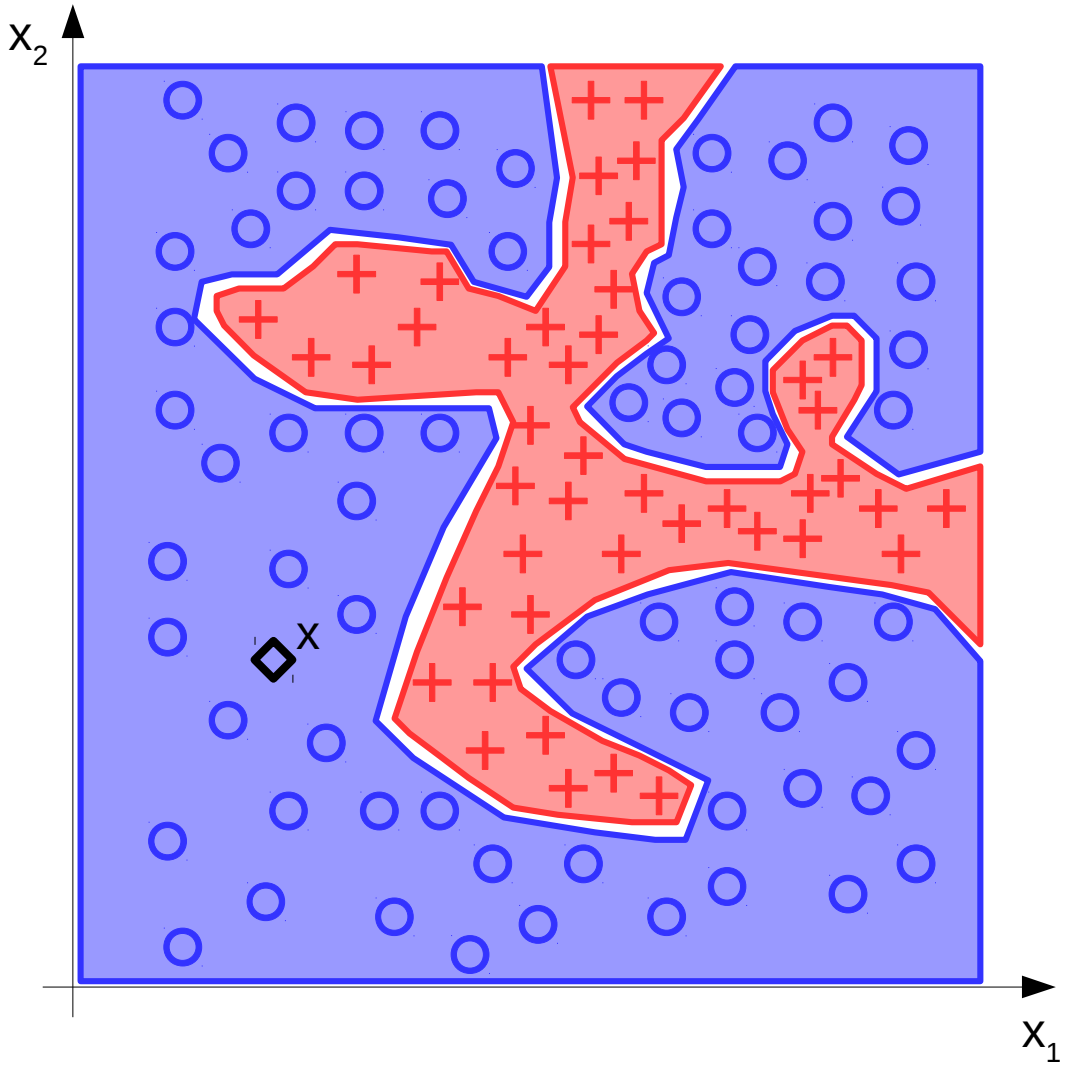
We have an increasing amount of data to deal with

Clustering



Discover space areas where patterns are more dense and group them in subsets called clusters

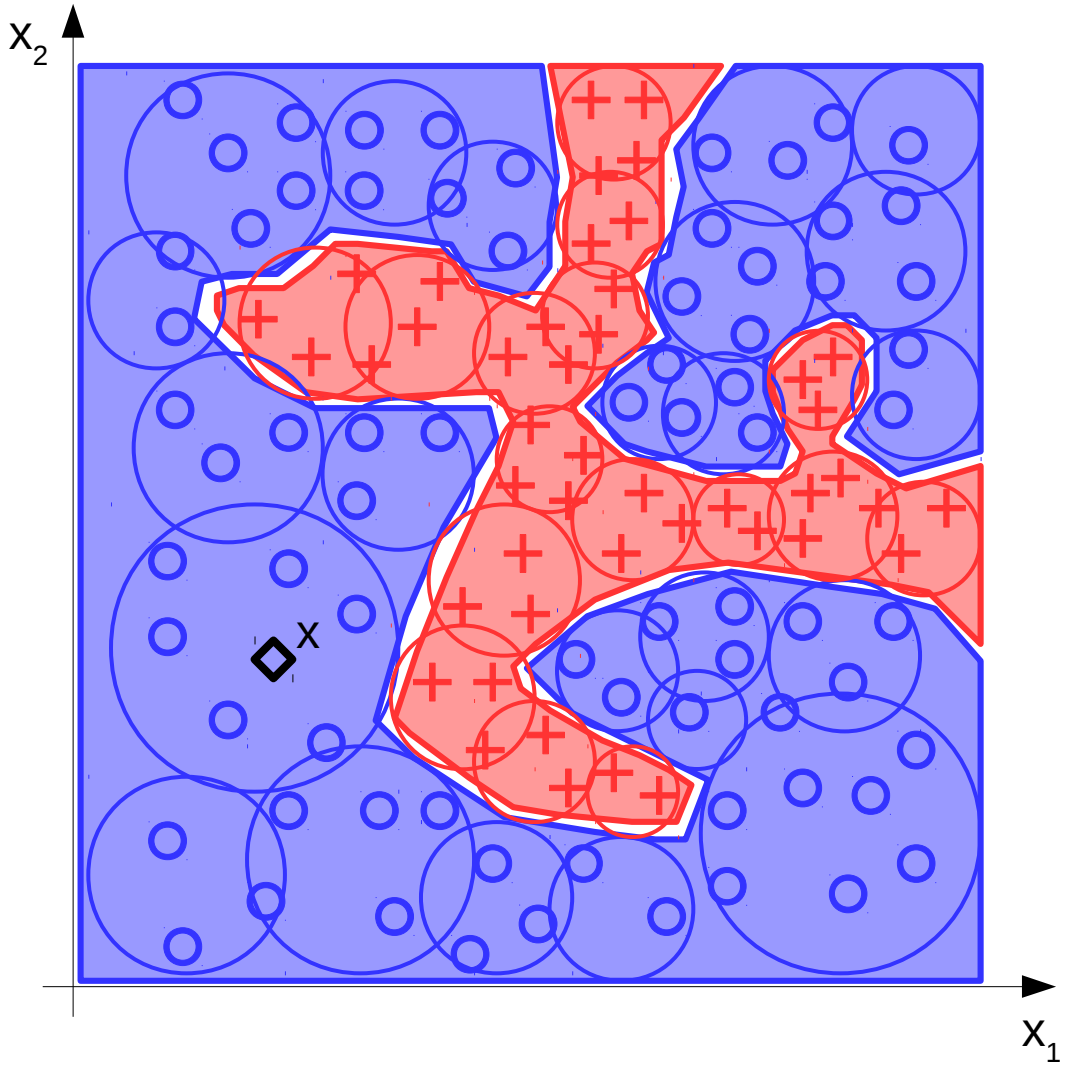
Classification



Segment space and associate a class label to each area.

The built model is used to predict the class of unknown patterns

Clustering based Classification

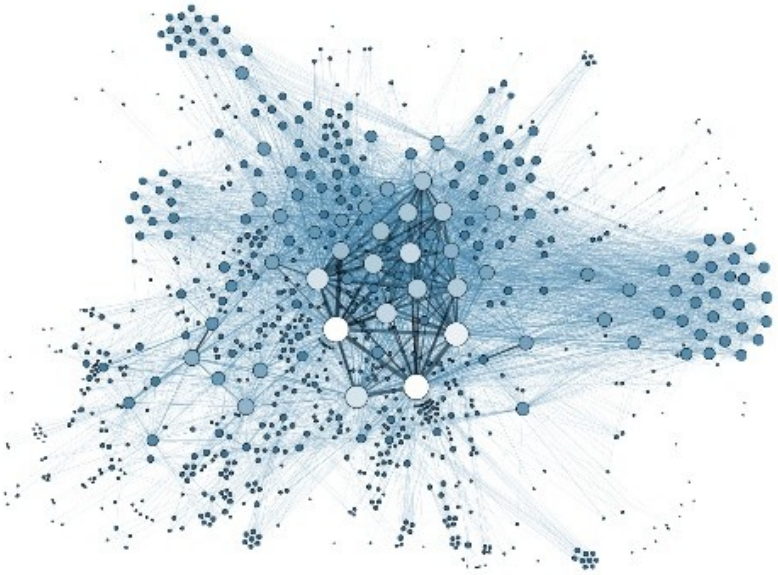


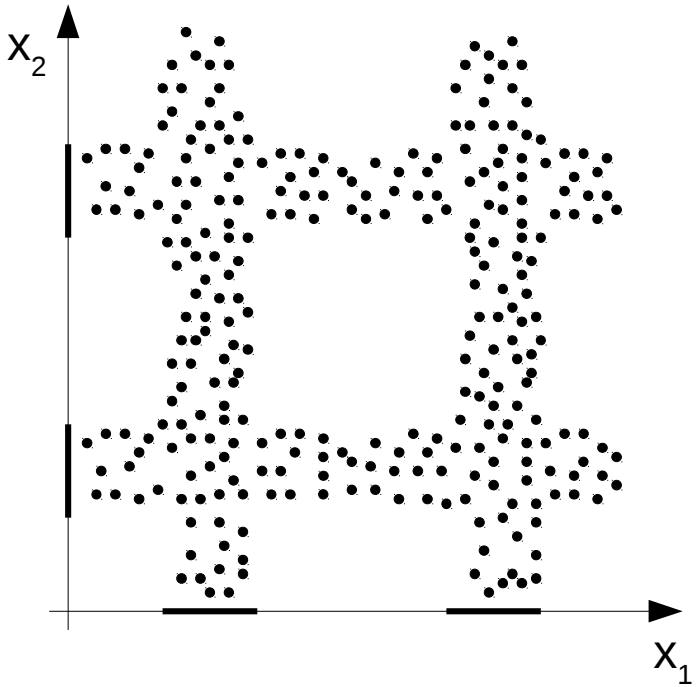
Classification may take advantage of clustering to extract information granules used for building a classification model

A Supervised Classification System based on Evolutive Multi-Agent Clustering

E-ABC: Evolutive Agent Based Clustering

Need: huge datasets analysis tools

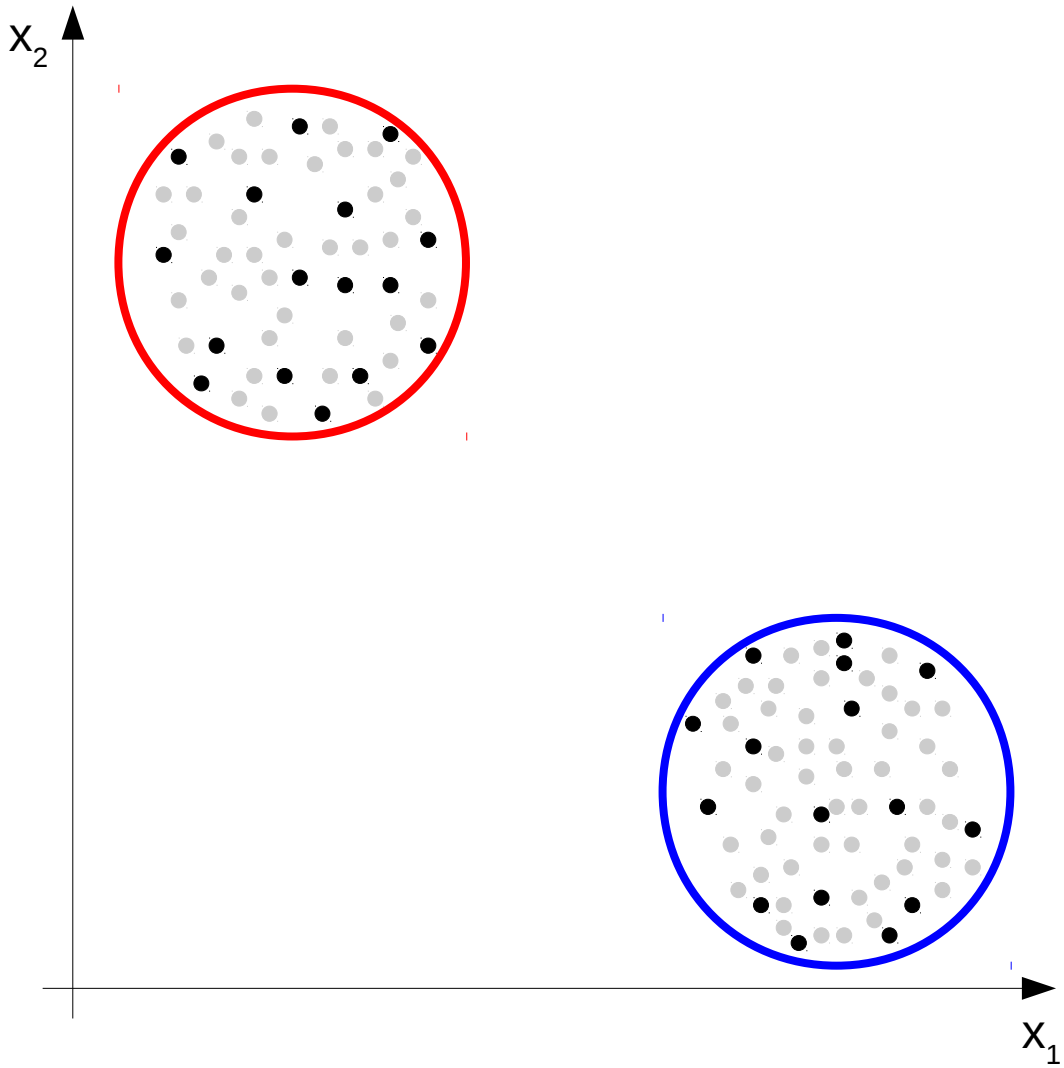




Observation: clusters may lie in different subspaces

E-ABC: Evolutive Agent Based Clustering

Assumption: clusters are still visible in a subsampled dataset



State of the art

Main approaches:

- Multi-agent random walks grouping patterns in clusters
- Concurrent agents employing different algorithms
- Cooperating agents for clusters generation and validation

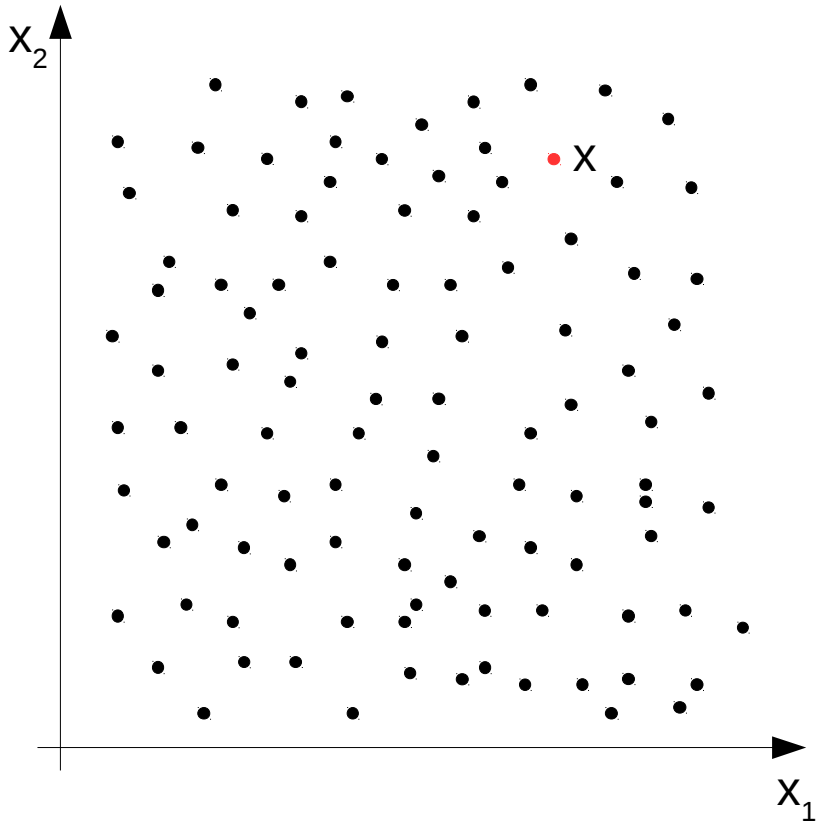
Missing:

- Simultaneous clustering and local metric learning (Acea requirements)

E-ABC: main features

- Structure suitable for big data analysis: each agent analyses a subsampling of the dataset
- Local metric learning: each agent acts in a different subspace
- Coping with non trivial (and possibly non metric) spaces: sequences, graphs, structured data

Pattern representation

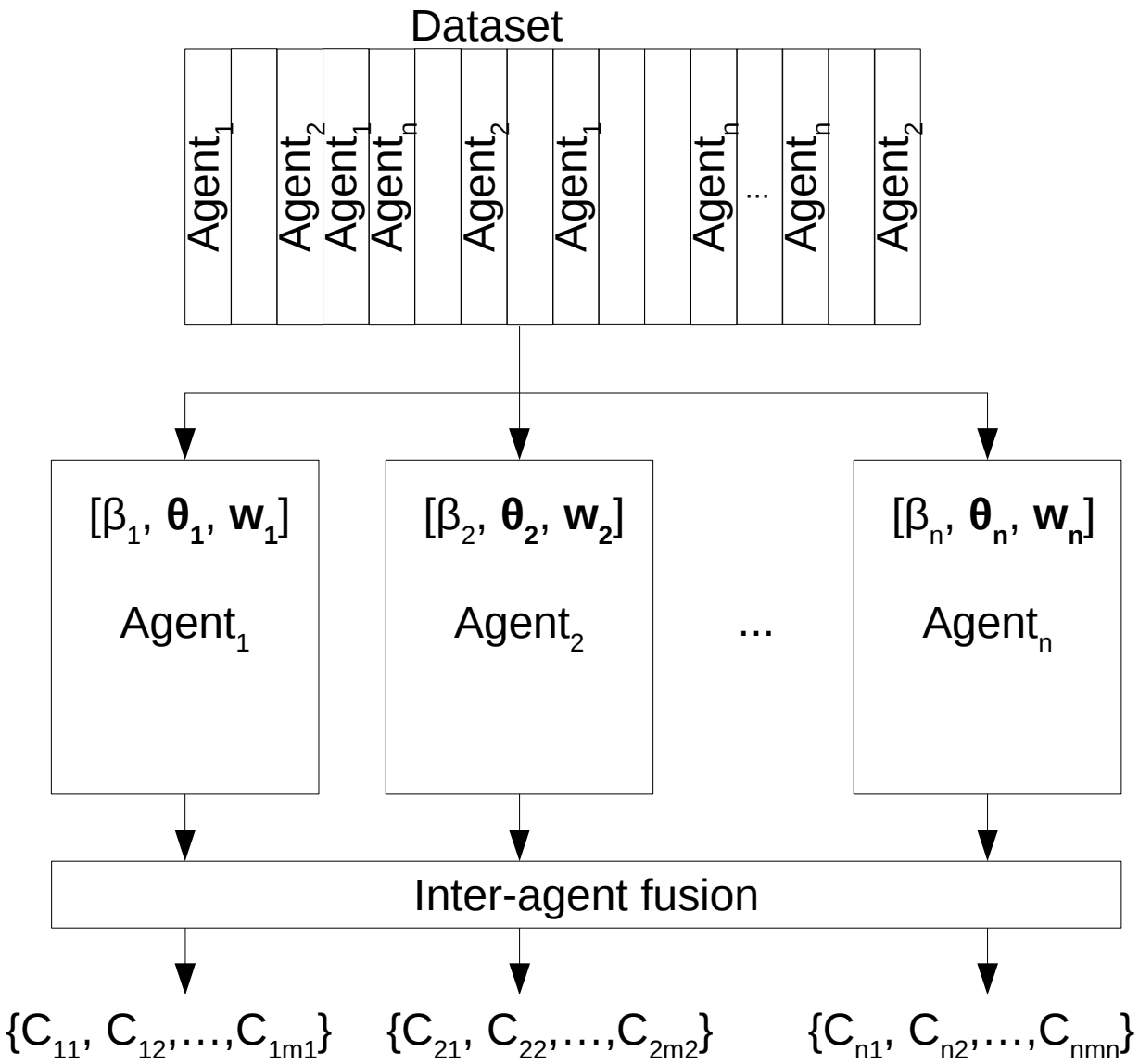


$$x = (x_1, x_2, \ldots, x_n)$$

$$x_1, x_2, \ldots, x_n \in [0,1]$$

For easier graphical representations, in the following we assume patterns can be described by cartesian coordinates in the unitary hypercube

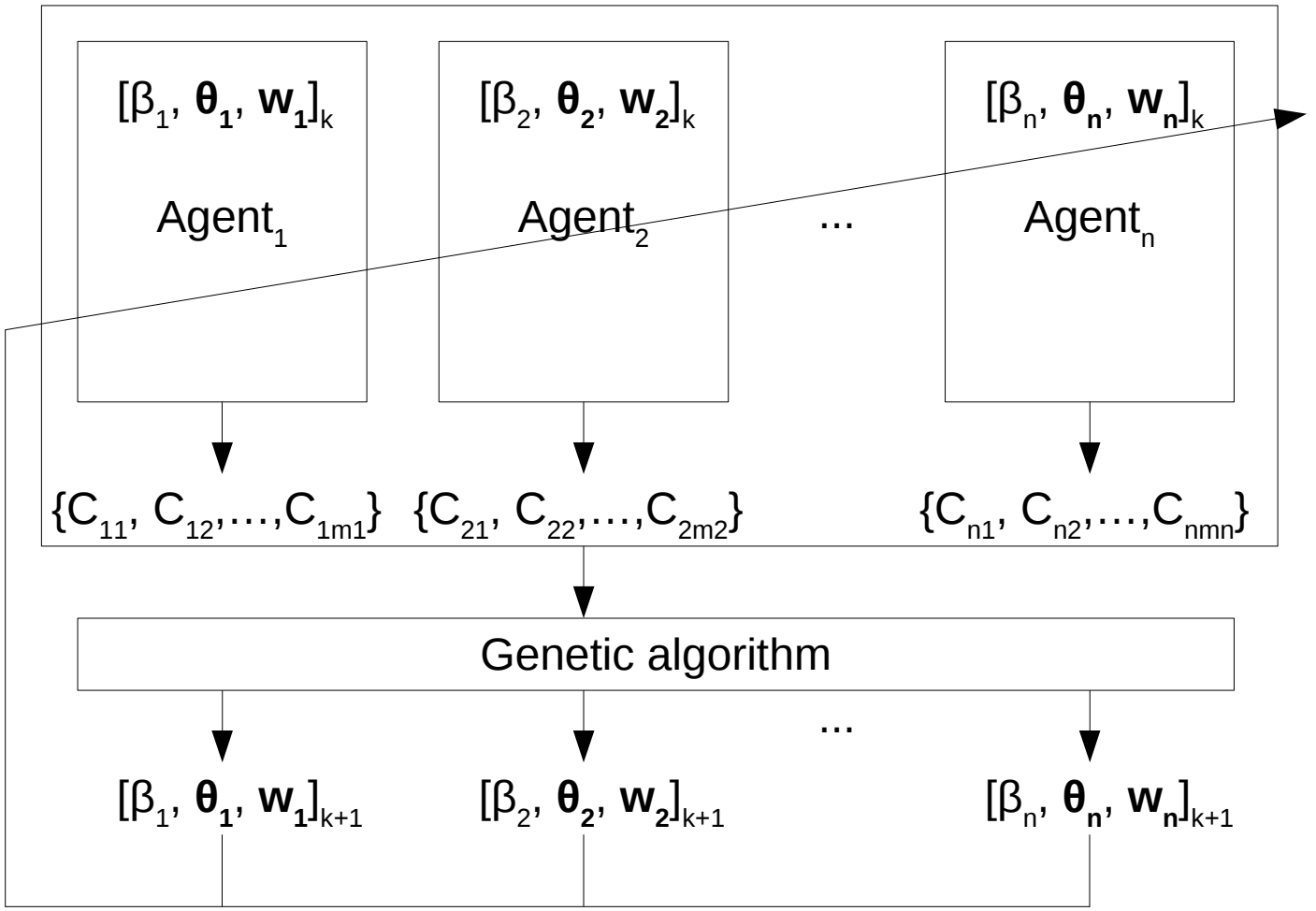
The algorithm: multi-agent structure



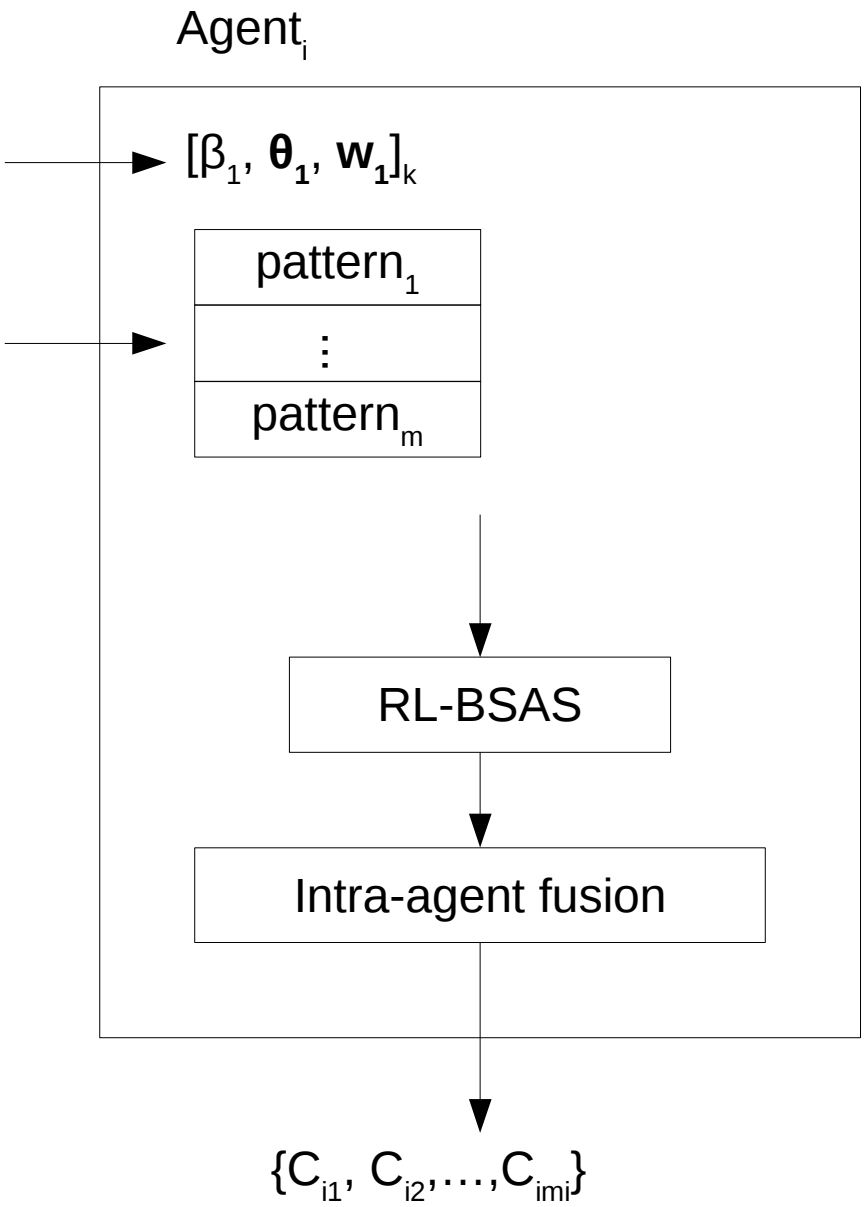
Each agent performs a simple clustering algorithm for a different dataset subsampling and with different parameters

The algorithm: genetic optimization

The genetic algorithm updates $[\beta_i, \boldsymbol{\theta}_i, \mathbf{w}_i]$ by ranking the agents with respect to a suitable fitness function

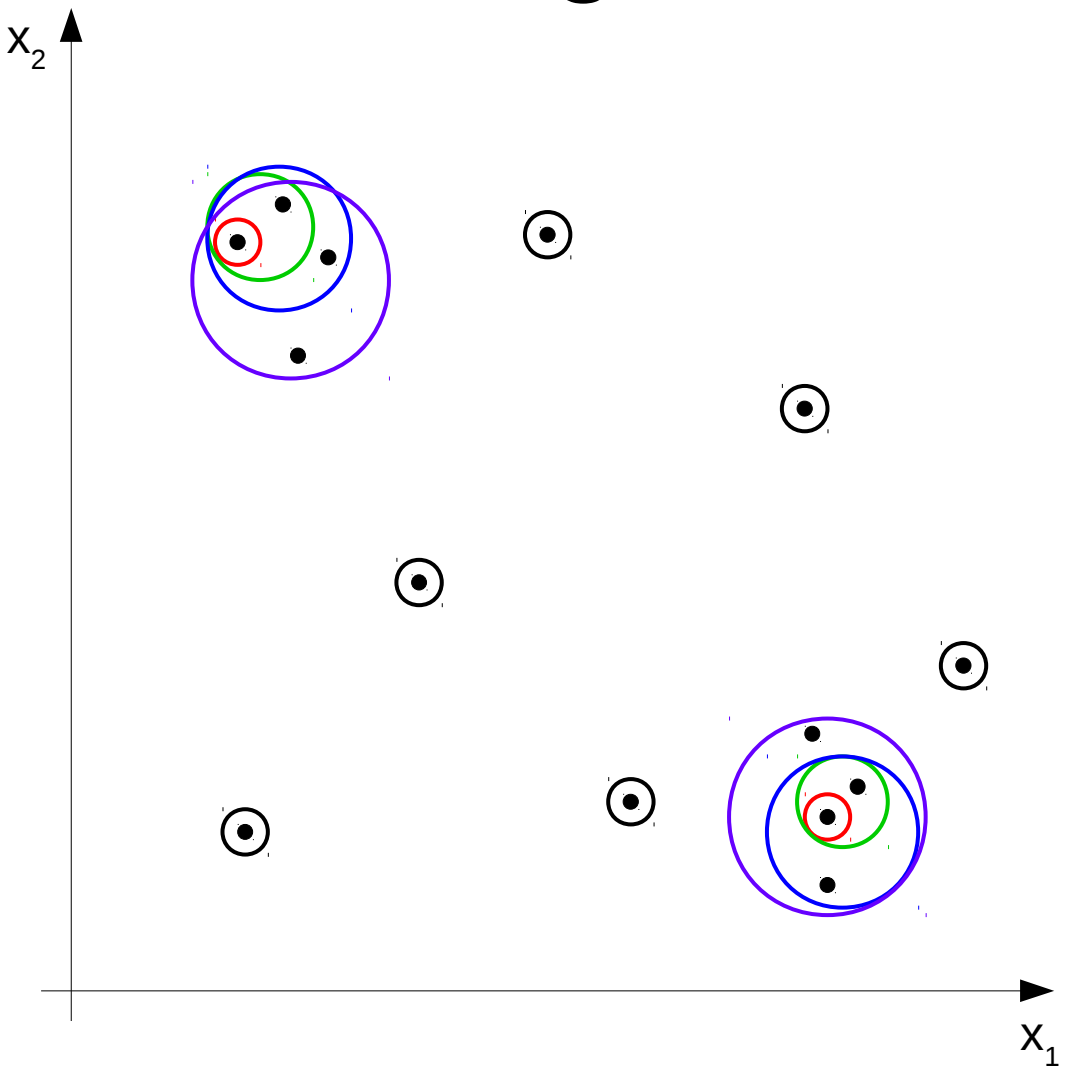


The algorithm: agent structure



Each agent performs RL-BSAS initialized with its own parameters on the input dataset subsample. Close clusters are merged before the output is returned

RL-BSAS (Basic Sequential Algorithmic Scheme)



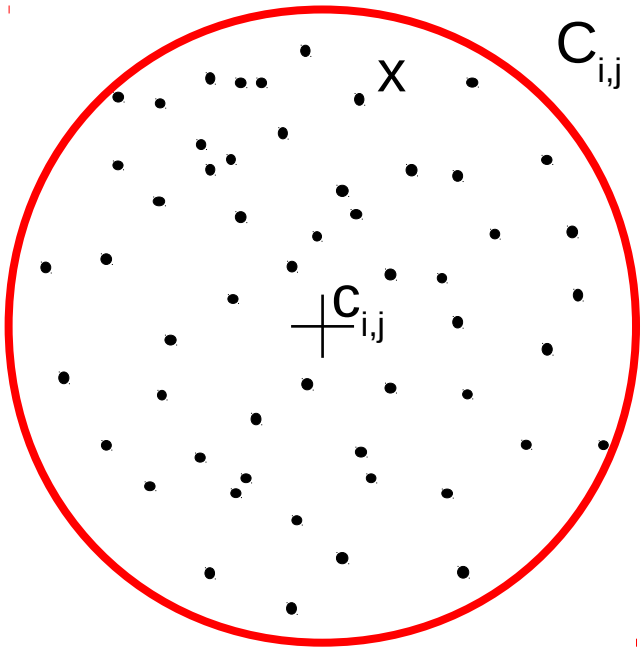
BSAS extracts a pattern and puts it in an existing cluster if close enough (below a treshold θ) or creates a new one otherwise. RL-BSAS deletes outlier made clusters which do not receive new patterns for a while (β deletion rate)

The algorithm: genetic optimization

$$f_{co}(C_{i,j}) = \frac{\left(1 - \frac{\sum_{x \in C_{i,j}} d(x, c_{i,j})}{|C_{i,j}|}\right) - co_{min}}{co_{max} - co_{min}}$$

The considered fitness function is a convex linear combination of compactness and cardinality

$$f_{ca}(C_{i,j}) = \frac{|C_{i,j}| - ca_{min}}{ca_{max} - ca_{min}}$$



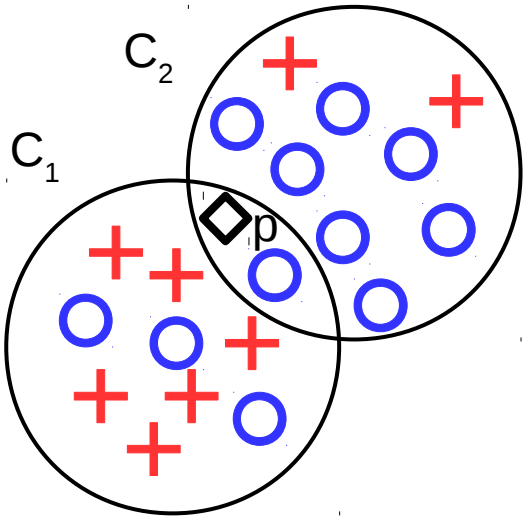
$$F_{cc}(C_{i,j}) = \lambda \cdot f_{co}(C_{i,j}) + (1 - \lambda) \cdot f_{ca}(C_{i,j})$$

E-ABC Classifier (E-ABC²): classification model

$$V_l = \sum_{i=1}^{I} v_{i,l}$$

$$l_{max} = \underset{l}{argmax}\, V_l$$

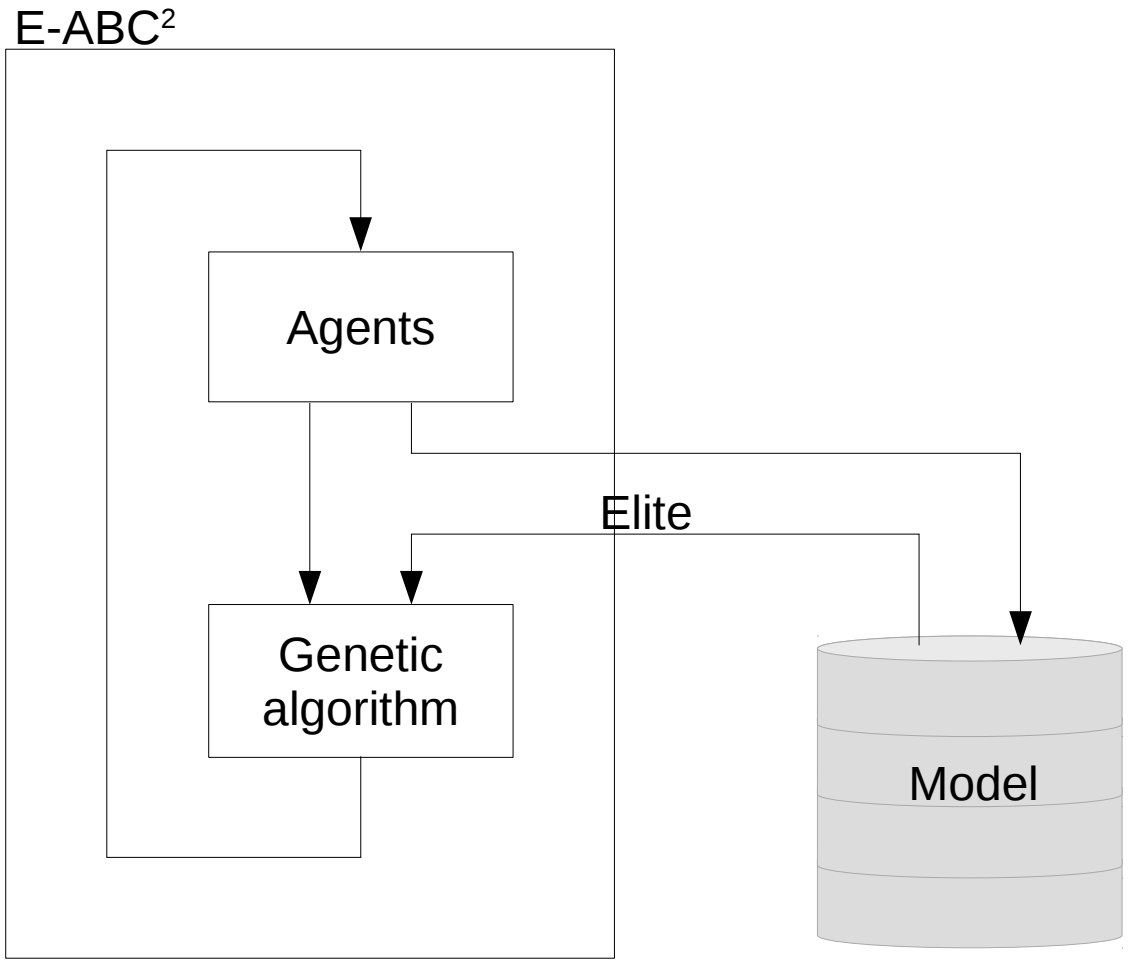
$$L_p = L_{l_{max}}$$



Cluster	Vote for [%] +	Vote for [%] o
C_1	0.6	0.4
C_2	0.2	0.8
tot	0.8	1.2

Each cluster votes for each class proportionally to its classes distribution. Votes are summed and the most voted label is associated to the pattern p to be classified

E-ABC²: classification model building



At each generation the best clusters in terms of accuracy are collected in the classification model. The clusters collected in the model are used to build the elite for the next generation

E-ABC²: genetic optimization

$$Acc(C_{i,j}) = \begin{cases} \dfrac{Cl^r_{C_{i,j}}}{Cl^t_{C_{i,j}}}, & Cl^t_{C_{i,j}} > 0 \\ 0, & Cl^t_{C_{i,j}} = 0 \end{cases}$$

$$Acc_{gl} = \frac{Cl^r_{gl}}{|S_v|}, \quad M_C = \{C_1, C_2, \ldots, C_m\}$$

$$F(C_{i,j}) = Acc_{gl} \cdot Acc_{C_{i,j}} + (1 - Acc_{gl}) \cdot F_{cc}(C_{i,j})$$

The clusters fitness is updated by taking into accout their classification accuracy and global model accuracy. Accuracy can be replaced with a more robust performances measure (e.g. F_1-score)

Experimental setup

Dataset provided by ACEA (Italian electric distribution network in Rome), related to faults of Rome power grid:

- 2080 patterns
- 17 features (quantitative, temporal, categorical, sequences)

Patterns equally distributed over two classes:

- Standard functioning state
- Localized fault

Evaluation over 10 dataset split in:

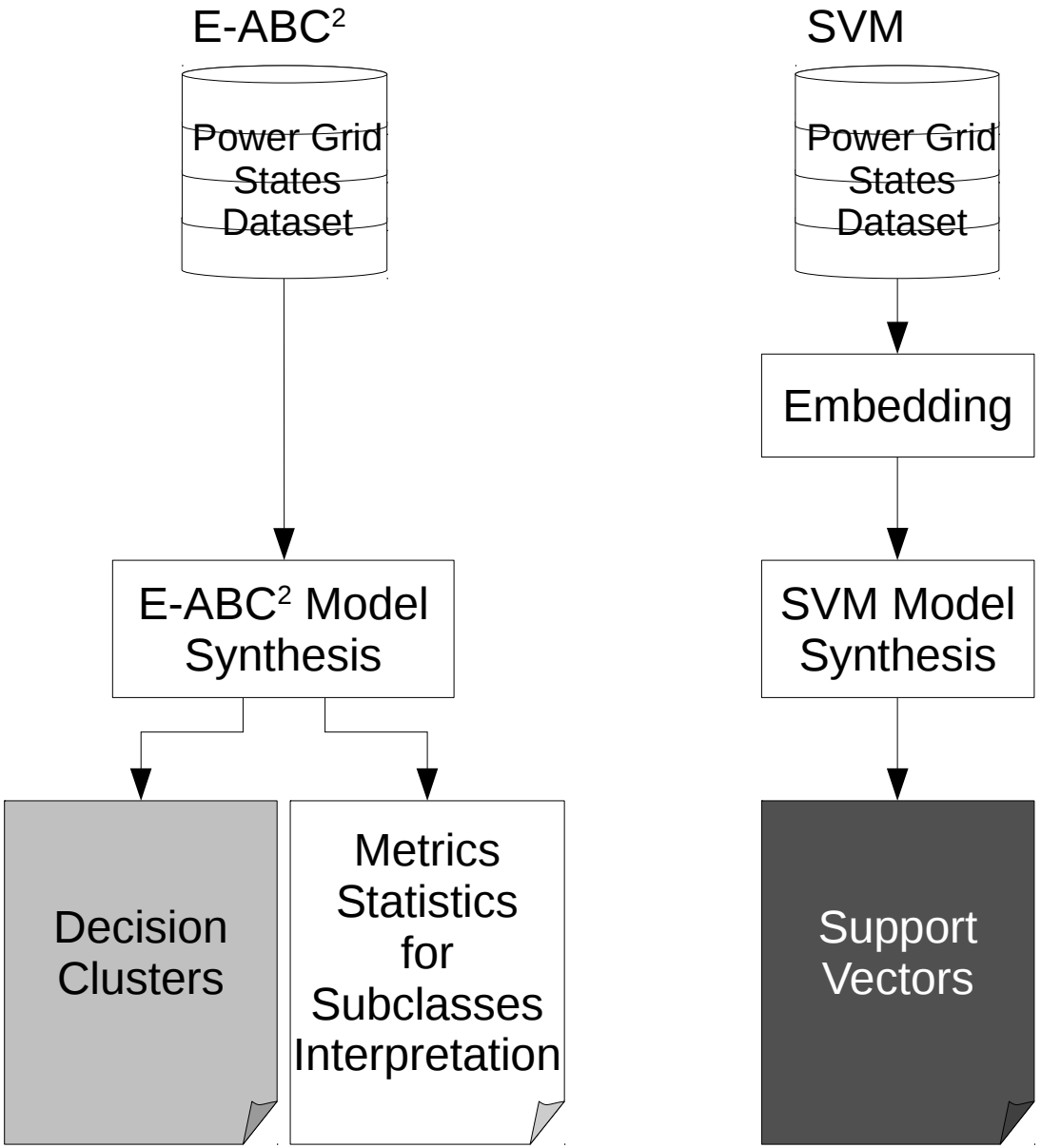
- Training set (50%)
- Validation set (25%)
- Test set (25%)

Pattern

Each pattern is composed of 17 unevenly typed features grouped in:

- Temporal – timestamp about local fault (LF)
- Physical power grid – equipments type, material, line voltage, current values
- Geographical – LF position, primary station position
- Weather – temperature max-min, humidity, mm of rain
- Event data (Time Series) – short outages

Workflow comparison



E-ABC² does not require embedding; it directly works in the features space: the result is a “grey box” model which is easily interpreted by Acea experts

Experimental results: evaluation metrics

$$Acc = \frac{\sum_{i=1}^{L} cm_{ii}}{\left|(S_{ts})\right|} \qquad Rec_i = \frac{cm_{ii}}{\sum_{j=1}^{L} cm_{ij}} \qquad Prec_i = \frac{cm_{ii}}{\sum_{j=1}^{L} cm_{ji}}$$

$$Spec_i = \frac{\sum_{j=1, j\neq i}^{L} cm_{jj}}{\sum_{j=1, j\neq i}^{L} cm_{jj} + \sum_{j=1, j\neq i}^{L} cm_{ji}}$$

$$F_{1i} = 2 \cdot \frac{Rec_i \cdot Prec_i}{Rec_i + Prec_i}$$

Five evaluation metrics:

- Accuracy
- Recall by class
- Precision by class
- Specificity by class
- F_1-score by class

Experimental results: performances

Analysis of 10 executions, with different random dataset splits, in terms of accuracy, recall, precision, specificity and F_1-score

	Accuracy	Recall	Precision	Specificity	F_1-score
T_1	0.906	0.790	0.991	0.995	0.879
T_2	0.842	0.707	0.907	0.945	0.795
T_3	0.918	0.810	1.000	1.000	0.895
T_4	0.890	0.745	1.000	1.000	0.854
T_5	0.984	0.962	1.000	1.000	0.981
T_6	0.909	0.790	1.000	1.000	0.882
T_7	0.952	0.893	0.996	0.997	0.942
T_8	0.936	0.855	0.996	0.997	0.920
T_9	0.970	0.945	1.000	0.990	0.965
T_{10}	0.976	0.945	0.810	1.000	0.972
Mean	0.928	0.844	0.988	0.992	0.909
Variance	0.002	0.008	0.001	0.000	0.003

Experimental results: comparative test

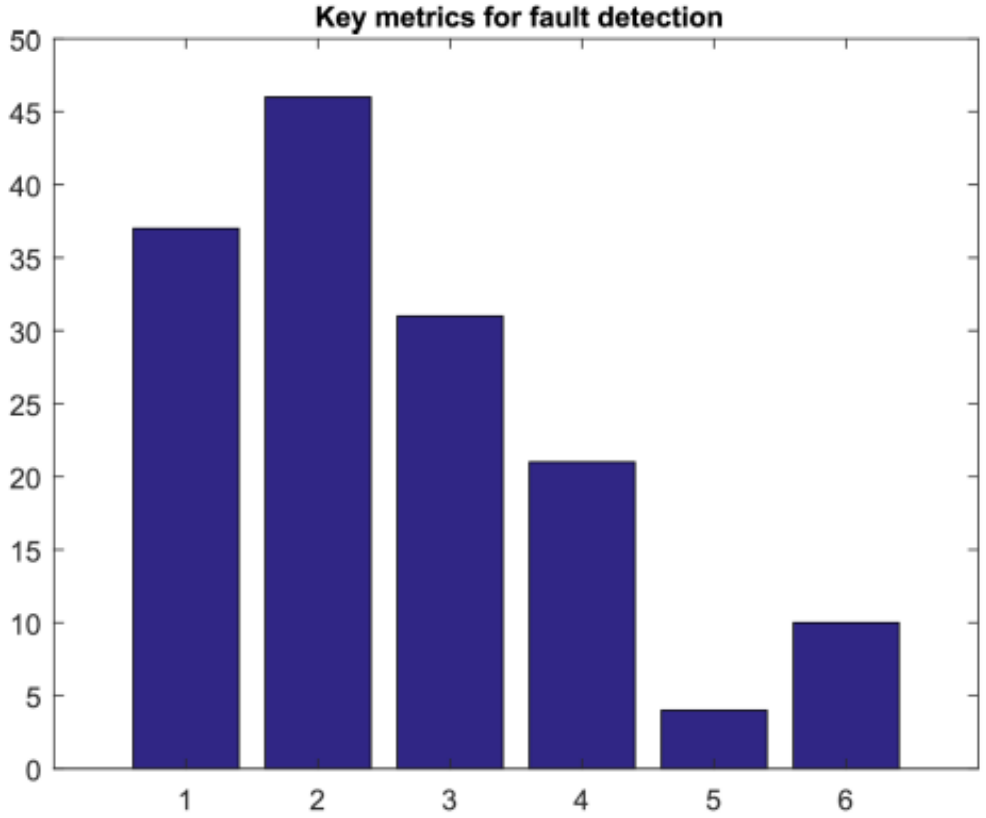
Results are comparable with those obtained through well-known algorithms

	Accuracy	Recall	Precision	Specificity	F_1-score
SVM	0.925	0.862	0.962	0.973	0.909
E-ABC^2	0.928	0.844	0.988	0.992	0.909

E-ABC^2 pros:

- Embedding not required
- “Grey box” model allowing results interpretation

Experimental results: statistics



1 ['Location', '#SS', 'Max Temp', 'Min Temp', 'Rain', 'Int Breaker ']

2 ['#SS', 'Max Temp', 'Min Temp', 'Delta Temp', 'BEC', 'Int Breaker ', 'Petersen Alarms']

3 ['PS-LF dist', 'Max Temp', 'Min Temp', 'Delta Temp', 'Rain', 'BEC']

4 ['Location', 'Material', 'Max Temp', 'Rain']

5 ['Location', 'Material', 'Current OFB', 'Max Temp', 'Rain']

6 ['PS-LF dist', '#SS', 'Max Temp', 'Min Temp', 'BEC', 'Petersen Alarms']

Statistics with the most common selected metrics:

- Temperatures
- Rain

Conclusions

- Good performances in terms of accuracy, recall, precision, specificity and F_1-score
- Scalable with respect to the dataset size
- Suitable approach for big data analysis
- Remarkable generalization capability in classification model synthesis
- Interpretable results provided by local metric learning

Work in progress

- Tests on datasets with higher complexity (number of patterns and number of features)
- Parallelize and distribute the agents
- Introduce a merging mechanism to reduce the number of clusters in the final model

Future works

- Improve speed and performance distributing agents and taking advantage of gossip protocol
- Synthesize the final model as a probability distribution function by means of kernel density estimation
- Tests with sequences and graphs datasets
- Compare performance when adopting different core clustering algorithms

Publications

[1] Martino A., Giampieri M., Luzi M., Rizzi A. (2019) Data Mining by Evolving Agents for Clusters Discovery and Metric Learning. In: Esposito A., Faundez-Zanuy M., Morabito F., Pasero E. (eds) Neural Advances in Processing Nonlinear Dynamic Signals. WIRN 2017 2017. Smart Innovation, Systems and Technologies, vol 102. Springer, Cham

[2] M. Giampieri, A. Rizzi. “An Evolutionary Agents Based System for Data Mining and Local Metric Learning”. Proceedings of the 19th International Conference on Industrial Technology February 20-22, Lyon, France. 2018.

[3] M. Giampieri, E. De Santis, A. Rizzi, F. M. Frattale Mascioli “A Supervised Classification System based on Evolutive Multi-Agent Clustering for Smart Grids Faults Prediction”. Proceedings of International Joint Conference on Neural Networks (IJCNN) 2018, Rio De Janeiro, Brazil, July 8-13, 2018, pp. 4359-4366.