Dynamic Classifier Selection: Recent advances and perspectives

Rafael M. O. Cruz[1], Robert Sabourin[1] and George D. C. Cavalcanti[2]
[1] Laboratoire d' imagerie, de vision et d' intelligence artificielle (LIVIA) - École de Technologie Supérieure (ÉTS), Canada
www.livia.etsmtl.ca
[2] Centro de informática – Universidade Federal de Pernambuco, Recife – Brazil
www.cin.ufpe.br/~viisar

Outline

- Part I
  1) Problem statement
  2) Multiple Classifier Systems
  3) Dynamic Selection
  4) Comparative study

- Part II
  1) Analysis with KNN
  2) FIRE-DES
  3) Applications of dynamic selection in different contexts
  4) Perspectives

The Oracle Model

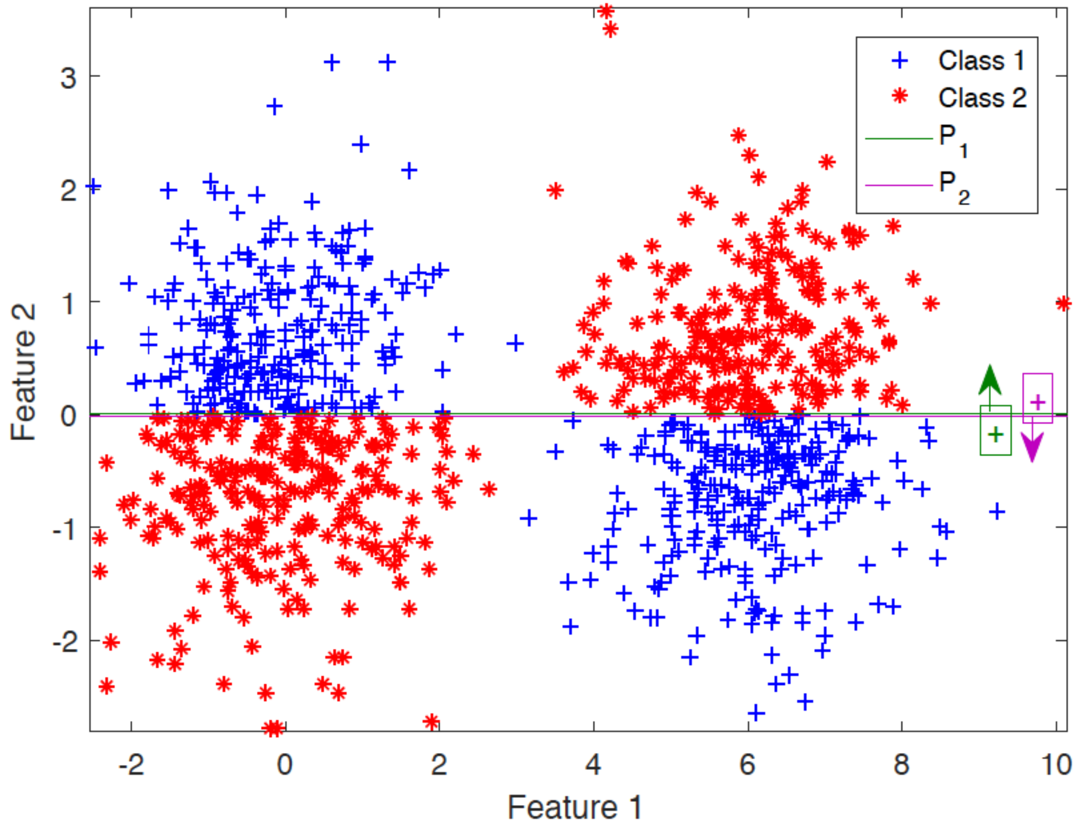
- **The Oracle** is an abstract model that always selects the base classifier that predict the correct label if such classifier exists

- Provides a **theoretical limit** for a DCS technique given a pool of classifiers.

- The Oracle has been used in the design of different MCS & DS methods

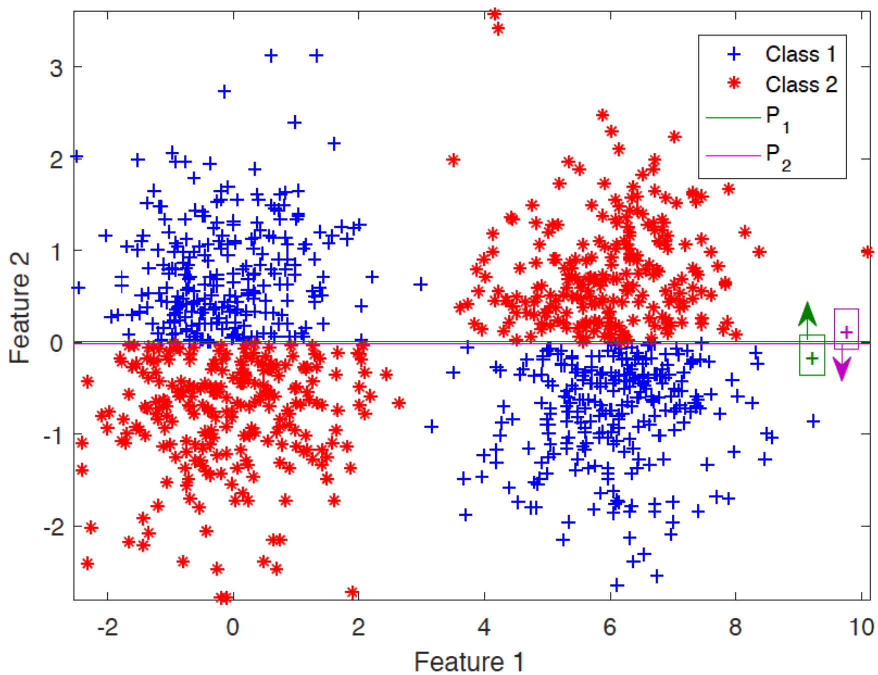
The Oracle Model

- Example of a two-class 2D problem. Perceptrons **P1** and **P2** have individual accuracy rates of 50%.



The Oracle Model

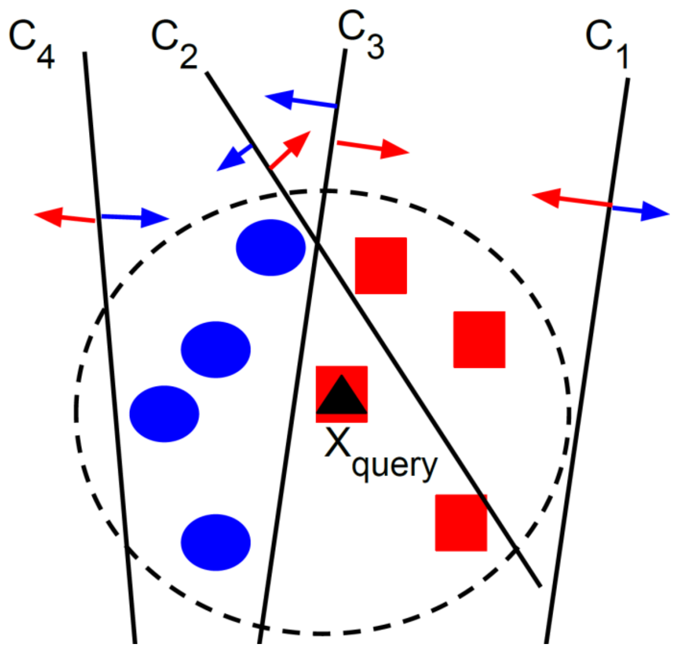
- Example of a two-class 2D problem. Perceptrons **P1** and **P2** have individual accuracy rates of 50%.



- Oracle accuracy rate of 100%, since it covers the entire feature space of this problem.

Dynamic Selection: how it works

- Instead of combining the output of all base classifiers, uses the one that is more competent locally according to the given query



- C1: 43% accuracy
- C2: 85% accuracy
- **C3: 100% accuracy**
- C4: 57% accuracy

- **Using DS C3** is select to predict the label of $\mathbf{x_j}$
- 7-NN would predict the wrong label (Blue Circle)

Objectives

- Present an updated Taxonomy of dynamic selection techniques
- Review of the state-of-the-art Dynamic Selection (DS) techniques
- Comparison of 18 dynamic selection techniques under the same experimental conditions
  - Experimental study focusing on classification only
- Discussion about the most recent findings in the field as well as perspectives for future research

Multiple-Classifier Systems (MCS): Introduction

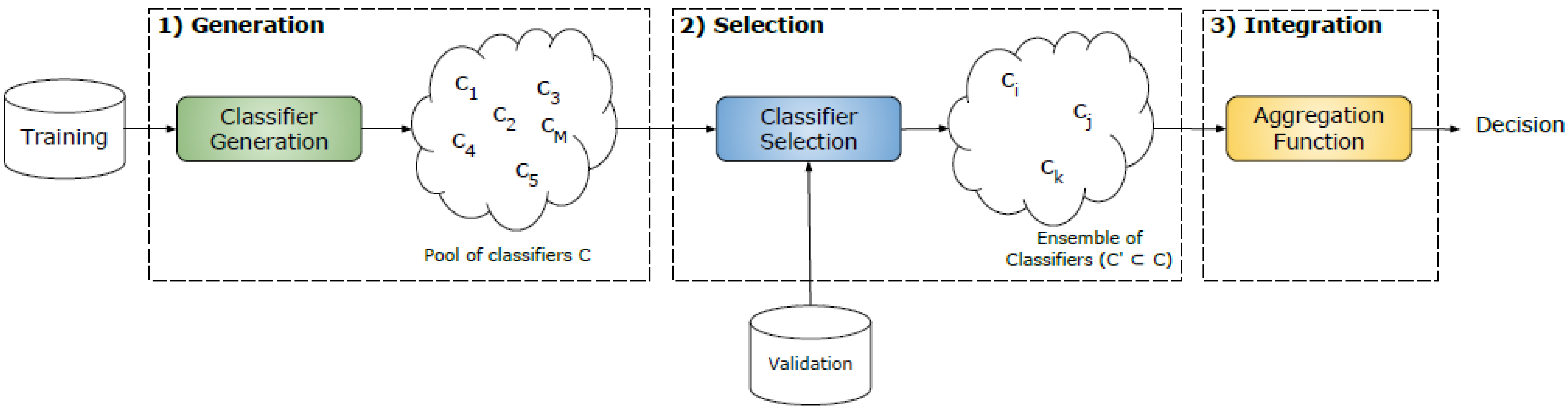
- There is **no clear guideline** to choose a good learning method
- Selecting the **best current classifier** can lead to the choice of the **worst classifier** for future data
- No Free Lunch Theorem
  - No dominant classifier exists for all the data distributions, and the data distribution of the task at hand is usually unknown

Multiple-Classifier Systems (MCS)

- MCS is often used for dealing with challenging problems where standard classification methods have been insufficient
  - Noisy data, changing environments, imbalanced distributions...
- Combine the outputs of several experts for a better prediction
  - In the hope that it will be better than the individual ones
- Several works have demonstrated its advantages over a single classifier model from a theoretical and empirical point of view
- One of the hottest areas in pattern recognition!
  - Used in several real world applications such as face recognition, credit scoring and intrusion detection...

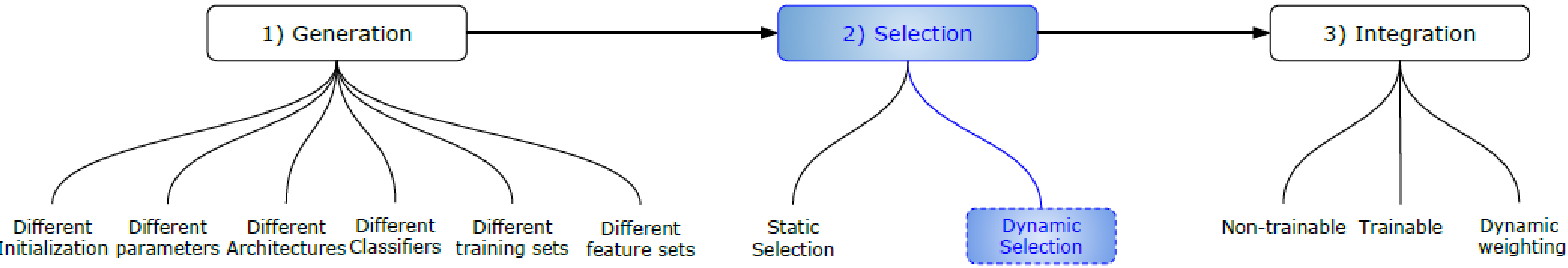
MCS phases

- Three phases of an MCS
  - 1) Generation
  - 2) Selection
  - 3) Integration



MCS: Taxonomy

- Research in MCS conducted in each of these stages
  - Generation
  - Selection
  - Integration



1) Generation

- Create a pool of classifiers C = {C_1, …, C_M} that are both accurate and diverse
  - Base classifiers should be different, since thee is no reason to combine experts that always presents the same output

- Six main strategies (Duin, ICPR 2002):
  - Different initialization
  - Different parameters
  - Different architectures
  - Different classifier models
  - Different training sets
  - Different feature sets

1) Generation

- Create a pool of classifiers $C = \{C_1, \ldots, C_M\}$ that are both accurate and diverse
  - Base classifiers should be different, since thee is no reason to combine experts that always presents the same output

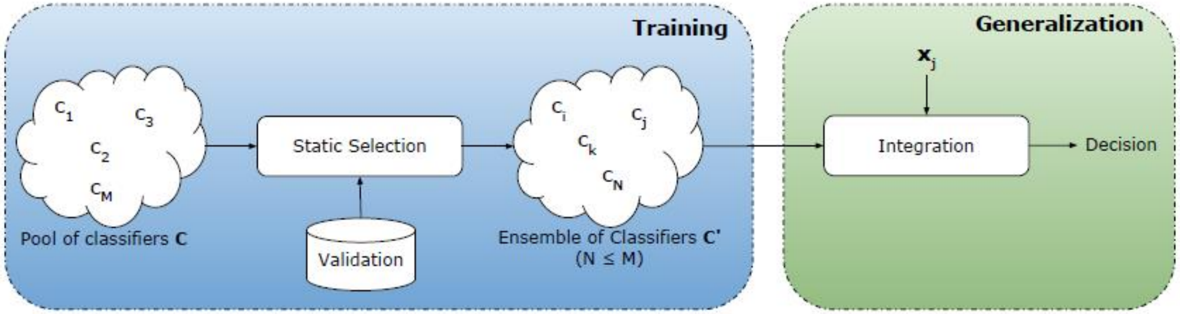
- Six main strategies (Duin, ICPR 2002):
  - Different initialization
  - Different parameters
  - Different architectures
  - **Different classifier models**
  - **Different training sets**
  - **Different feature sets**

Best ways of generating classifiers

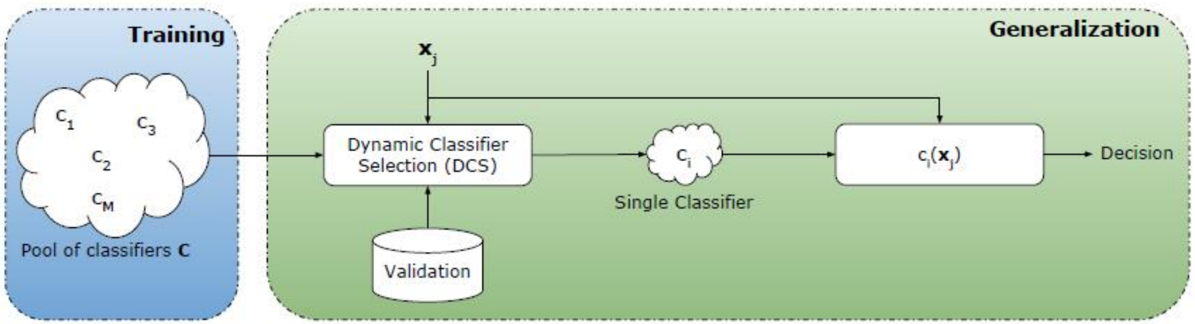
1) Generation

- Different classifier models
  - Also known as **Heterogeneous ensembles**
- Different training sets
  - Bagging, Boosting, clustering based classifier generation
- Different feature sets
  - Classifiers trained (e.g., image and music classification)
- More than one strategy can be used together to generate the pool of classifiers!

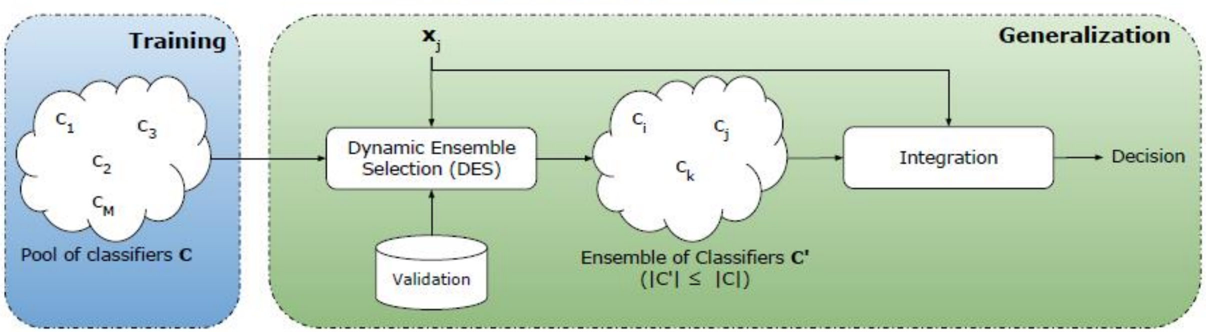
Static Selection VS Dynamic Selection



(a) Static Selection



(b) Dynamic Classifier Selection (DCS)



(c) Dynamic Ensemble Selection (DES)

2)Selection

- Can be either static or dynamic

- Static Selection: performed in the training stage, according to the performance of the base classifiers in either the training or validation data
  - Key concepts here are diversity and accuracy

- Dynamic selection: performed during the test phase according to each new query
  - Notion of local competence

Selection Stage

- Static Selection (SS)
  - An ensemble of classifiers (EoC) is selected based on the training or validation data
- Dynamic Classifier Selection (DCS)
  - A single classifier is selected according to each new test sample
- Dynamic Ensemble Selection (DES)
  - An EoC containing all the classifiers that attained a certain competence level is selected

3) Aggregation

- Non-trainable rules
  - Sum, Product, Median, Maximum, Majority Voting, Borda Count, etc…
- Trainable rules
  - A meta-classifier is used to learn the combination of classifiers
    - e.g., Mixture of experts
- Dynamic weighting
  - Weights of the base classifiers are determined according to each sample to be classified
    - Similar to dynamic selection
  - Can be used together with dynamic selection techniques

Dynamic selection (DS)

- **Dynamic selection** (DS) techniques rely on the assumption that each base classifier is an local expert
- DES works by measuring the **level of competence** of the classifiers considering **a new test sample**
- Only the **most competent classifiers** are selected to predict the label of a given query sample

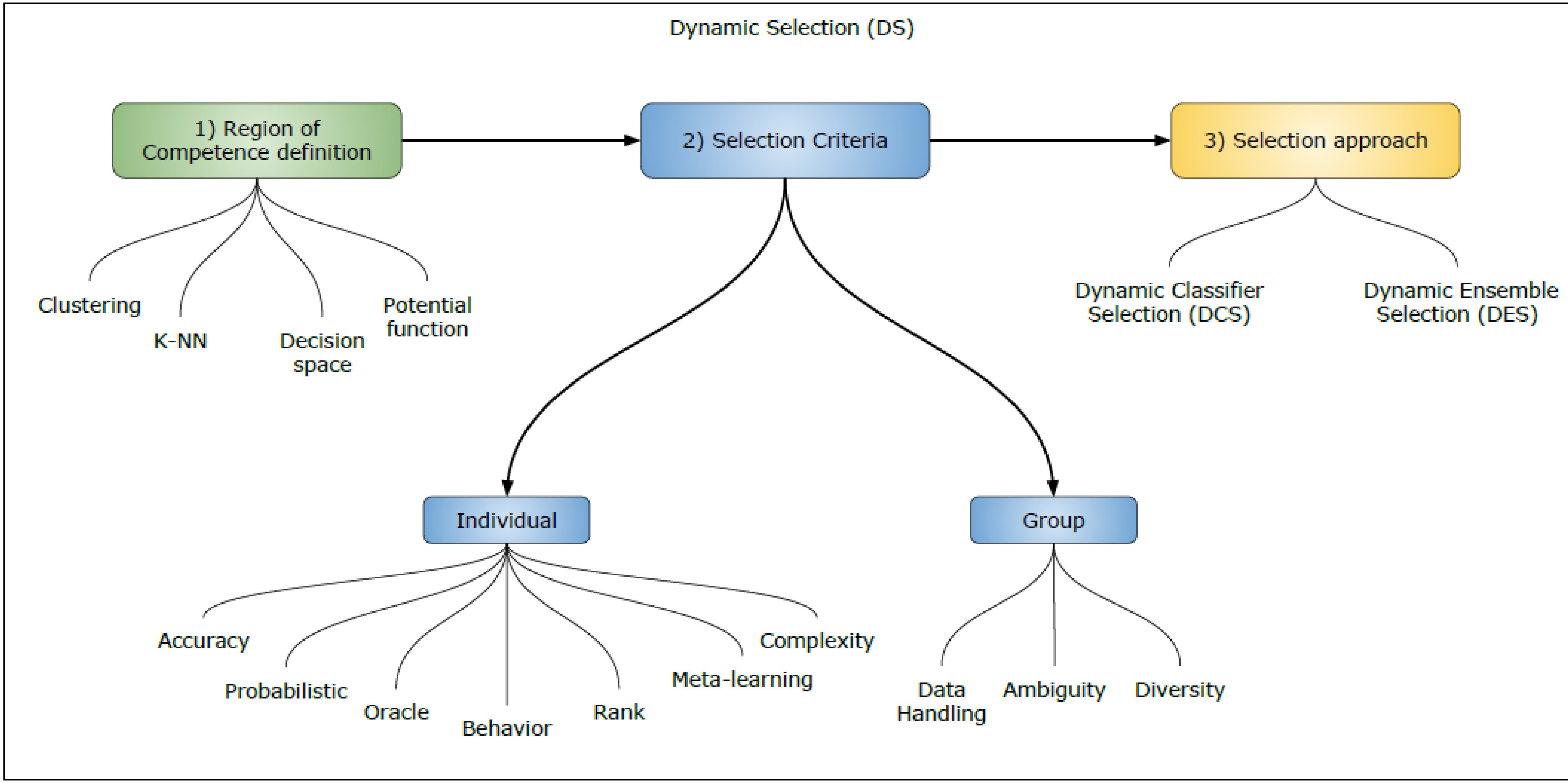
Dynamic selection (DS): Three steps

1. Definition of the region of competence
   - K-NN, clustering, decision space

2. Determination of a selection criteria used to measure the competence level of the base classifier
   - E.g., Accuracy, Ranking, Probabilistic, Oracle

3. Determination of the selection mechanism
   - How to select the most competent clasifier(s) based on their estimated competence level

Dynamic Selection: Taxonomy (Cruz et al., IF 2018)



1) Region of competence definition

- Local region used to estimate the competence level of the base classifiers

- Defined using a set of labeled data which can be either the training or validation set
  - This set is called dynamic selection dataset (DSEL)

- Can be defined using the following strategies:
  - Clustering
  - K-NN
  - Decision space
  - Potential function (Competence maps)

Clustering

- Define clusters in the dynamic selection set (DSEL)
  - E.g., K-Means
- For each cluster the competence level of the base classifiers is pre-calculated during the training phase
  - According to a selection criteria
- The competence is estimated taking into account the closest cluster to the query, $\mathbf{x}_j$

Clustering

- Faster during generalization since the competence level of the base classifiers are pre-calculated
  - Only the distance between the query and the cetroid of each cluster needs to be calculated

- Performance is very sensitive to the number of clusters

K-Nearest Neighbors

- Allows for a more precise estimation of the local region, leading to many different configurations of EoC when compared to Clustering
- The K-Nearest neighbors of the query, $\mathbf{x}_j$, is calculated on DSEL
- The selection criteria is applied over the neighborhood of $\mathbf{x}_j$
- Costly for large datasets due to the calculation of the nearest neighbors

Decision space

- Similarity between samples are estimated based on the decisions obtained by the base classifier

- Samples are transformed to the decision space using output profiles

- The output profile of a sample $\mathbf{x}_j$ is denoted by where each element is the decision yielded by the base classifier c_i for $\mathbf{x}_j$
  - Can be either the hard decision (class label) or soft (probabilities) predicted

- Costly for large pool of classifiers and datasets

Competence map (Potential function)

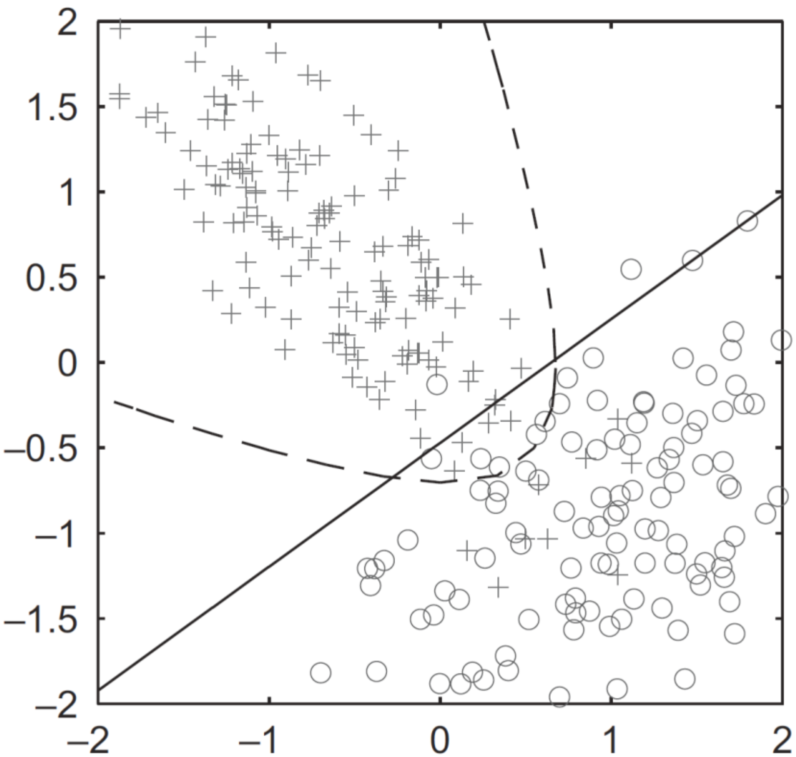
- Uses the whole DSEL for the estimation of competence
- Influence of each neighbor decreases according to a Gaussian potential function
  - Samples that are closer to the query have a higher influence in the computation of competence

$$K(\mathbf{x}_k, \mathbf{x}_j) = exp(-d(\mathbf{x}_k, \mathbf{x}_j)^2)$$

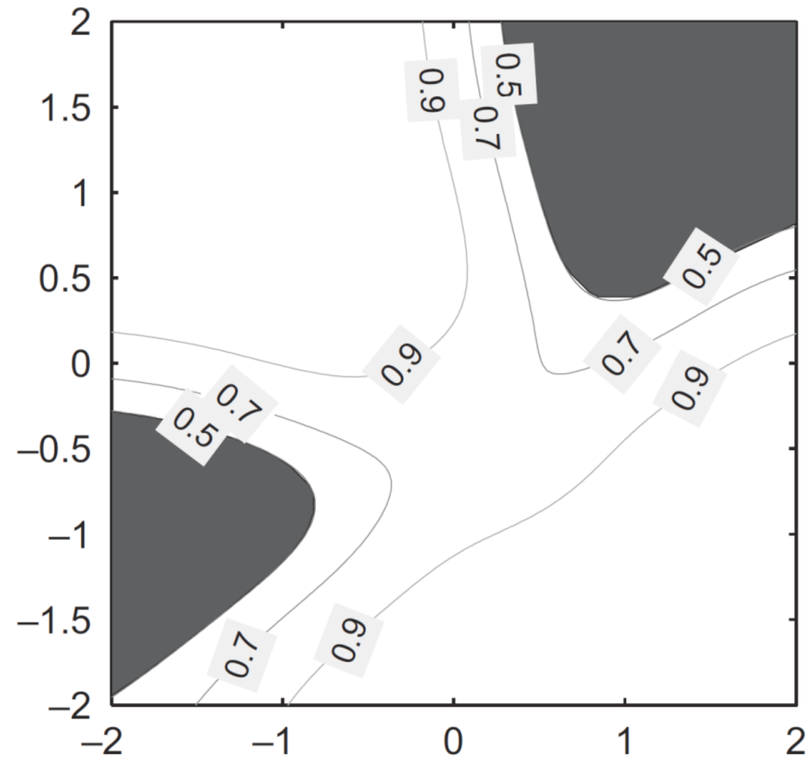
- No need to define a neighborhood size a priori
- Costly due to the fact the competence measure takes into account each data point in DSEL

Competence map (Potential function)

- Competence map calculated using the potential function model



(a)



(b)

2) Selection criteria

- The criteria used to measure the competence level of the base classifiers
- Applied only over the defined region of competence
- Can be either Individual or group based

2) Selection criteria

- Individual measures are related to the concept of competence
  - Whether or not the base classifier is competent enough for the classification of the query
  - Measure the competence of the base classifier independently of the rest of the pool

- Groups based are related to the concept of relevance
  - Whether the base classifier is complimentary to the rest of classifiers
  - Is usually applied as a post-processing after the most competent classifiers is selected
    - E.g., select the most competent classifiers first, then add the most diverse classifiers to the pre-selected EoC

2) Selection criteria: individual based

- The competence of each base classifier is measured individually in the pre-defined region of competence
  - Concept of local competence
- There are several individual based criteria:
  - Accuracy
  - Ranking
  - Oracle
  - Probabilistic
  - Behavior
  - Data complexity
  - Meta-learning*

*Meta-learning presents a different perspective of how the competence of the base classifiers can be "learned" based on different criteria

2) Selection criteria: group based

- Take into account the interaction of the members of the pool
  - E.g., Diversity among the base classifiers

- Concept of relevance
  - Whether the base classifier works well in conjunction to with the other classifiers in the ensemble

- Used after a initial pool is selected using a individual based measure
  - E.g., select the most diverse classifiers among the selected classifier ensemble

3) Selection mechanism

- The rule used to the select the base classifiers according to their estimated competence level
  - **Overall Local Accuracy (OLA):** Only the classifier with the highest accuracy (competence) in the local region is selected
  - **DES-Performance (DES-P):** All base classifiers that obtained a competence level higher than the random classifier is selected (DES-Performance)
  - **KNORA-Eliminate (KNORA-E):** Only the classifiers with a perfect accuracy in the region of competence are selected

- A single classifier can be selected (DCS) or an ensemble of classifiers can be selected (DES)

Dynamic selection techniques

- Most well-known dynamic selection techniques from 1993 to 2016
  - Different concepts are considered based on the proposed taxonomy

Technique	Region of competence definition	Selection criteria	Selection Approach	Reference	Year
Classifier Rank (DCS-Rank)	K-NN	Ranking	DCS	Sabourin et al. [33]	1993
Overall Local Accuracy (OLA)	K-NN	Accuracy	DCS	Woods et al.[31]	1997
Local class accuracy (LCA)	K-NN	Accuracy	DCS	Woods et al.[31]	1997
A Priori	K-NN	Probabilistic	DCS	Giacinto[129]	1999
A Posteriori	K-NN	Probabilistic	DCS	Giacinto[129]	1999
Multiple Classifier Behavior (MCB)	K-NN	Behavior	DCS	Giacinto et al.[112]	2001
Modified Local Accuracy (MLA)	K-NN	Accuracy	DCS	P.C. Smits[32]	2002
DES-Clustering	Clustering	Accuracy & Diversity	DES	Soares et al.[30, 116]	2006
DES-KNN	K-NN	Accuracy & Diversity	DES	Soares et al.[30, 116]	2006
K-Nearest Oracles Eliminate (KNORA-E)	K-NN	Oracle	DES	Ko et al.[26]	2008
K-Nearest Oracles Union (KNORA-U)	K-NN	Oracle	DES	Ko et al.[26]	2008
Randomized Reference Classifier (RRC)	Potential function	Probabilistic	DES	Woloszynski et al.[25]	2011
Kullback-Leibler (DES-KL)	Potential function	Probabilistic	DES	Woloszynski et al.[34]	2012
DES Performance (DES-P)	Potential function	Probabilistic	DES	Woloszynski et al.[34]	2012
K-Nearest Output Profiles (KNOP)	K-NN	Behavior	DES	Cavalin et al.[106]	2013
META-DES	K-NN	Meta-Learning	DES	Cruz et al.[27]	2015
META-DES.Oracle	K-NN	Meta-Learning	DES	Cruz et al.[130]	2016
Dynamic Selection On Complexity (DSOC)	K-NN	Accuracy & Complexity	DCS	Brun et al.[123]	2016

Dynamic selection techniques

- Modified classifier rank
- Overal local accuracy (OLA)
- Local class accuracy (LCA)
- A Priori
- A Posteriori
- Multiple Classifier Behavior (MCB)
- DES-Clustering
- DES-KNN
- K-Nearest Oracles-Eliminate (KNORA-E)
- K-Nearest Oracles Union (KNORAU)
- DES-RRC
- DES-Performance
- DES-Kullback Leibler (DES-KL)
- K-Nearest Output Profiles (KNOP)

The META-DES Framework

The **dynamic selection mechanism** can be defined as a **meta-problem**:

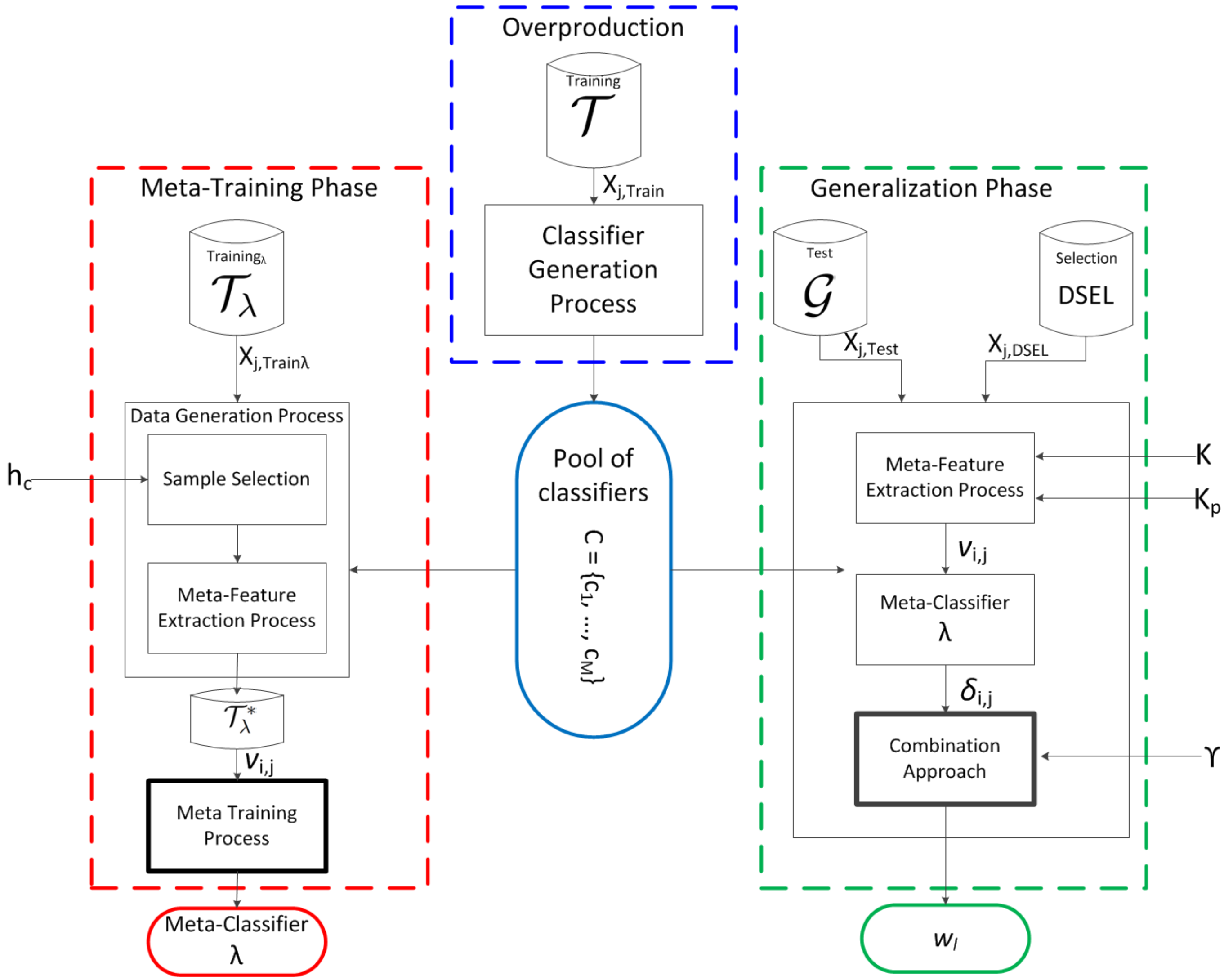
- The **meta-classes** of this meta-problem are either "competent" (1) or "incompetent" (0) to classify a given query sample $\mathbf{x}_j$

- Each set of **meta-features** f_i corresponds to a different criterion for measuring the level of competence of a base classifier

- The meta-features are encoded into a **meta-features vector** $v_{i,j}$

- A **meta-classifier** is trained based on the meta-features $v_{i,j}$ to predict whether or not c_i will achieve the correct prediction for $\mathbf{x}_j$, i.e., if it is competent enough to classify $\mathbf{x}_j$

META-DES general architecture

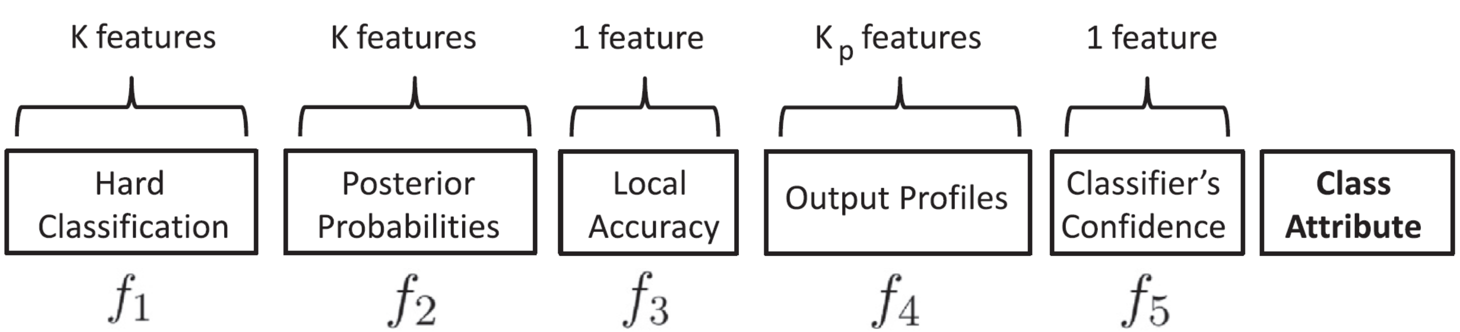


META-DES: Meta-features

- **Multiple criteria** are used to estimate the competence of base classifiers
  - These criteria are encoded as meta-features

Table 1
Relationship between each meta-features and different paradigms to compute the level of competence of a base classifier.

Meta-feature	Criterion	Paradigm
f_1	Local accuracy in the region of competence	Classifier accuracy over a local region
f_2	Extent of consensus in the region of competence	Classifier consensus
f_3	Overall accuracy in the region of competence	Accuracy over a local region
f_4	Accuracy in the decision space	Decision templates
f_5	Degree of confidence for the input sample	Classifier confidence



META-DES: Meta-features

- **f_1 - Neighbors' hard classification:** For each sample $\mathbf{x}_k$, belonging to the region of competence θ_j , if c_i correctly classifies $\mathbf{x}_k$, the k-th position of the vector is set to 1, otherwise it is 0
- **f_2 - Posterior Probability:** For each sample $\mathbf{x}_k$, belonging to the region of competence θ_j, the posterior probability of c_i , $P(w_l \mid \mathbf{x}_k)$ is computed
- **f_3 - Overall Local Accuracy:** The accuracy of c_i over the whole region of competence θ_j
- **f_4 - Outputs' profile classification:** For each member $\tilde{\mathbf{x}}_k$ belonging to the set of output profiles ϕ_j , if the label produced by c_i for $\mathbf{x}_k$ is equal to the label $w_{l,k}$ of $\tilde{\mathbf{x}}_k$, the k-th position of the vector is set to 1, otherwise it is set to 0
- **f_5 - Classifier's confidence:** The perpendicular distance between the reference sample $\mathbf{x}_j$ and the decision boundary of the base classifier c_i

META-DES: Meta-features

- **f_1 - Neighbors' hard classification:** For each sample $\mathbf{x}_k$, belonging to the region of competence θ_j , if c_i correctly classifies $\mathbf{x}_k$, the k-th position of the vector is set to 1, otherwise it is 0
- **f_2 - Posterior Probability:** For each sample $\mathbf{x}_k$, belonging to the region of competence θ_j, the posterior probability of c_i , $P(w_l \mid \mathbf{x}_k)$ is computed
- **f_3 - Overall Local Accuracy:** The accuracy of c_i over the whole region of competence θ_j
- **f_4 - Outputs' profile classification:** For each member $\tilde{\mathbf{x}}_k$ belonging to the set of output profiles ϕ_j , if the label produced by c_i for $\mathbf{x}_k$ is equal to the label $w_{l,k}$ of $\tilde{\mathbf{x}}_k$, the k-th position of the vector is set to 1, otherwise it is set to 0
- **f_5 - Classifier's confidence:** The perpendicular distance between the reference sample $\mathbf{x}_j$ and the decision boundary of the base classifier c_i

META-DES: Meta-features

- **f_1 - Neighbors' hard classification:** For each sample $\mathbf{x}_k$, belonging to the region of competence θ_j , if c_i correctly classifies $\mathbf{x}_k$, the k-th position of the vector is set to 1, otherwise it is 0
- **f_2 - Posterior Probability:** For each sample $\mathbf{x}_k$, belonging to the region of competence θ_j, the posterior probability of c_i , $P(w_l \mid \mathbf{x}_k)$ is computed
- **f_3 - Overall Local Accuracy:** The accuracy of c_i over the whole region of competence θ_j
- **f_4 - Outputs' profile classification:** For each member $\tilde{\mathbf{x}}_k$ belonging to the set of output profiles ϕ_j , if the label produced by c_i for $\mathbf{x}_k$ is equal to the label $w_{l,k}$ of $\tilde{\mathbf{x}}_k$, the k-th position of the vector is set to 1, otherwise it is set to 0
- **f_5 - Classifier's confidence:** The perpendicular distance between the reference sample $\mathbf{x}_j$ and the decision boundary of the base classifier c_i

META-DES: Meta-features

- f_1 - **Neighbors' hard classification:** For each sample $\mathbf{x}_k$, belonging to the region of competence θ_j, if c_i correctly classifies $\mathbf{x}_k$, the k-th position of the vector is set to 1, otherwise it is 0
- f_2 - **Posterior Probability:** For each sample $\mathbf{x}_k$, belonging to the region of competence θ_j, the posterior probability of c_i, $P(w_l \mid \mathbf{x}_k)$ is computed
- f_3 - **Overall Local Accuracy:** The accuracy of c_i over the whole region of competence θ_j
- $\mathbf{f_4}$ - **Outputs' profile classification:** For each member $\tilde{\mathbf{x}}_k$ belonging to the set of output profiles ϕ_j, if the label produced by c_i for $\mathbf{x}_k$ is equal to the label $w_{l,k}$ of $\tilde{\mathbf{x}}_k$, the k-th position of the vector is set to 1, otherwise it is set to 0
- f_5 - **Classifier's confidence:** The perpendicular distance between the reference sample $\mathbf{x}_j$ and the decision boundary of the base classifier c_i

META-DES: Meta-features

- **f_1 - Neighbors' hard classification:** For each sample $\mathbf{x}_k$, belonging to the region of competence θ_j , if c_i correctly classifies $\mathbf{x}_k$, the k-th position of the vector is set to 1, otherwise it is 0
- **f_2 - Posterior Probability:** For each sample $\mathbf{x}_k$, belonging to the region of competence θ_j, the posterior probability of c_i , $P(w_l \mid \mathbf{x}_k)$ is computed
- **f_3 - Overall Local Accuracy:** The accuracy of c_i over the whole region of competence θ_j
- **f_4 - Outputs' profile classification:** For each member $\tilde{\mathbf{x}}_k$ belonging to the set of output profiles ϕ_j , if the label produced by c_i for $\mathbf{x}_k$ is equal to the label $w_{l,k}$ of $\tilde{\mathbf{x}}_k$, the k-th position of the vector is set to 1, otherwise it is set to 0
- **f_5 - Classifier's confidence:** The perpendicular distance between the reference sample $\mathbf{x}_j$ and the decision boundary of the base classifier c_i

META-DES: (2) Meta-Training

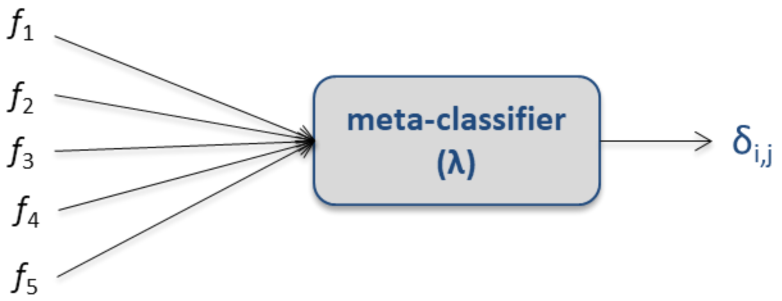
- Four classifier models are considered for the meta-classifier λ
- A radial basis SVM with a Gaussian Kernel
  - A grid search was performed to set the values of the regularization parameter c and the Kernel spread γ
- Random Forest
  - 200 decision trees. The depth of each tree set was fixed as 5
- Multinomial Naïve Bayes classifier
- MLP Neural Network

META-DES: (2) Meta-Training

- Four classifier models are considered for the meta-classifier λ
- A radial basis SVM with a Gaussian Kernel
  - A grid search was performed to set the values of the regularization parameter *c* and the Kernel spread *γ*
- Random Forest
  - 200 decision trees. The depth of each tree set was fixed as 5
- **Multinomial Naïve Bayes classifier**
- MLP Neural Network

META-DES: (3) Generalization

- Given a new test sample $\mathbf{x}_j$
- For each new base classifier, c_i the five sets of meta-features are computed, and used as inputs to the meta-classifier
  - The outputs of the meta-classifier is the level of competence, $\delta_{i,j}$



META-DES: (3) Generalization

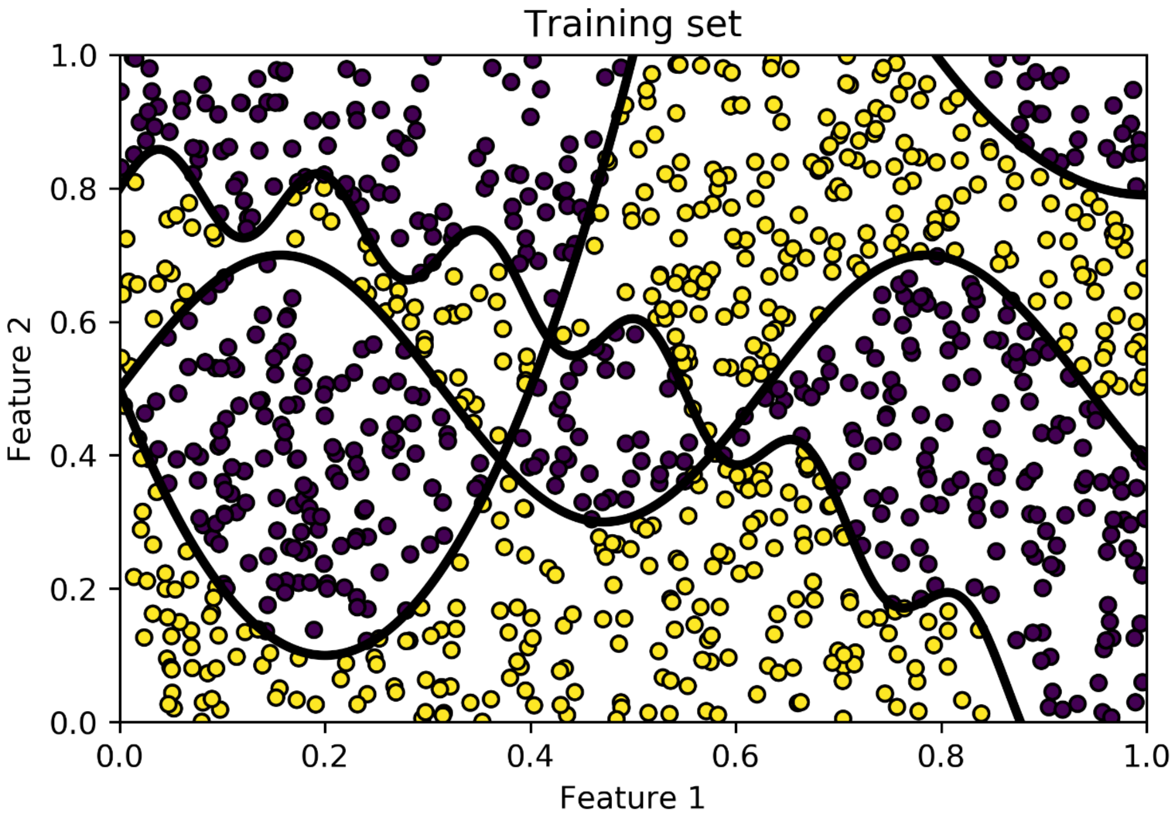
- **META-DES.S:** The classifiers that achieve a level of competence $\delta_{i,j} > \Upsilon$, are considered competent and are selected to form the ensemble C'. The final decision is obtained by a majority voting scheme.

- **META-DES.W:** Every classifier in the pool is used to predict the label of $\mathbf{x}_{j,test}$. The decision of each base classifier is weighted based on its level of competence $\delta_{i,j}$

- **META-DES.H*:** First an ensemble C' is selected. Next, the level of competence $\delta_{i,j}$ for the classifiers in C' are used as its weights in a weighted majority voting scheme

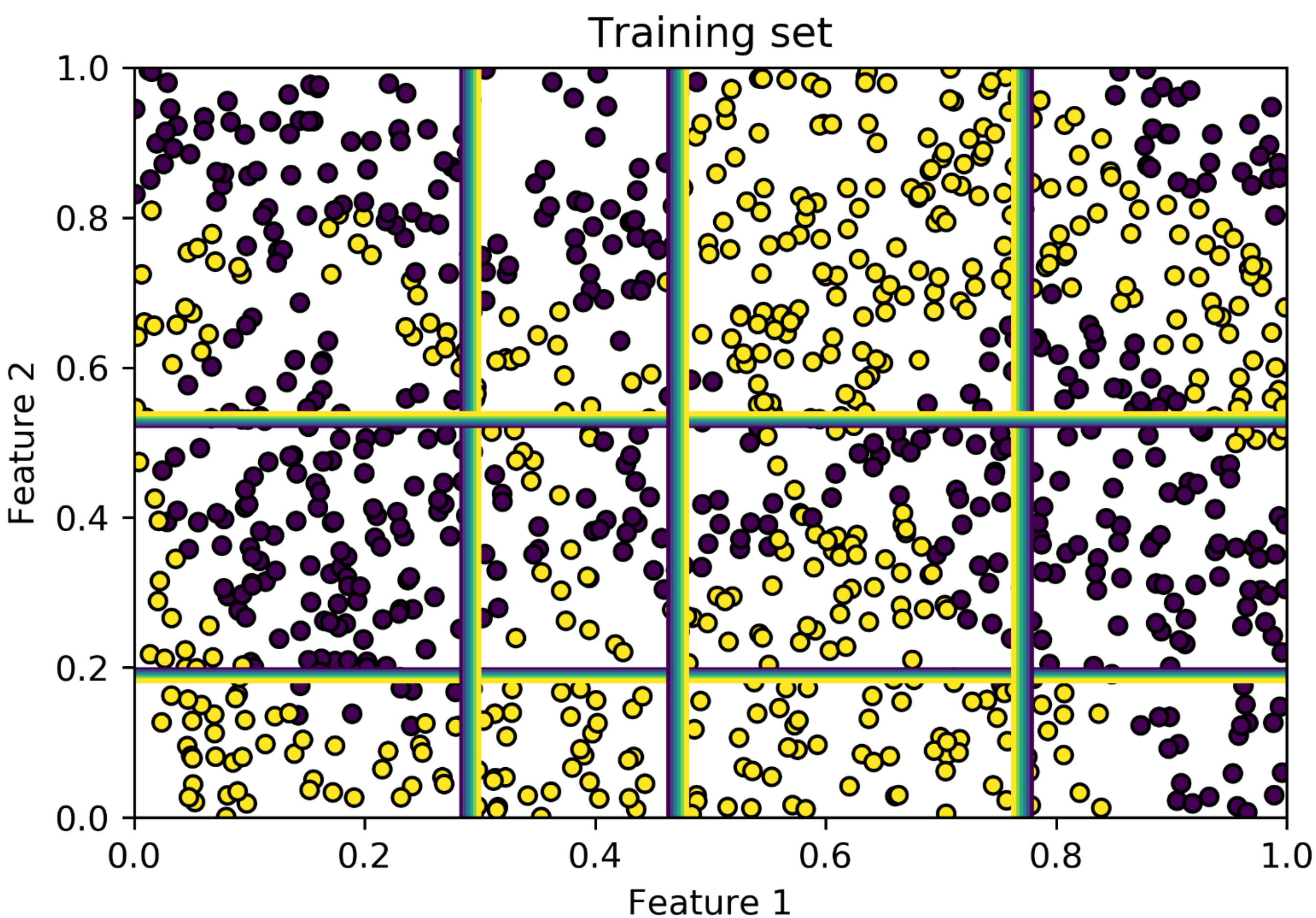
Case study: The P2 problem

- Classification problem with a very complex decision border with multi-modal classes
  - $\varepsilon_t = 0\%$ and $p(\omega_1) = p(\omega_2)$

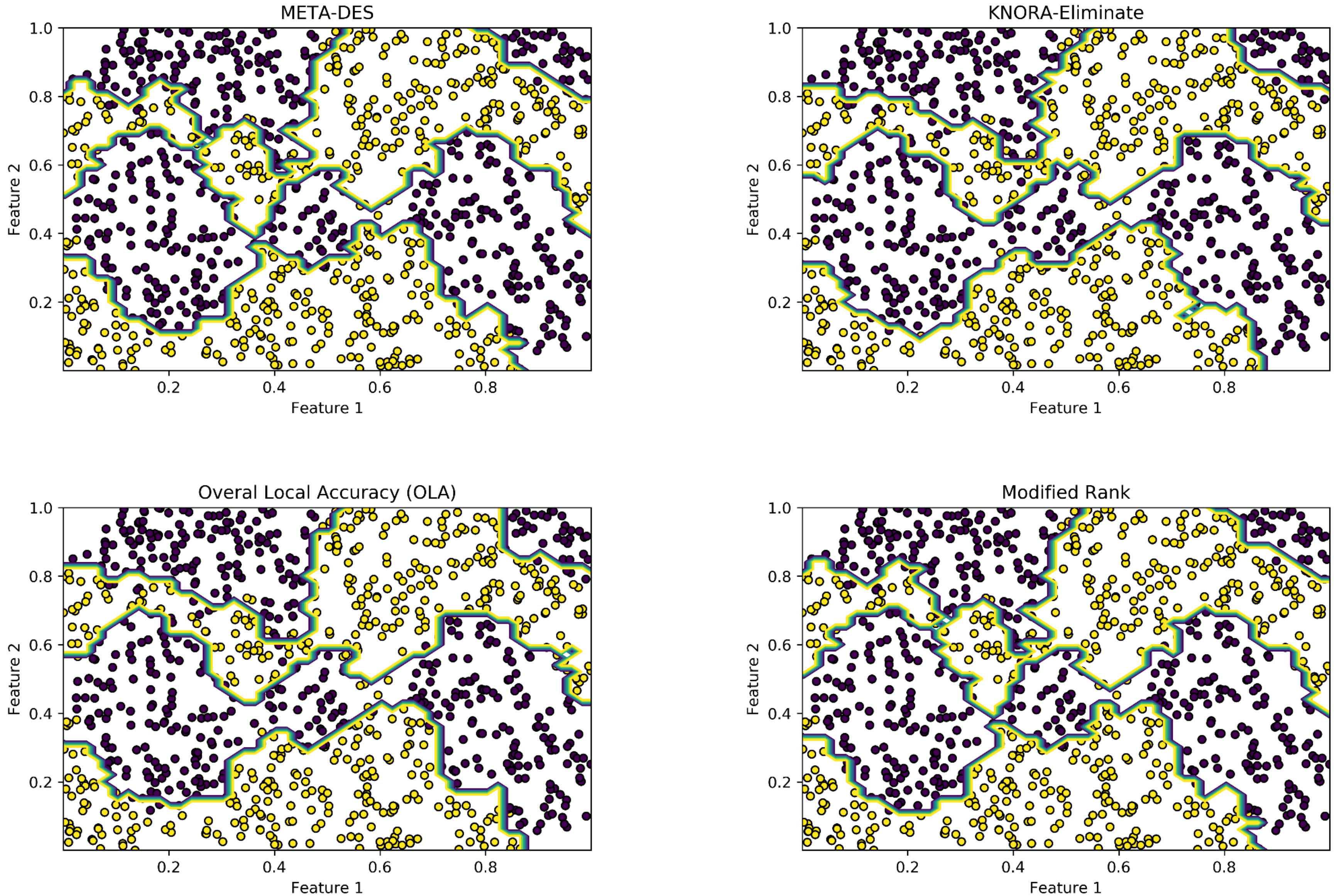


Case study

- Five Linear classifiers

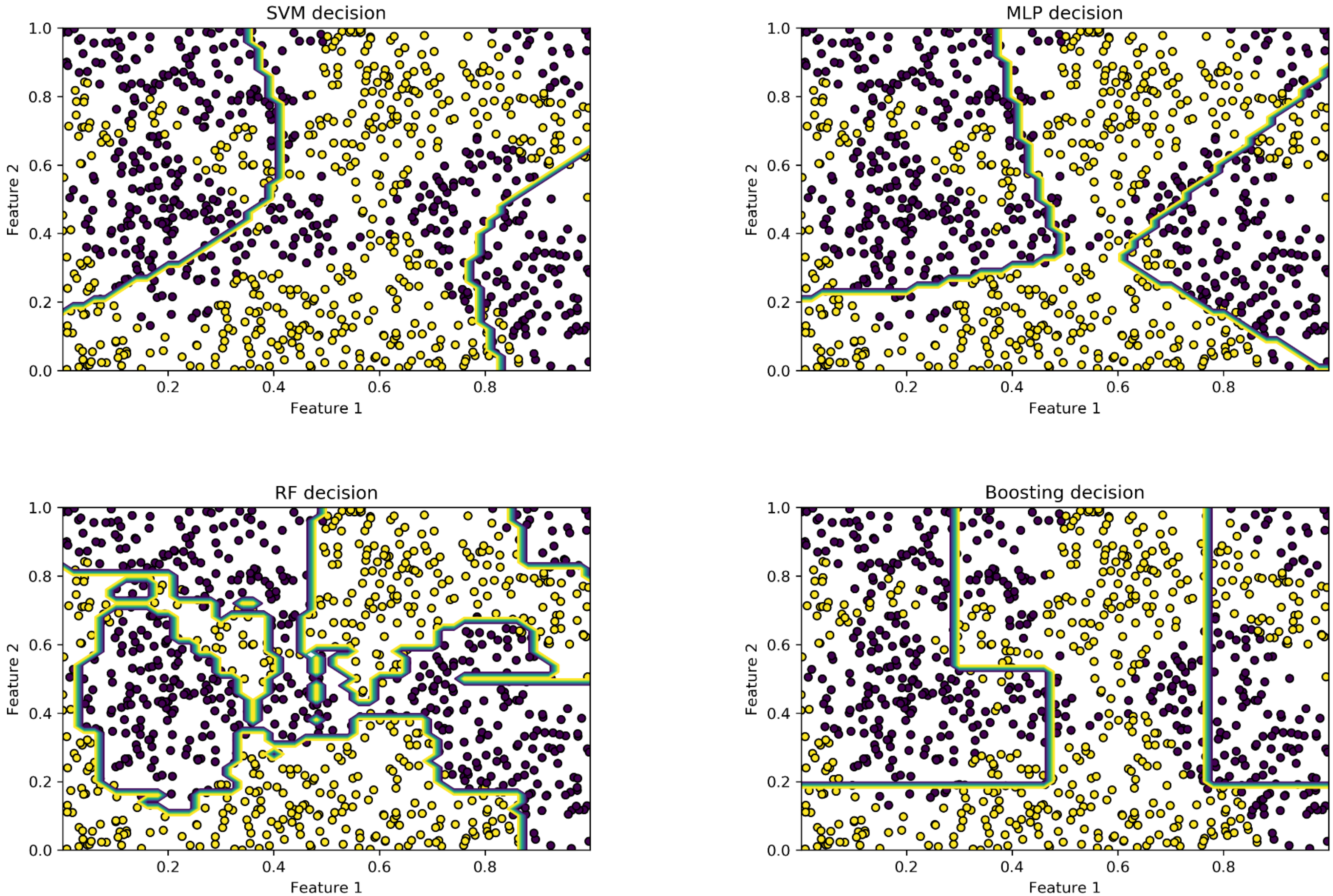


Case study: Dynamic Selection techniques



Case study

- Monolithic classifiers as well as static ensemble methods



Performance classification methods

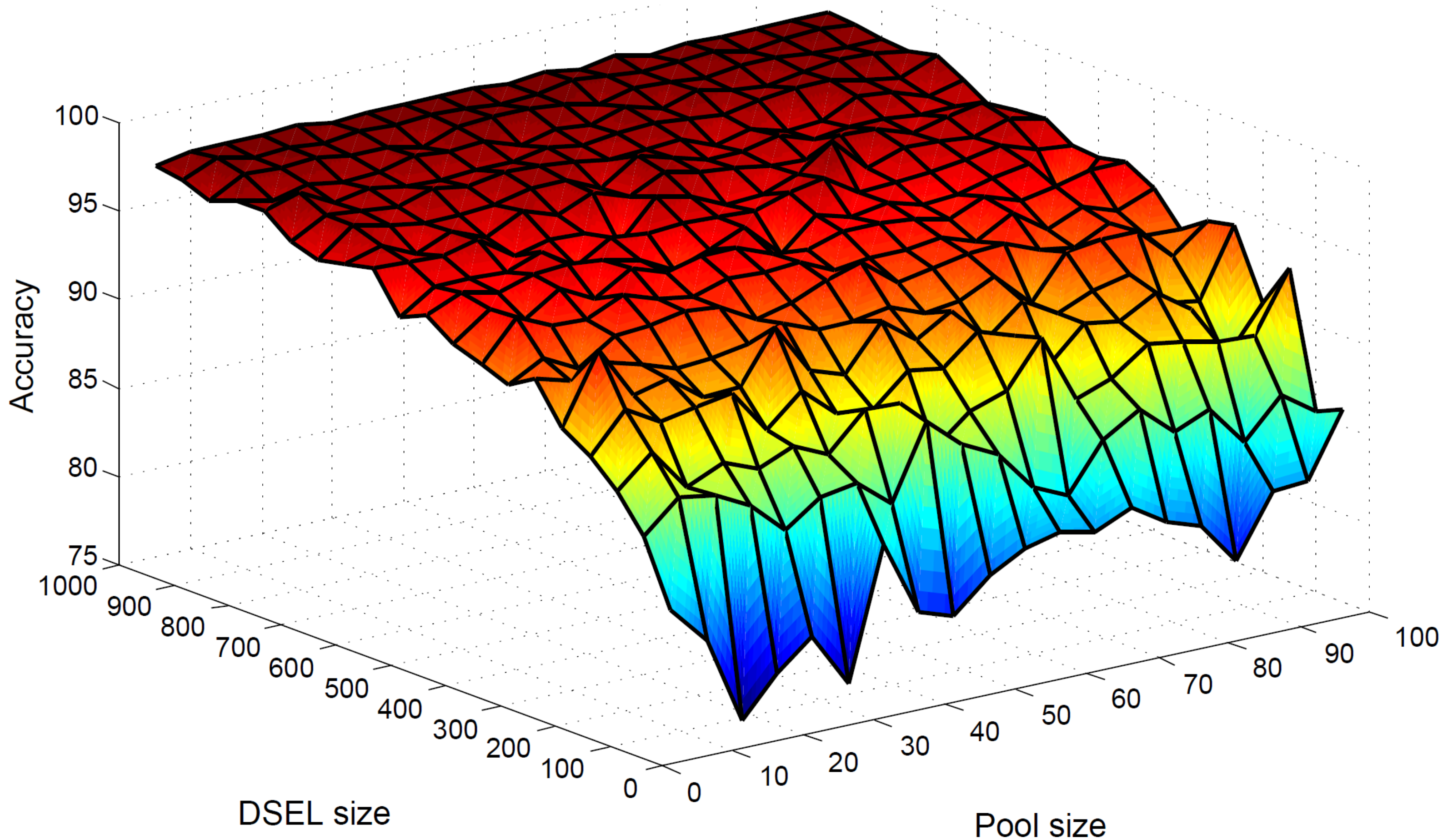
- Classification accuracy DS methods*
  - KNORA-Eliminate – 96.10%
  - DES-Performance – 94.20%
  - META-DES – 96.10%
  - OLA – 94.20%
  - Classifier Rank – 96.10%

- Classification accuracy single classifier and static combination*
  - SVM – 72.20%
  - MLP – 77.76%
  - Random Forest – 92.80%
  - AdaBoost – 70.10%

*using the default hyper-parameters configuration from scikit-learn and DESlib

Case study: Pool size vs DSEL size

- The size of the dynamic selection dataset (DSEL) has more impact in the classification accuracy



META-DES.Oracle: Differences

- 15 sets of meta-features
  - Ranking, probabilistic models, ambiguity, information theory, etc...

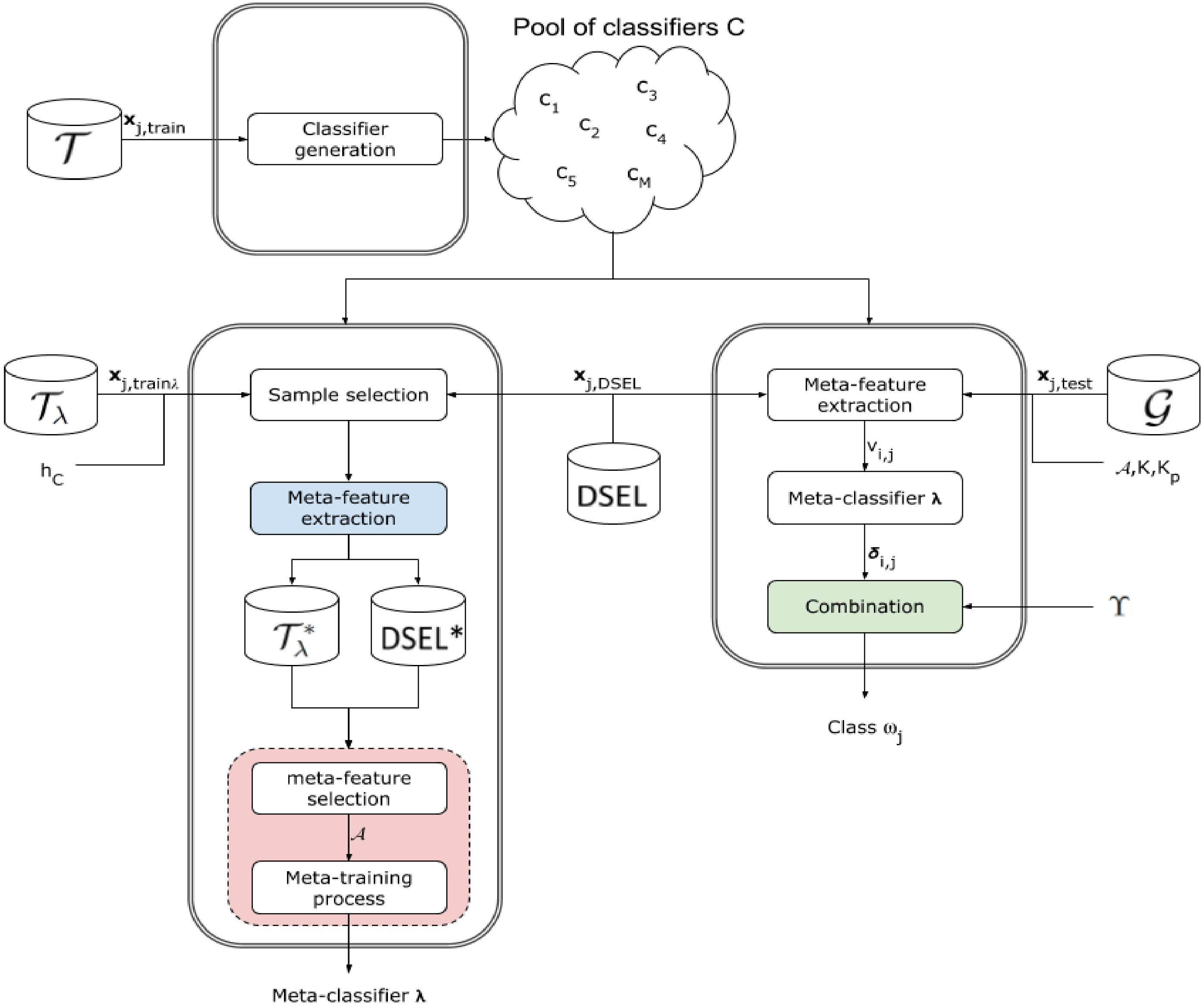
- An optimization scheme for the meta-classifier based on the formal definition of the Oracle

- Meta-features selection scheme based on Binary Particle Swarm Optimization (BPSO)

- **15 sets of meta-features** are used to estimate the competence of base classifiers
  - Different sources of information such as Ranking, Ambiguity and Probabilistic functions are considered

Meta-Feature	Criterion	Domain	Object	No. of Features
f_{Hard}*	Classification of the K-Nearest Neighbors	Accuracy	θ_j	K
f_{Prob}*	Posterior probability obtained for the K-Nearest Neighbors	Probabilistic	θ_j	K
$f_{Overall}$*	Overall accuracy in the region of competence	Accuracy	θ_j	1
f_{OP}*	Output profiles classification	Behavior	ϕ_j	K_p
f_{Conf}*	Degree of confidence for the input sample	Confidence	$\mathbf{x}_j$	1
f_{Cond}	Conditional accuracy in the region of competence	Accuracy	θ_j	1
f_{Amb}	Ambiguity in the vector of class supports	Ambiguity	$\mathbf{x}_j$	1
f_{Log}	Logarithmic difference between the class supports	Probabilistic	$S(\mathbf{x}_j)$	K
f_{PRC}	Probability of Random Classifier	Probabilistic	$S(\mathbf{x}_j)$	K
f_{MD}	Minimum difference between the predictions	Probabilistic	$S(\mathbf{x}_j)$	K
f_{Ent}	Entropy in the vector of class supports	Probabilistic	$S(\mathbf{x}_j)$	K
f_{Exp}	Exponential difference between the class supports	Probabilistic	$S(\mathbf{x}_j)$	K
f_{KL}	Kullback-Leibler divergence	Probabilistic	$S(\mathbf{x}_j)$	K
f_{Rank}	Classifier ranking in the feature space	Ranking	DSEL	1
$f_{Rank_{OP}}$	Classifier ranking in the decision space	Ranking	ϕ_j	1

META-DES.Oracle



Experiments: Datasets

- 30 Datasets from different repositories (UCI, KEEL …)

Database	No. of Instances	Dimensionality	No. of Classes	Source
Adult	48842	14	2	UCI
Banana	1000	2	2	PRTOOLS
Blood transfusion	748	4	2	UCI
Breast (WDBC)	568	30	2	UCI
Cardiotocography (CTG)	2126	21	3	UCI
Ecoli	336	7	8	UCI
Steel Plate Faults	1941	27	7	UCI
Glass	214	9	6	UCI
German credit	1000	20	2	STATLOG
Haberman's Survival	306	3	2	UCI
Heart	270	13	2	STATLOG
ILPD	583	10	2	UCI
Ionosphere	315	34	2	UCI
Laryngeal1	213	16	2	LKC
Laryngeal3	353	16	3	LKC
Lithuanian	1000	2	2	PRTOOLS
Liver Disorders	345	6	2	UCI
MAGIC Gamma Telescope	19020	10	2	KEEL
Mammographic	961	5	2	KEEL
Monk2	4322	6	2	KEEL
Phoneme	5404	6	2	ELENA
Pima	768	8	2	UCI
Satimage	6435	19	7	STATLOG
Sonar	208	60	2	UCI
Thyroid	215	5	3	LKC
Vehicle	846	18	4	STATLOG
Vertebral Column	310	6	2	UCI
WDG V1	5000	21	3	UCI
Weaning	302	17	2	LKC
Wine	178	13	3	UCI

Experimental setup

- Size of the Pool: 100 base classifiers (Perceptrons)*
- 20 replications:
  - 50% Training, 25% DSEL and 25% Test
  - Prior probabilities for each class were maintaned
- Neighborhood size: K = 7
- For the DES-KMeans and DES-KNN the values of N and J were equally set at 30% of the pool (Soares et al., 2006)
- Parameters Kp, Hc for the META-DES were set at 5 and 80% based on previous publications (Cruz, 2015)

* R. M. O. Cruz, R. Sabourin, G. D. C. Cavalcanti, T. I. Ren, *META-DES: A dynamic ensemble selection framework using meta-learning*, Pattern Recognition 48 (5) (2015) 1925-1935.

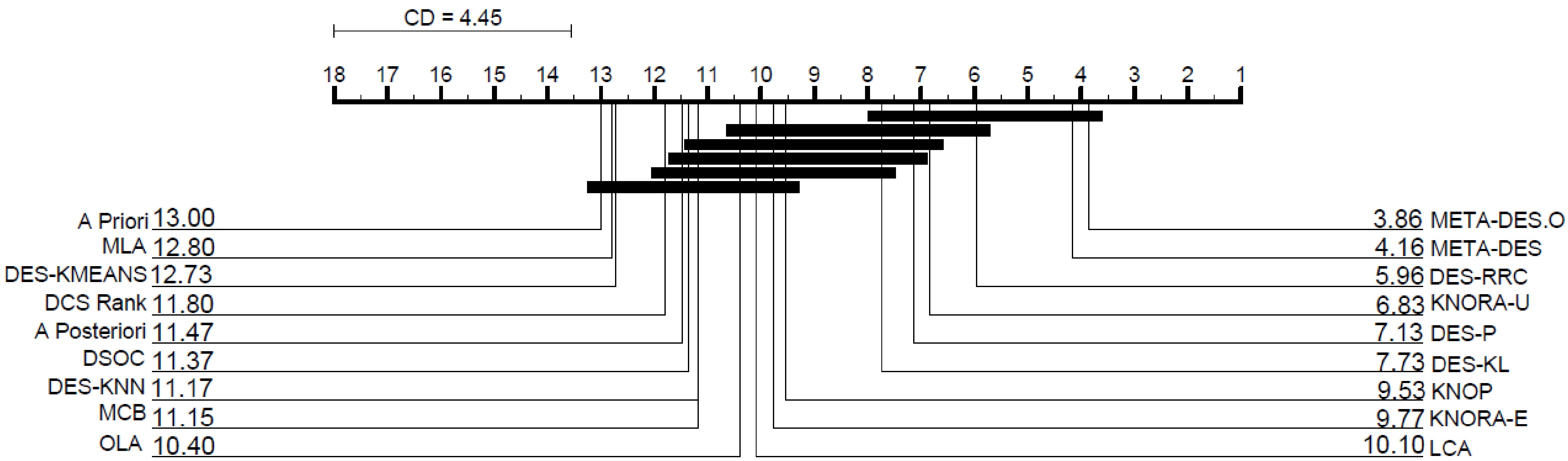
Experiments: All DS techniques

- Average rank and mean accuracy over 30 classification datasets

DS method	Avg. Rank	DS method	Mean Accuracy
META-DES.O	3.87(3.54)	META-DES.O	83.92(9.13)
META-DES	4.17(2.98)	META-DES	83.24(8.94)
DES-RRC	5.97(4.66)	DES-P	82.26(9.26)
KNORA-U	6.83(4.11)	DES-RRC	82.11(8.76)
DES-P	7.13(3.69)	KNORA-U	81.69(9.82)
DES-KL	7.73(4.92)	DES-KL	81.52(8.77)
KNOP	9.53(3.98)	KNOP	80.81(8.92)
KNORA-E	9.77(3.88)	KNORA-E	80.36(10.75)
LCA	10.10(4.66)	OLA	79.87(10.67)
OLA	10.40(4.95)	DCS Rank	79.69(10.38)
MCB	11.17(4.74)	LCA	79.57(9.84)
DES-KNN	11.17(4.40)	MCB	79.56(9.70)
DSOC	11.37(5.74)	DSOC	79.33(9.44)
A Posteriori	11.47(5.56)	DES-KNN	79.29(10.23)
DCS Rank	11.80(4.20)	A Priori	78.57(11.18)
DES-KMEANS	12.73(3.84)	DES-KMEANS	78.49(10.40)
MLA	12.80(4.60)	A Posteriori	78.14(11.53)
A Priori	13.00(4.53)	MLA	77.34(9.78)

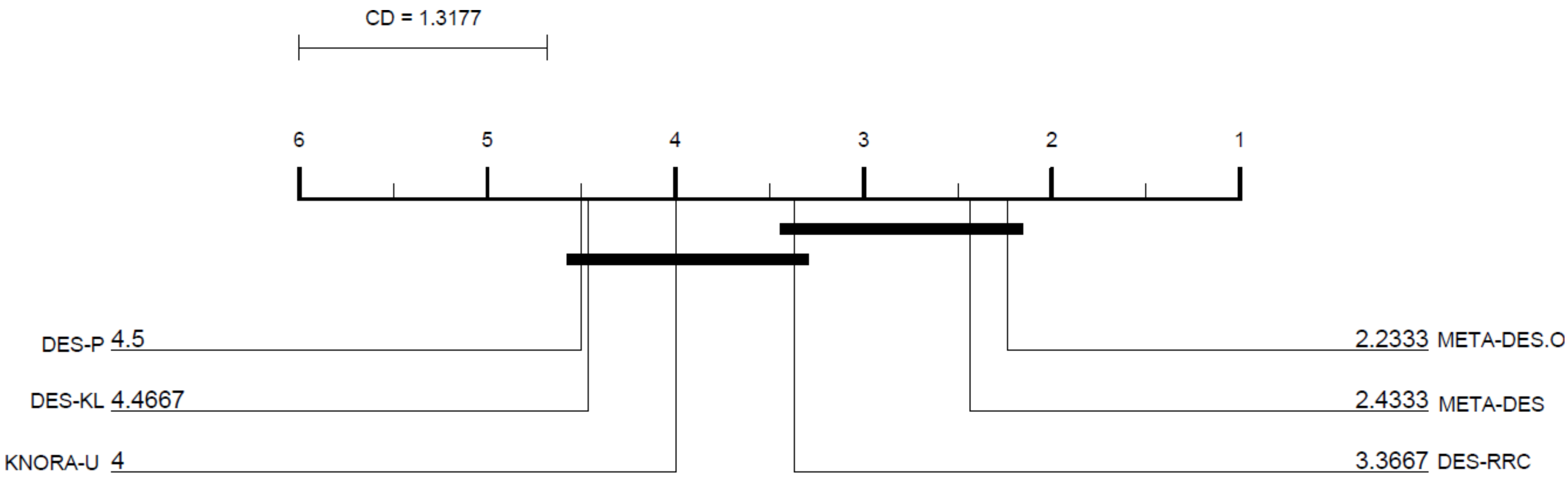
Experiments: All DS techniques

- Comparison based on Friedman rank with the Bonferonni-Dunn post-hoc test (Demsar, 2006)
  - Techniques that are statistically equivalent are connected with a black bar



Eperiments: Top 6 DS techniques

- Ranking analysis considering the top 6 methods



Comparison with other classification approaches

- Ensemble models
  - Single Best (SB)
  - Majority Voting (MV)
  - Static Selection (SS)
  - AdaBoost
  - Random Forest (RF)

- Single clasifiers
  - Multi-Layer Perceptron (MLP)
  - Gaussian Support Vector Machine (SVM)
  - K-Nearest Neighbor (K=1 and K=7)

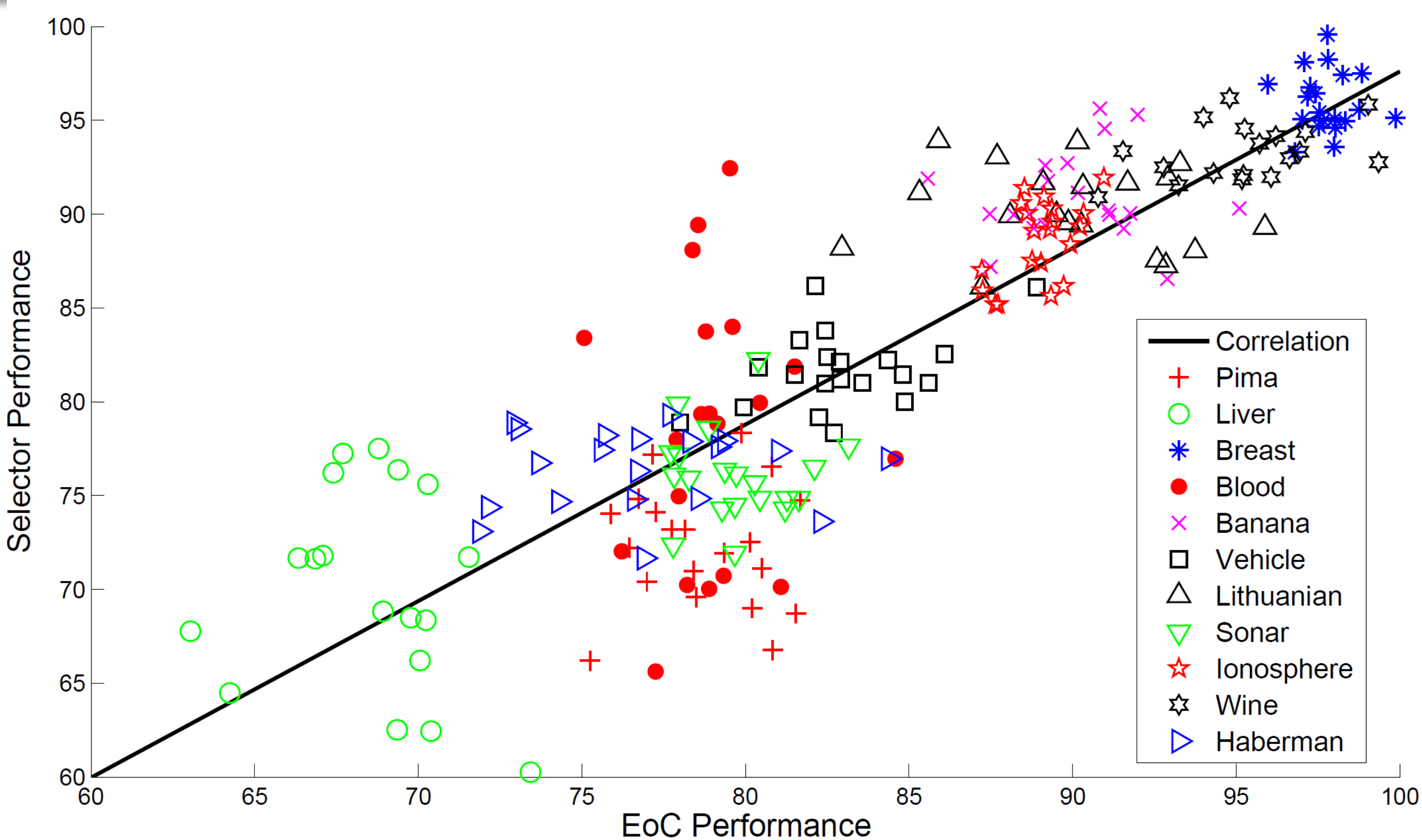
Comparison with other classification approaches

Algorithm	Avg. Rank	Algorithm	Accuracy
META-DES.O	5.43(4.92)	META-DES.O	83.92(9.13)
META-DES	5.70(4.28)	META-DES	83.24(8.94)
DES-RRC	7.67(6.23)	DES-P	82.26(9.26)
DES-P	9.17(5.27)	SVM	82.22(10.24)
KNORA-U	9.33(6.40)	DES-RRC	82.11(8.76)
DES-KL	9.90(6.42)	KNORA-U	81.69(9.82)
SVM	11.07(8.14)	DES-KL	81.52(8.77)
KNOP	13.07(5.86)	KNOP	80.81(8.92)
KNORA-E	13.23(5.62)	RF	80.78(10.98)
RF	13.77(9.50)	KNORA-E	80.36(10.75)
LCA	14.30(6.42)	OLA	79.87(10.67)
OLA	14.60(7.02)	DCS Rank	79.69(10.38)
MV	14.93(6.62)	LCA	79.57(9.84)
SS	14.97(6.38)	MCB	79.56(9.70)
MCB	15.03(7.49)	MV	79.51(9.39)
AdaBoost	15.43(7.63)	SS	79.40(10.12)
DES-KNN	15.53(6.49)	DES-KNN	79.29(10.23)
DCS Rank	16.33(5.77)	AdaBoost	79.23(10.32)
A Posteriori	16.40(7.91)	MLP	79.20(11.74)
SB	16.47(6.04)	SB	79.06(9.98)
DSOC	16.87(7.93)	DSOC	79.00(9.44)
MLP	16.90(8.45)	A Priori	78.57(11.18)
7-NN	17.40(8.59)	DES-KMEANS	78.49(10.40)
DES-KMEANS	17.50(6.13)	A Posteriori	78.14(11.53)
MLA	18.20(7.41)	7-NN	77.42(13.06)
A Priori	18.30(6.24)	MLA	77.34(9.78)
1-NN	20.50(8.10)	1-NN	76.64(11.98)

Lessons learned

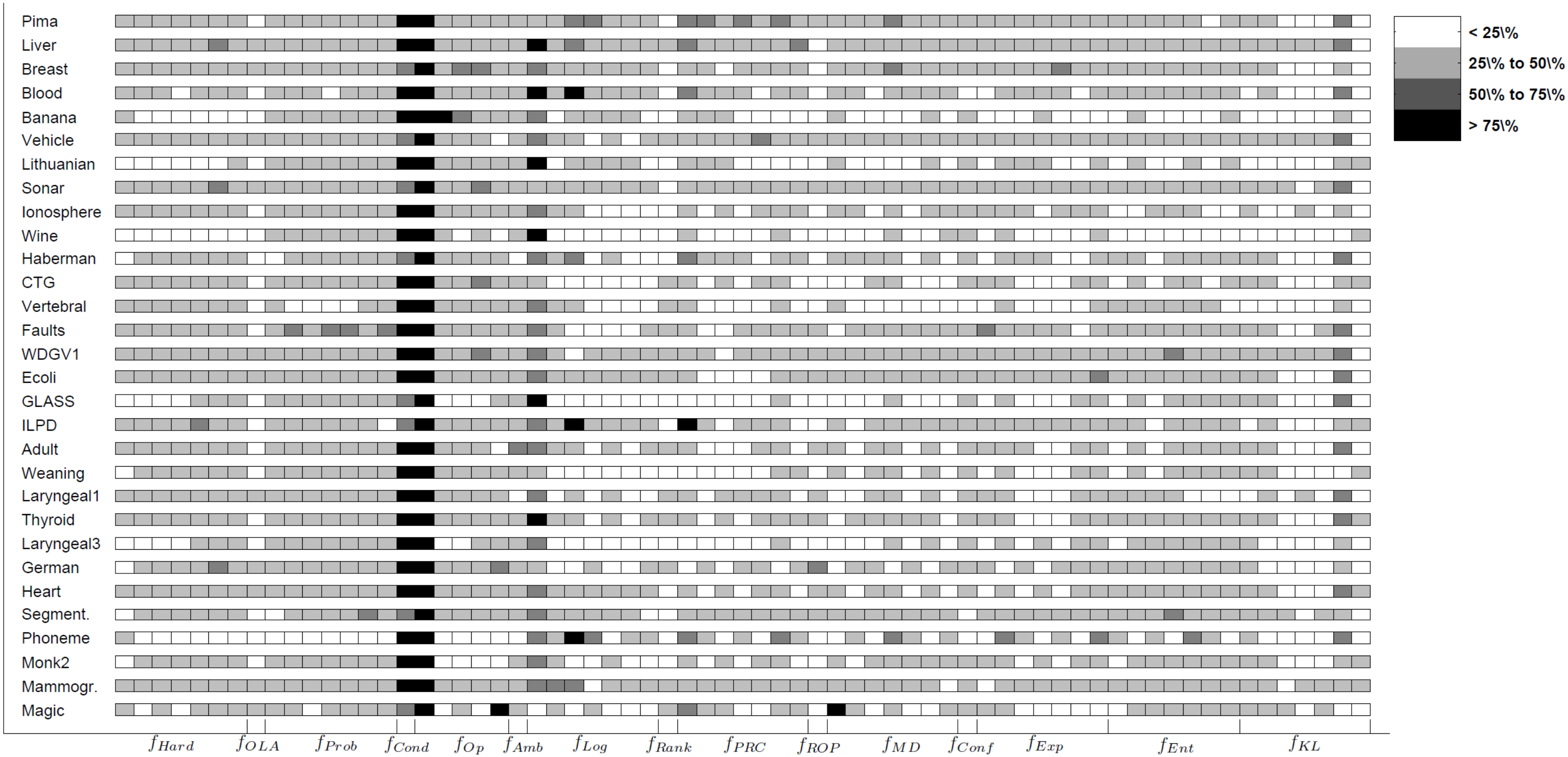
- DES outperforms DCS
  - 8 techniques in the top 10 are DES
- Methods based on local diversity (DS-KNN and DS-KMEANS) did not achieved good results
  - Promote consensus locally instead of diversity
- Majority of DS improves upon the baseline ensemble methods (Single Best, Static Selection and Majority Voting)
- Top 6 DS techniques presents comparable results with the best classifier models
  - SVM, Random Forests, NN, Adaboost...

Correlation meta-features



- strong correlation (= 0:88) between the meta-classifier and the META-DES performances.

Selected meta features per problem



- Selected meta-features changes according to classification problem

Part II New advancements

Outline Part II

1. DESlib: A dynamic ensemble selection library in Python
2. Dynamic selection vs KNN analysis
3. FIRE-DES framework
4. Dynamic selection for imbalanced datasets
5. Different contexts
6. Perspectives

DESlib: A Dynamic ensemble selection library in Python

- Open source project on GitHub: https://github.com/Menelau/DESlib
  - Contributions are welcomed!
- Implementation of key dynamic classifier and ensemble selection techniques as well as baseline methods
- Modular approach
  - Implementation follows the same taxonomy presented in this tutorial
  - Easy to implement new methods
- Project written in pure python and compatible with any platform

DESlib: A Dynamic ensemble selection library in Python

- Based on scikit-learn which is the main machine learning API in Python

- The library acccepts any classifier model from the scikit-learn library as the base classifier model
  - Any classifier model that implements the **predict**(X) and **predict_proba**(X) can be used

- Each DS method follow the same method signature of the scikit-learn API: **fit**(X, y), **predict**(X), **predict_proba**(X), **score**(X, y)
  - Making it a very easy to use API!

DESlib: A Dynamic ensemble selection library in Python

- A total of 13 DES and 7 DCS techniques already implemented
  - Variations of the methods also included
  - Can change DES methods into dynamic weighting by an easy parameter change.
- Implementation of baseline methods
  - Single Best, Oracle, Static Selection
- FIRE-DES framework available with the library
- Mechanism to select between using DS or KNN based on instance hardness

DESlib: A Dynamic ensemble selection library in Python

- Code Example:

```
from sklearn.ensemble import RandomForestClassifier
from deslib.des.knora_e import KNORAE


# Train a pool of 10 classifiers
pool_classifiers = RandomForestClassifier(n_estimators=10)
pool_classifiers.fit(X_train, y_train)

# Initialize the DES model
knorae = KNORAE(pool_classifiers)

# Preprocess the Dynamic Selection dataset (DSEL)
knorae.fit(X_dsel, y_dsel)

# Predict new examples:
knorae.predict(X_test)
```

- Jupyter Notebooks examples available on [GitHub]()

Dynamic selection VS K-NN (Cruz et al., IPTA 2017)

- The majority of DS techniques uses the K-NN algorithm to define a local region for the estimation of the competence level of the base classifiers

- Work conducted to understand why and when DS techniques outperforms the K-NN
  - even though its performance is heavily dependent on the K-NN for the definition of the regions of competence.

Dynamic selection VS K-NN (Cruz et al., IPTA 2017)

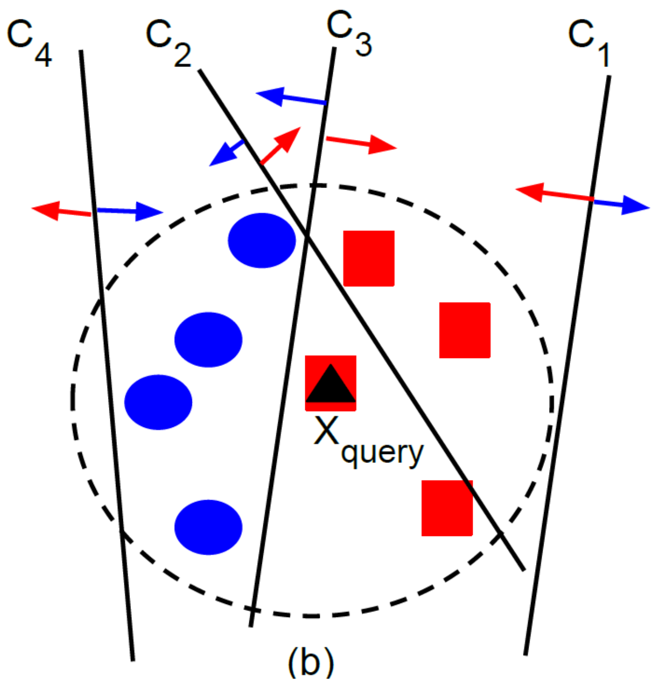
- The majority of DS techniques uses the K-NN algorithm to define a local region for the estimation of the competence level of the base classifiers

- Work conducted to understand why and when DS techniques outperforms the K-NN
  - even though its performance is heavily dependent on the K-NN for the definition of the regions of competence.

- **Our hypothesis is that DS can deal with instances that are located in indecision regions (i.e., close to the class borders).**

Recall the example

- Overall Local Accuracy (OLA)



- C1: 43% accuracy
- C2: 85% accuracy
- **C3: 100% accuracy**
- C4: 57% accuracy

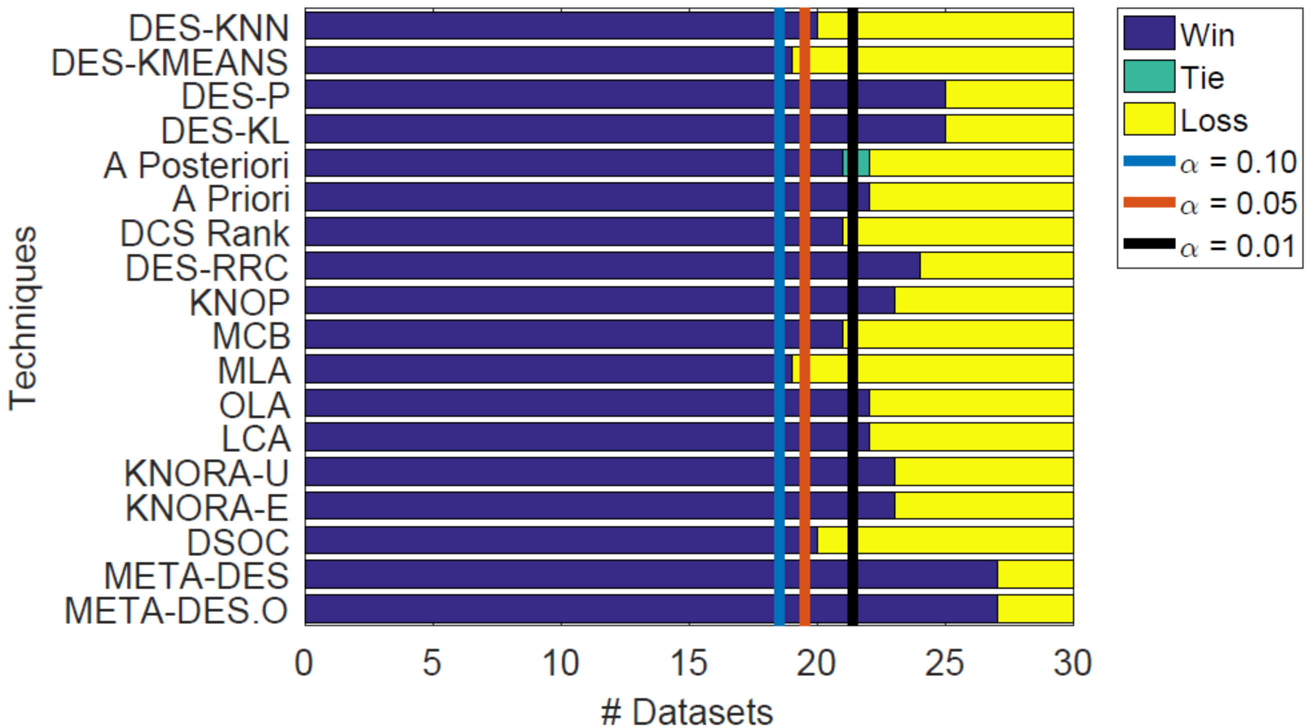
- **x_{query} is located in an hard region (overlap between classes)**
- 7NN would predict the wrong class, however DS can predict the correct class if it selects the local competent classifier

Research questions

- 1) Do DS techniques achieve higher classification performance than the K-NN?
- 2) Why does DS present better classification accuracy than K-NN even though the same neighborhood is considered for both techniques?
- 3) When should DS be used for classification instead of K-NN?

Comparison with the K-NN classifier (K = 7)

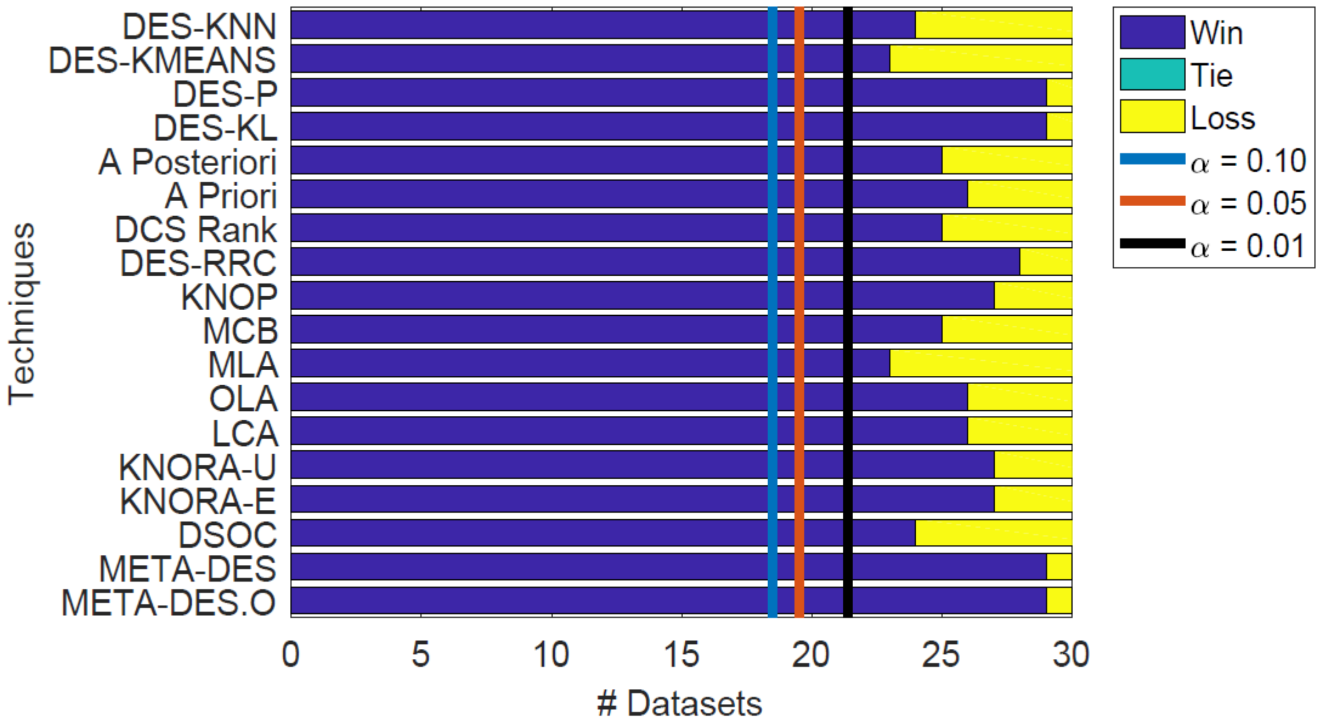
- All DS techniques present a significant number of wins for a α = 0.1,



- 16 and 12 DS techniques present a significant number of wins for an a α = 0.05 and 0.01 respectively

Comparison with the K-NN classifier (K = 1)

- All DS techniques obtained a significant number of wins for a α = 0.01,



Instance hardness (Smith et al., JMLR 2014)

- Instance hardness (IH) provides a framework for identifying which instances are hard to classify.

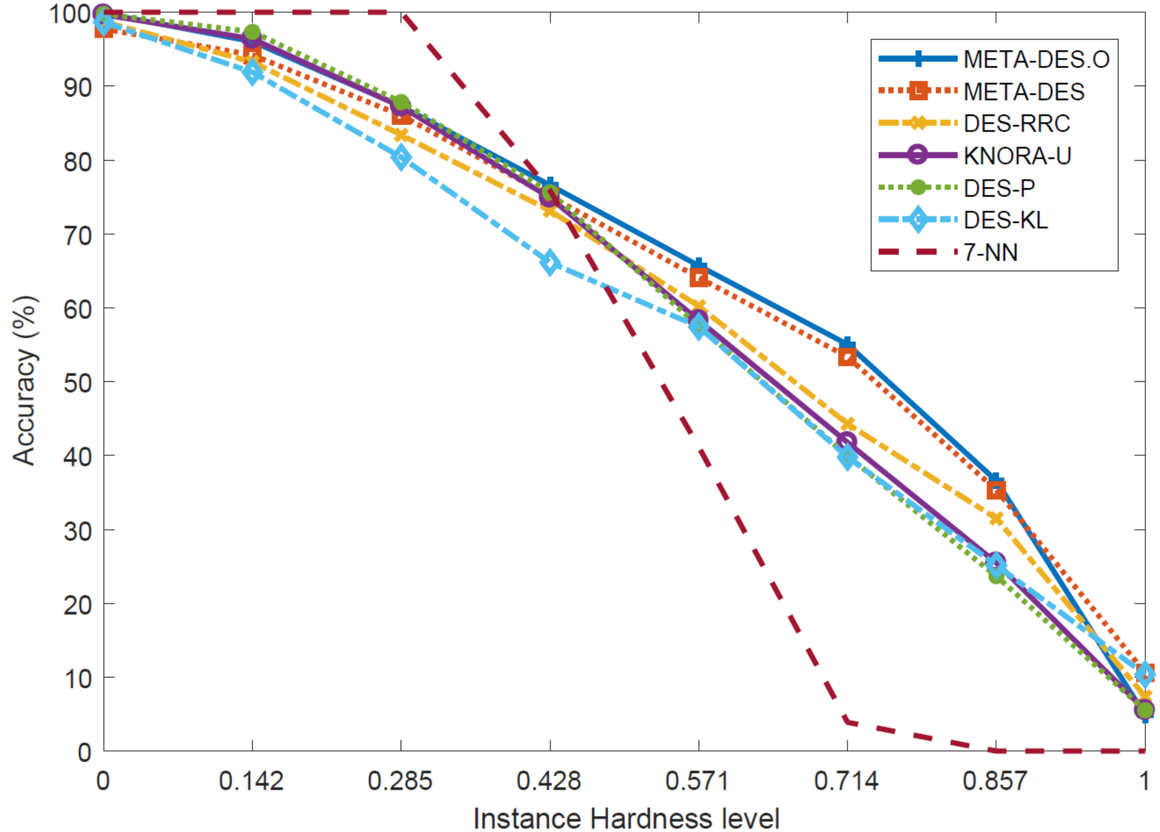
- The kDisagreeing Neighbors (kDN) (Equation 1) is considered
  - highest correlation with the probability that a given instance is misclassified by different classification methods.

$$kDN(\mathbf{x}_q) = \frac{|\ \mathbf{x}_k : \mathbf{x}_k \in KNN(\mathbf{x}_q) \wedge t(\mathbf{x}_k) \neq t(\mathbf{x}_q)\ |}{K}$$

- Samples with high kDN are close to the class borders

Instance Hardness analysis

- Top 6 DS techniques vs 7NN



- Classification accuracy of KNN significantly drop when the IH level increases

Dynamic Selection VS KNN: Lessons learned

- DS techniques indeed obtain a significant increase in classification peformance over the KNN
- The reason DS techniques achieves better performance is that they can correctly classify samples associated with high degree of instance hardness
  - Some conditions are necessary
- Use the IH level of an instance to determine whether to use the KNN or DS for classification
  - Can significantly reduce computational cost as only the “hard” samples will use the DS methods

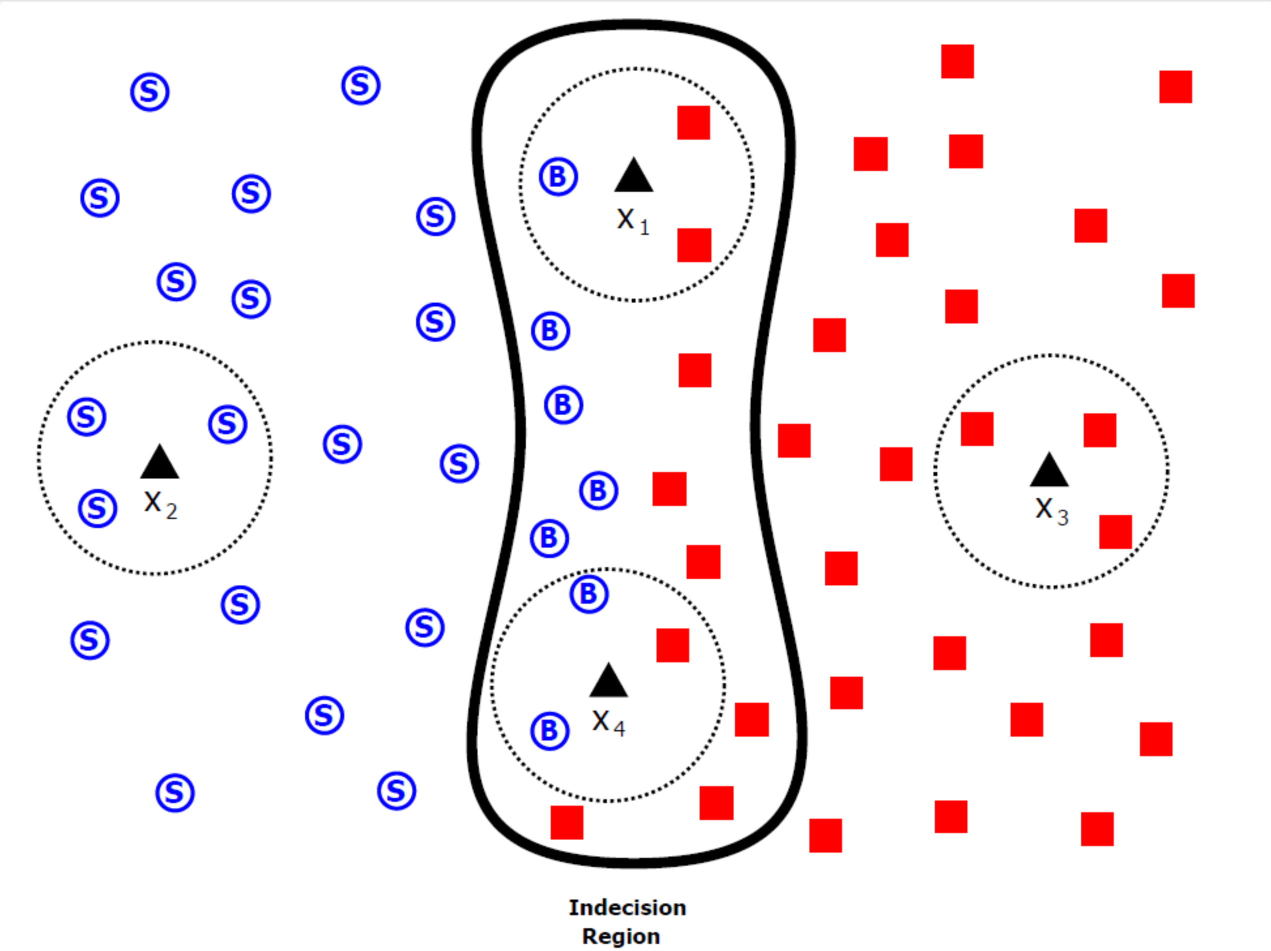
FIRE-DES (Oliveira et al., PR 2017)

- DS techniques do not take into consideration the existence of different scenarios when estimating the competence level of the base classifiers

- A test sample can be located in a region when almost all samples belong to the same class (safe region) or in a region where samples belongs to more than one class (indecision region)

- Moreover, the base classifier can have its decision boundary crossing or not crossing this local region

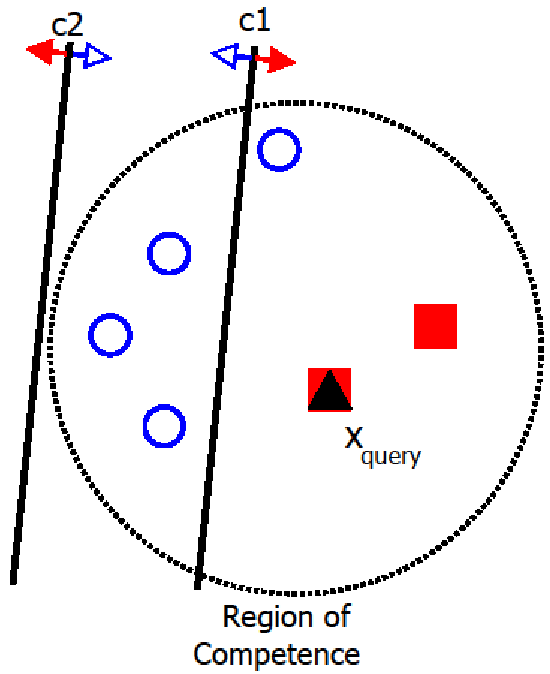
FIRE-DES: Indecision vs Safe region



- x_1 and x_4 are located in a indecision region
- x_2 and x_3 are located in a safe region

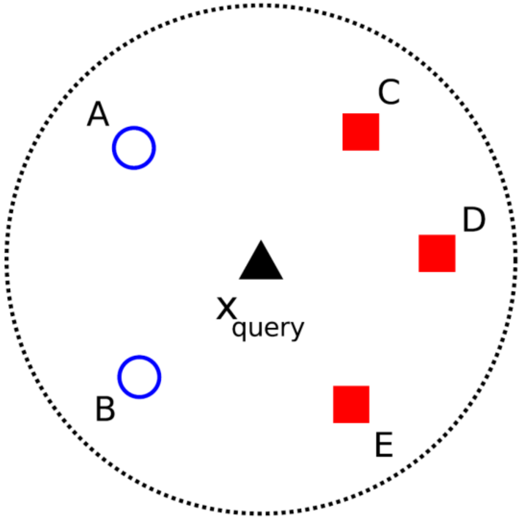
FIRE-DES

- c1 and c2 have the same local accuracy (80%)
- c1 crosses the region of competence and correctly classifies samples belonging to different classes, while c2 predicts all neighbors as belonging to the “blue circle” class



FIRE-DES

- Frienemies: two samples x_a and x_b are frienemies if:
    1. X_a and x_b are located in the region of competence;
    2. X_a and x_b have different classes



- *(A , C)*, (*A* , *D*), (*A* , *E*), (*B* , *C*), (*B* , *D*), (*B* , *E*) are the pairs of frienemies in the region of competence of the test sample

FIRE-DES

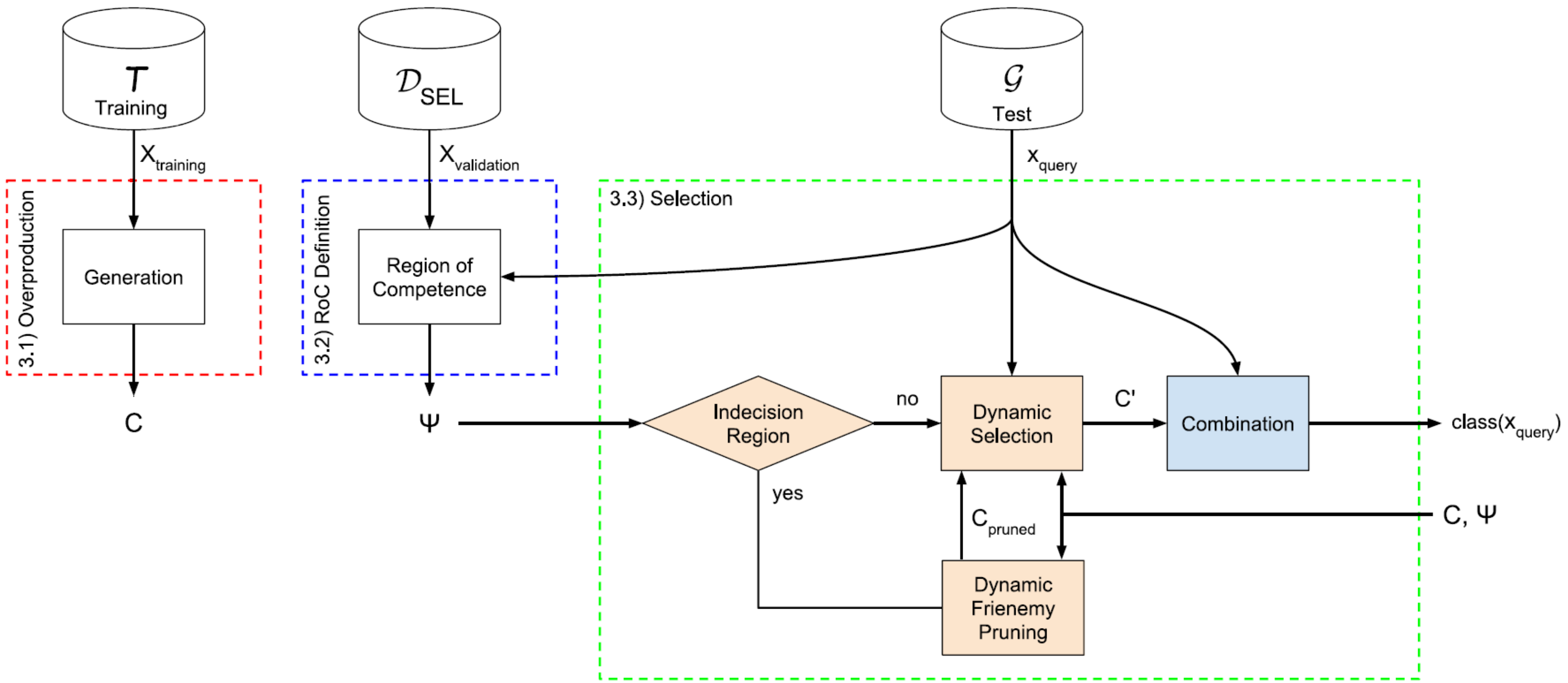
- Online pruning method applied before the dynamic selection stages

- Evaluate whether the query is located in a safe region or indecision region
  - Apply the online pruning mechanism for samples located in indecision regions

- Only consider base classifiers that correctly classify a pair of frienemies

- Can be applied as a previous step of any DS technique or static ensemble methods

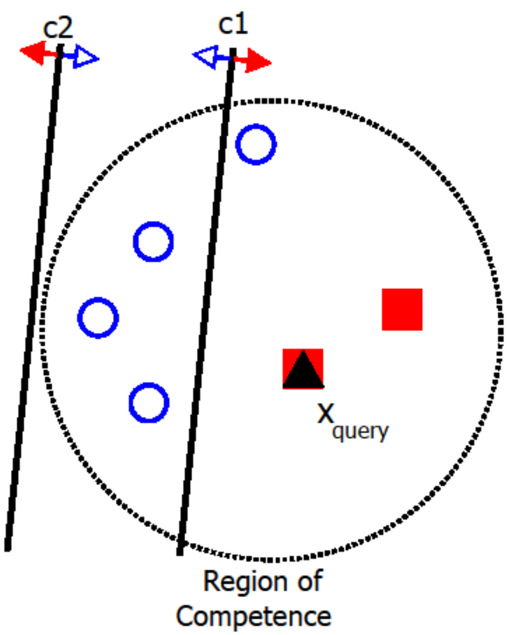
FIRE-DES framework



FIRE-DES

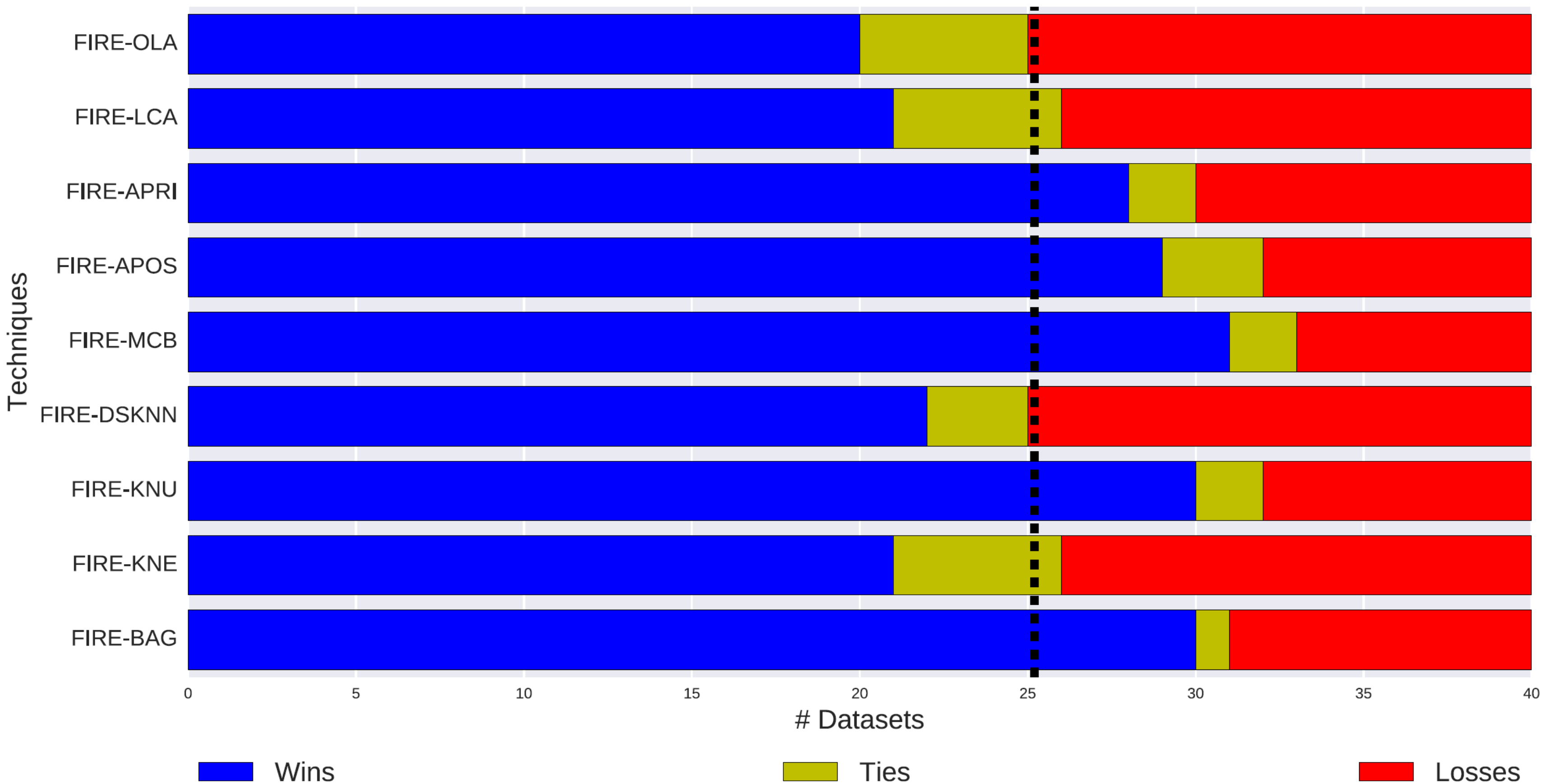
- c2 is removed from the pool since it does not correctly classify a single frienemy pair

- DS algorithm only take c1 into consideration



FIRE-DES

- Results on 40 imbalanced datasets (KEEL) show the proposed significantly improve the performance of several DS techniques



Dynamic selection for class imbalance

- Class-imbalance refers to classification problems in which many more instances are available for certain classes than for others.

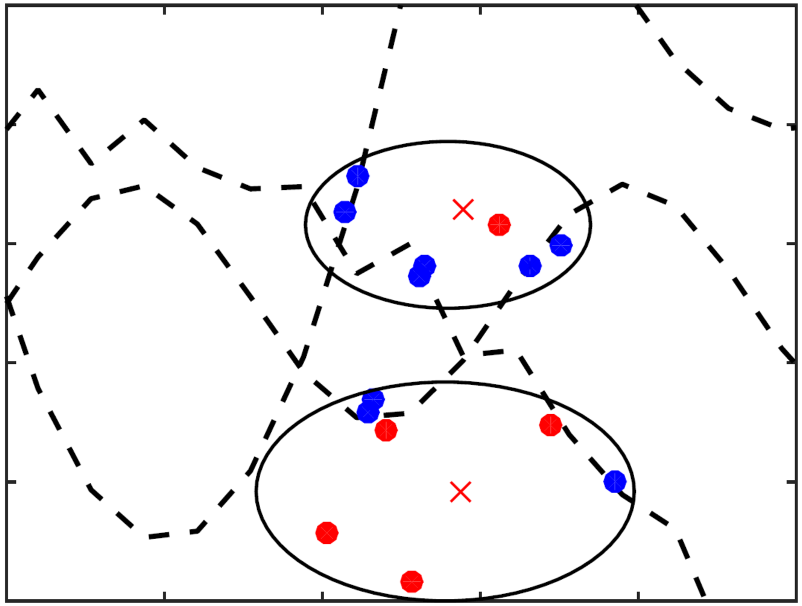
- Several real-world applications suffers from class imbalance
  - Image and text retrieval, fraudulent bank account transactions, medical diagnosis, activity recognition, sentiment analysis and so on

- Conventional classifiers typically favor the majority class
  - Fail to correctly classify the minority instances

Dynamic selection for class imbalance: Problem

- The performance of DS techniques is very dependent on the distribution of the dynamic selection dataset DSEL (Cruz et al., 2018)

- If this distribution becomes imbalanced, the estimation of competence can be biased towards the majority class



Dynamic selection for class imbalance: Problem

- The performance of DS techniques is very dependent on the distribution of the dynamic selection dataset DSEL (Cruz et al., 2018)

- If this distribution becomes imbalanced, the estimation of competence can be biased towards the majority class

Dynamic selection for class imbalance: Problem

- The performance of DS techniques is very dependent on the distribution of the dynamic selection dataset DSEL (Cruz et al., 2018)

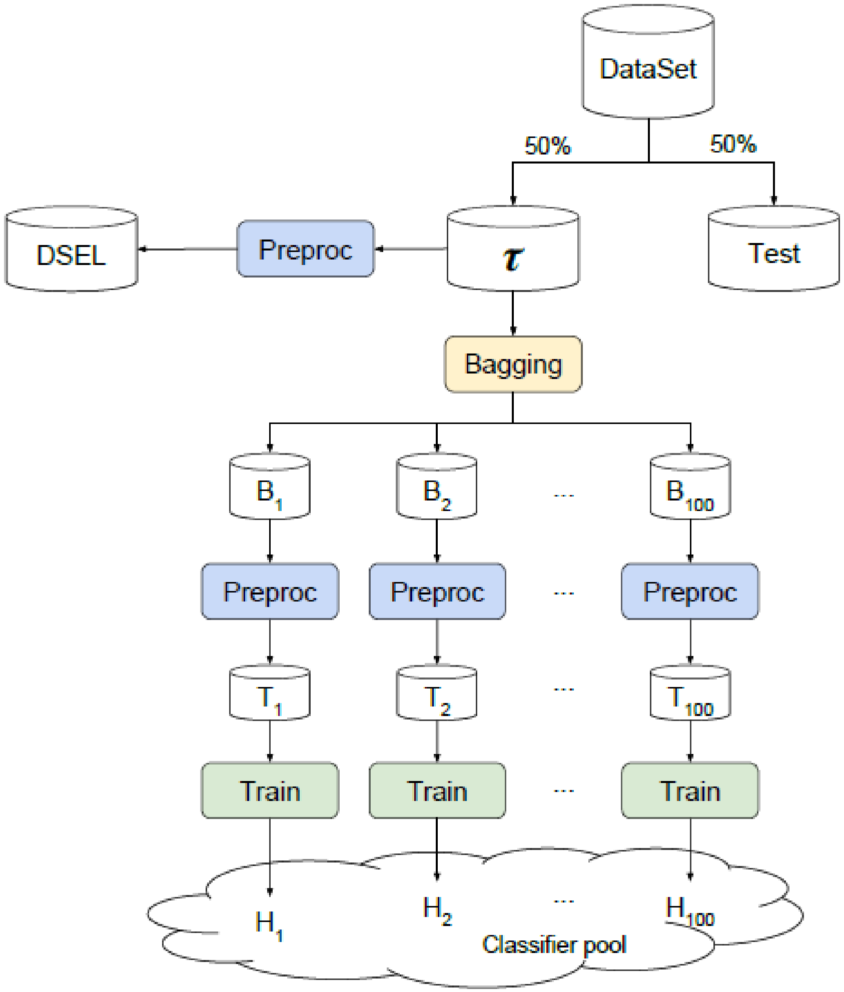
- If this distribution becomes imbalanced, the estimation of competence can be biased towards the majority class

- **Solution:** Apply data preprocessing techniques for balancing the distribution of DSEL

Dynamic selection for class imbalance: Data preprocessing

- Change the distribution of DSEL to compensate for the poor representativeness of the minority class
- Can be either undersampling, oversampling or hybrid
  - Under-sampling: remove instances from the majority class
  - Over-sampling: create artificial instances for the miniroty class
  - Hybrid: combine the two previous methods eliminating some examples before or after resampling
- Usually is the best approach for handling class imbalance with ensembles
  - Data preprocessing increases the diversity between classifiers

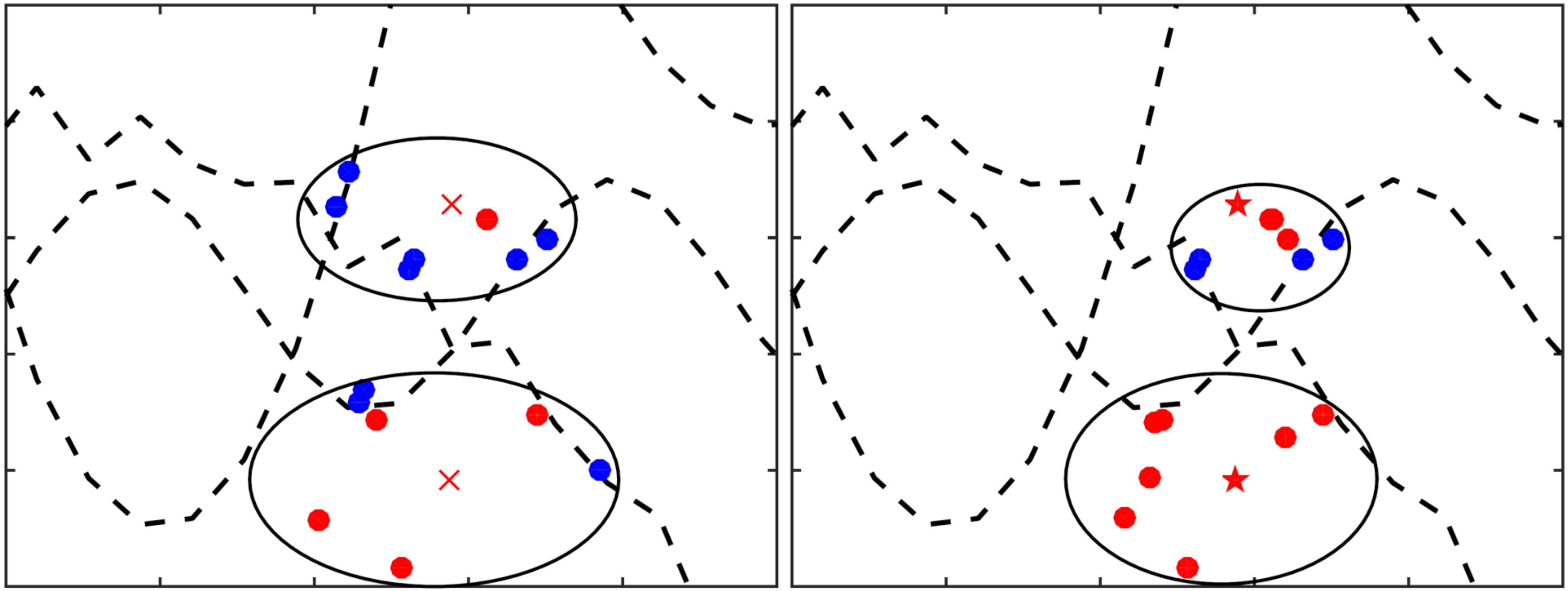
Dynamic selection for class imbalance: Data preprocessing framework (Roy et al., Neucom 2018)



- Augument the training dataset using data-preproessing to create the dynamic selection set (DSEL)
- Applies preprocessing method to each specific boostrap
- Diverse classifier pool due to the random nature of the preprocessing algorithm + Bagging

Dynamic selection for class imbalance: Data preprocessing

- Effect of data preprocessing on DSEL



(a) DSEL without preprocessing (b) DSEL with preprocessing

Dynamic selection for class imbalance: Data preprocessing

- Three data preprocessing techniques are considered:
  1. Synthetic Minority Over-sampling Technique (SMOTE):
  2. Ranked Minority Over-Sampling (RAMO):
  3. Random Balance (RB)

- Variations of these techniques also considered
  - Changing the number of samples generated

- Extension to multi-class scenario by applying over-sampling for each minority class

Experimental Setup

- 20 imbalanced datasets from the HDDT, 64 from the KEEL repository and 26 multi-class imbalanced datasets

- Pool composed of 100 decision trees (Weka J48)
  - Using Laplace smoothing at the leaves, but without pruning and collapsing also known as C4.4 (Diez-Pastor, Info Sciences 2015)

- Neighborhood size: K = 7

Experimental Setup

- Five data preprocessing methods applied together with Bagging

Bagging based methods		
Abbr.	Name	Description
Ba	Bagging	Bagging without preprocessing
Ba-RM100	Bagging+RAMO 100%	RAMO to double the minority class
Ba-RM	Bagging+RAMO	RAMO to make equal size for both classes
Ba-SM100	Bagging+SMOTE 100%	SMOTE to double the minority class
Ba-SM	Bagging+SMOTE	SMOTE to make equal size for both classes
Ba-RB	Bagging+RB	RB to randomly balance the two classes

- Two DCS and Two DES methods
  - Modified Rank, LCA, KNORA-Eliminate and KNORA-Union

Experimental results

AVERAGE RANKS FOR THE BEST ENSEMBLE METHODS. (A) ACCORDING TO AUC, (B) ACCORDING TO F-MEASURE AND (C) ACCORDING TO G-MEAN. RESULTS THAT ARE STATISTICALLY EQUIVALENT TO THE BEST ONE ARE IN BRACKETS.

(a) AUC		(b) F-measure		(c) G-mean	
Methods	Rank	Method	Rank	Method	Rank
Ba-SM+KNU	2.04	Ba-RM+KNU	2.15	Ba-SM+KNE	2.23
Ba-SM100+KNE	[2.42]	Ba-SM100+KNE	[2.31]	Ba-SM+KNU	[2.42]
Ba-SM	2.50	Ba-SM100	[2.46]	Ba-SM	[2.62]
Ba-SM100+RANK	3.77	Ba-SM100+RANK	3.58	Ba-SM100+RANK	3.38
Ba-SM100+LCA	4.27	Ba-SM100+LCA	4.50	Ba-SM+LCA	4.35

- Preprocessing methods significantly improved the results of DS techniques
- **DES techniques (KNE and KNU) achieves better results than Static combination**

Dynamic selection for class imbalance: Another solution

- Another possible solution is to use the class imbalance information during the competence estimation
- New paper being presented on this conference!

3:10PM: K-Nearest Oracles Borderline Dynamic Classifier Ensemble Selection

Session 2k-1: Mixture models, ensemble learning, and other meta-learning or committee algorithms. Monday, July 9, 2:10PM-4:10PM, Room: Oceania 9

Perspectives: Prototype Selection for DS

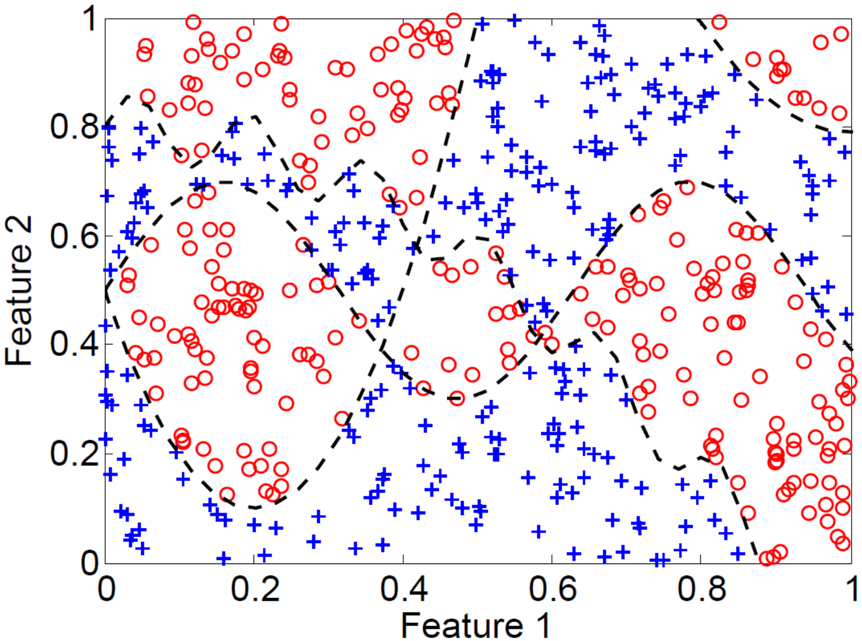
- The performance of DS techniques is very dependent on the distribution of DSEL (Cruz, NCAA 2018)

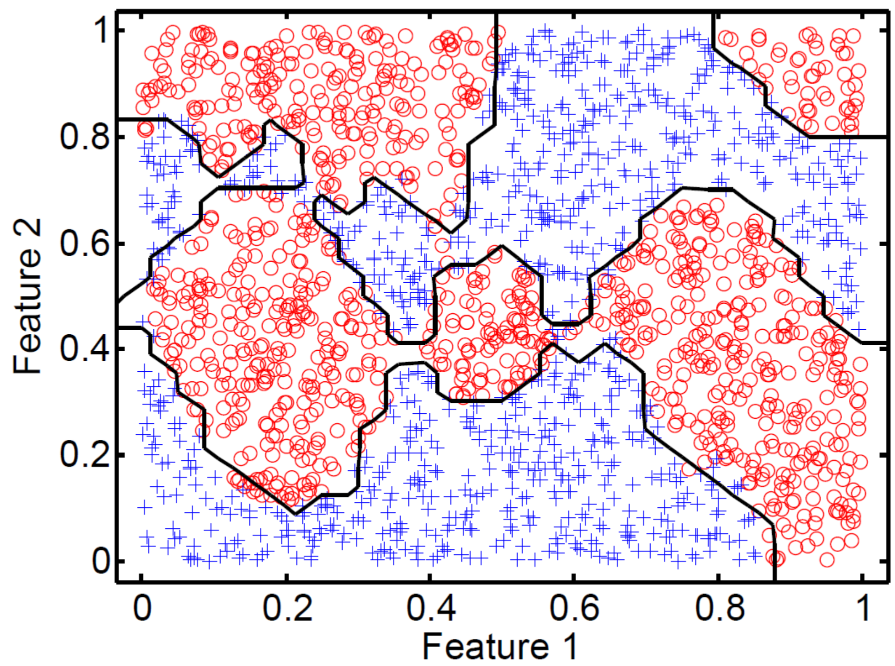
- The presence of noise in this set may hinder the dynamic selection scheme

- Definition of the region of competence can be very is expensive
  - Cost of computing the distance between $\mathbf{x}_j$ and each sample in DSEL
  - Reduction of redundant samples may significantly reduce the cost involved in DS systems

Perspectives: Prototype Selection for DS

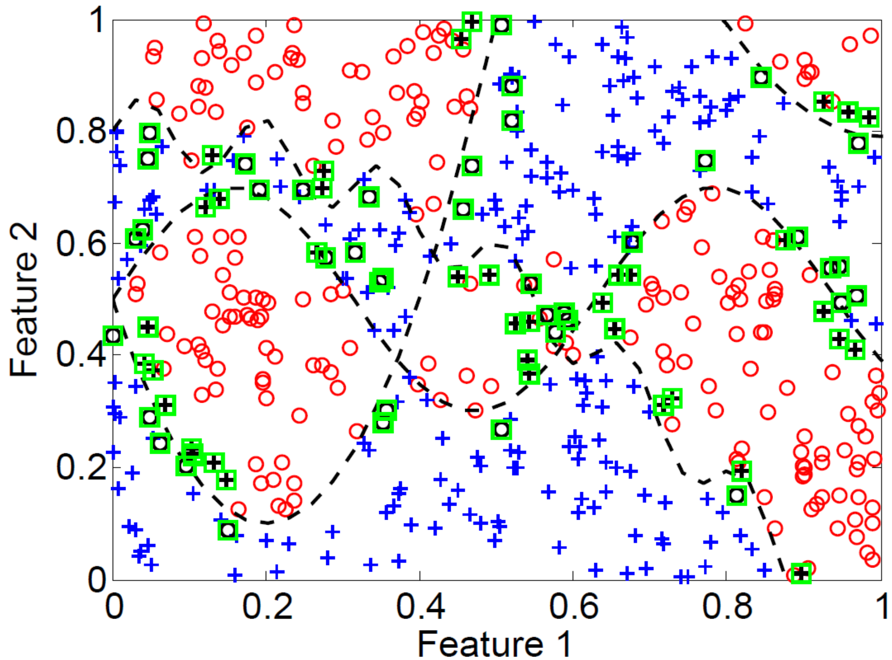
- Without any noise in DSEL, DS can easily approximate the complex decision boundary of the P2 problem using only linear classifiers (Cruz et al., NCAA 2018)

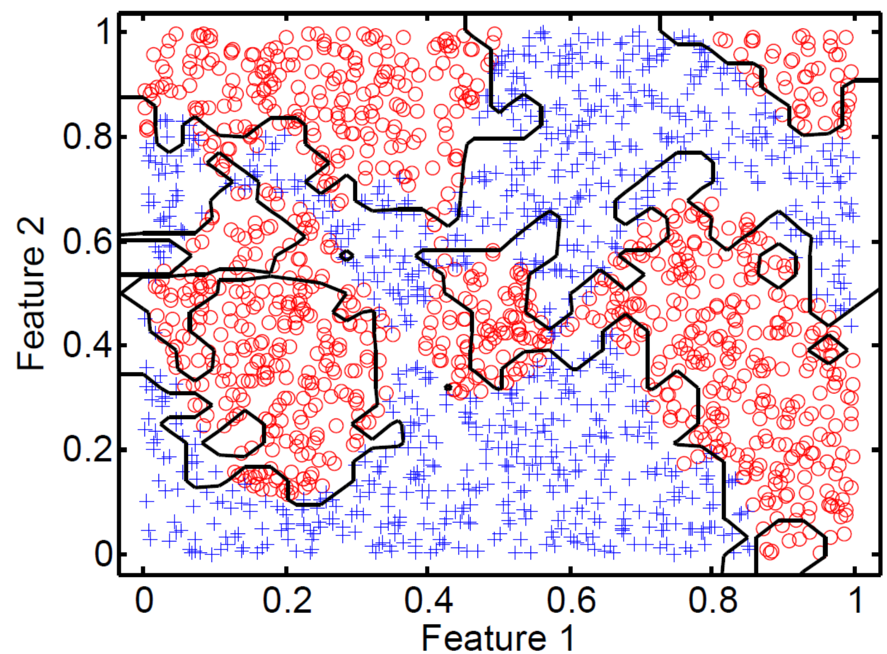




Perspectives: Prototype Selection for DS

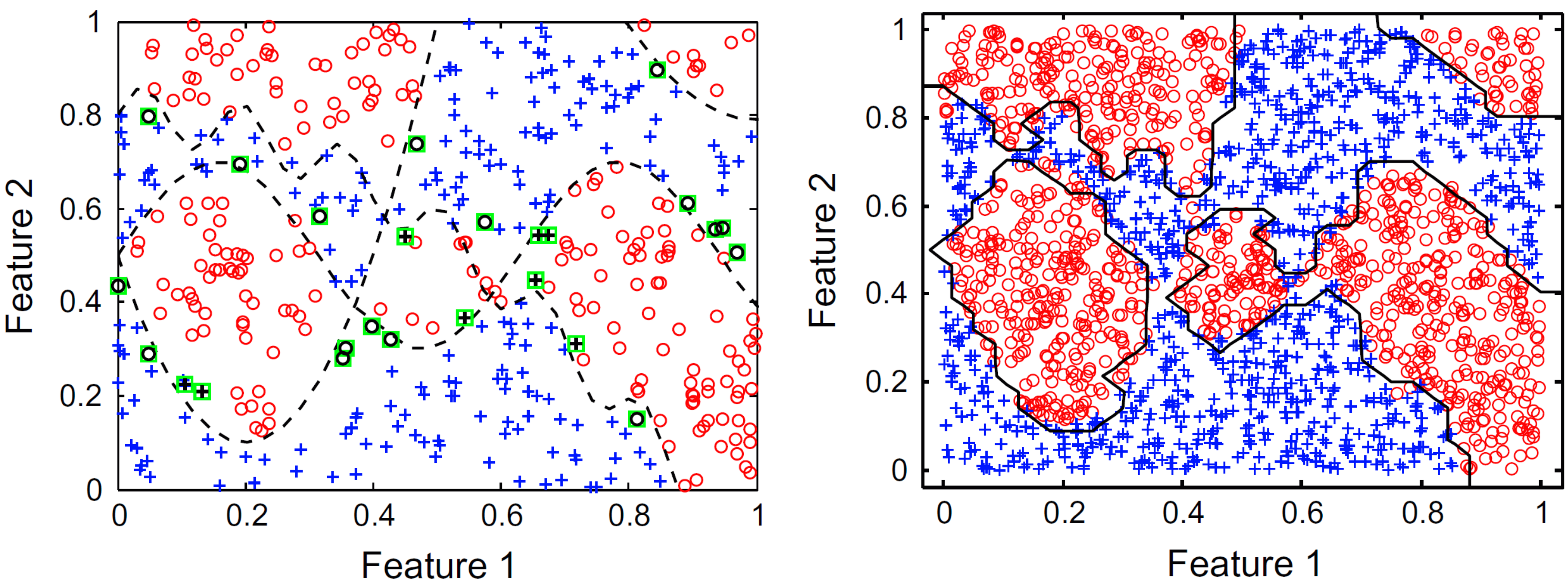
- The impact of noise in the decision of a DS algorithm. We can see the errors occurs in regions of the feature space where the presence of noise is more evident (Cruz et al., NCAA 2018)





Perspectives: Prototype Selection for DS

- After applying prototype selection
  - Edited Nearest Neighbor (ENN)



Perspectives: prototype generation for DS

- Study considering 6 PS techniques show that editing DSEL can significantly improve the performance of DS methods (Cruz et al., IJCNN 2017)

- PS techinques based on the performance of the K-NN reduced the performance of DS techniques
  - **Possible solution:** Include the performance of DS techniques rather than the K-NN in the fitness function of such methods

- The use of Prototype Generation (PG) techinques must be investigated
  - Such techniques also generate synthethic samples to fill the feature space

Perspectives: pool generation for DS

- Bagging, Boosting, Random Subspaces etc were proposed for static ensemble combination
  - Without taking into account local information
- Plenty of redundancy in the generation scheme
  - Usually less than a 100 base classifier is required for the optimal performance (Roy et al., ICPR2016)
- Pool generation method adapted to dynamic selection
  - Cover the whole feature space with local experts
  - Use the definition of the DS method in the pool generation scheme

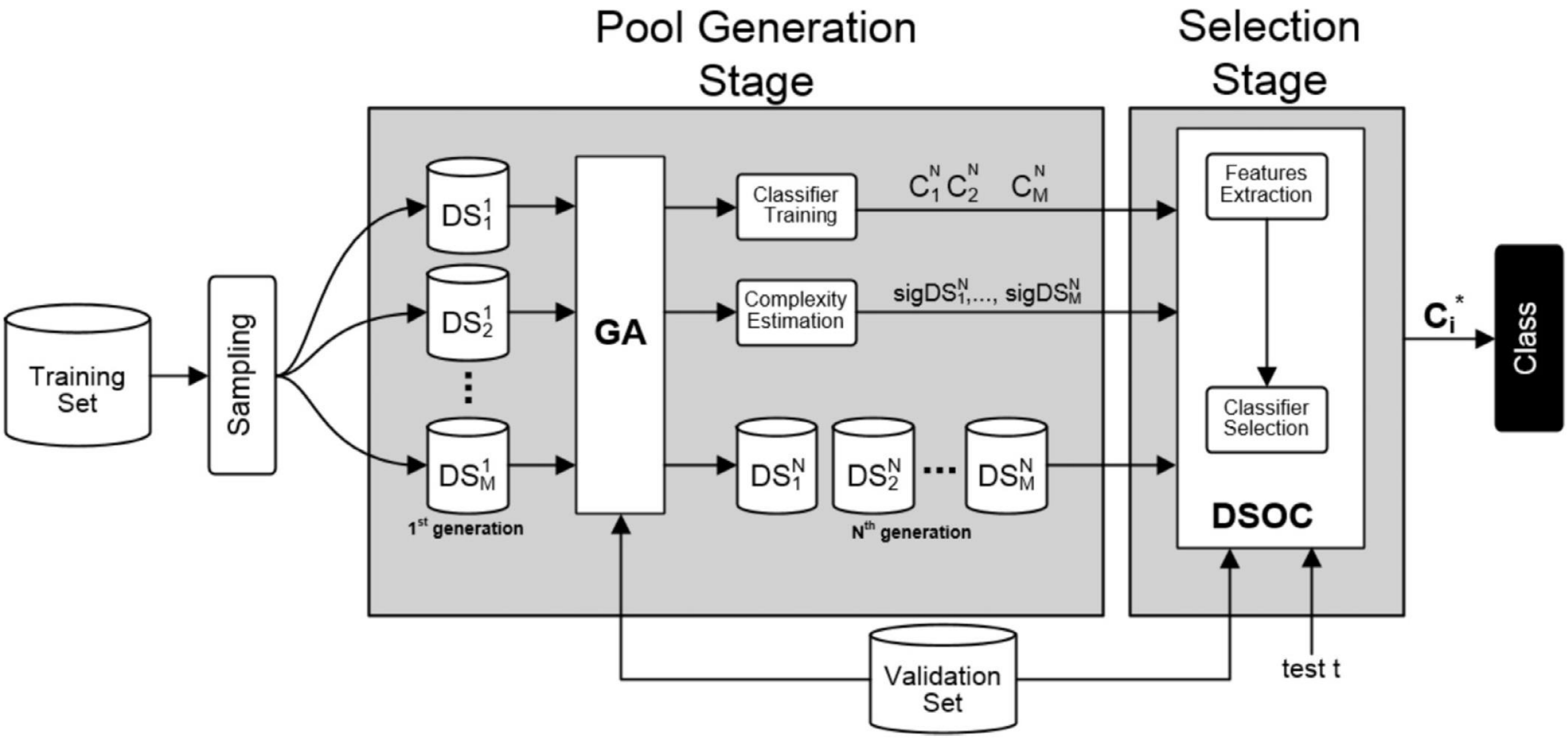
Dynamic Selection Over Complexity (DSOC)

- Take into account data complexity measures (Ho & Basu, 2002) to create a pool of classifiers and perform dynamic selection

- A Genetic Algorithm (GA) is used in order to evolve an initial pool of classifiers, taking into account data complexity
  - Create training distributions that cover the complexity space

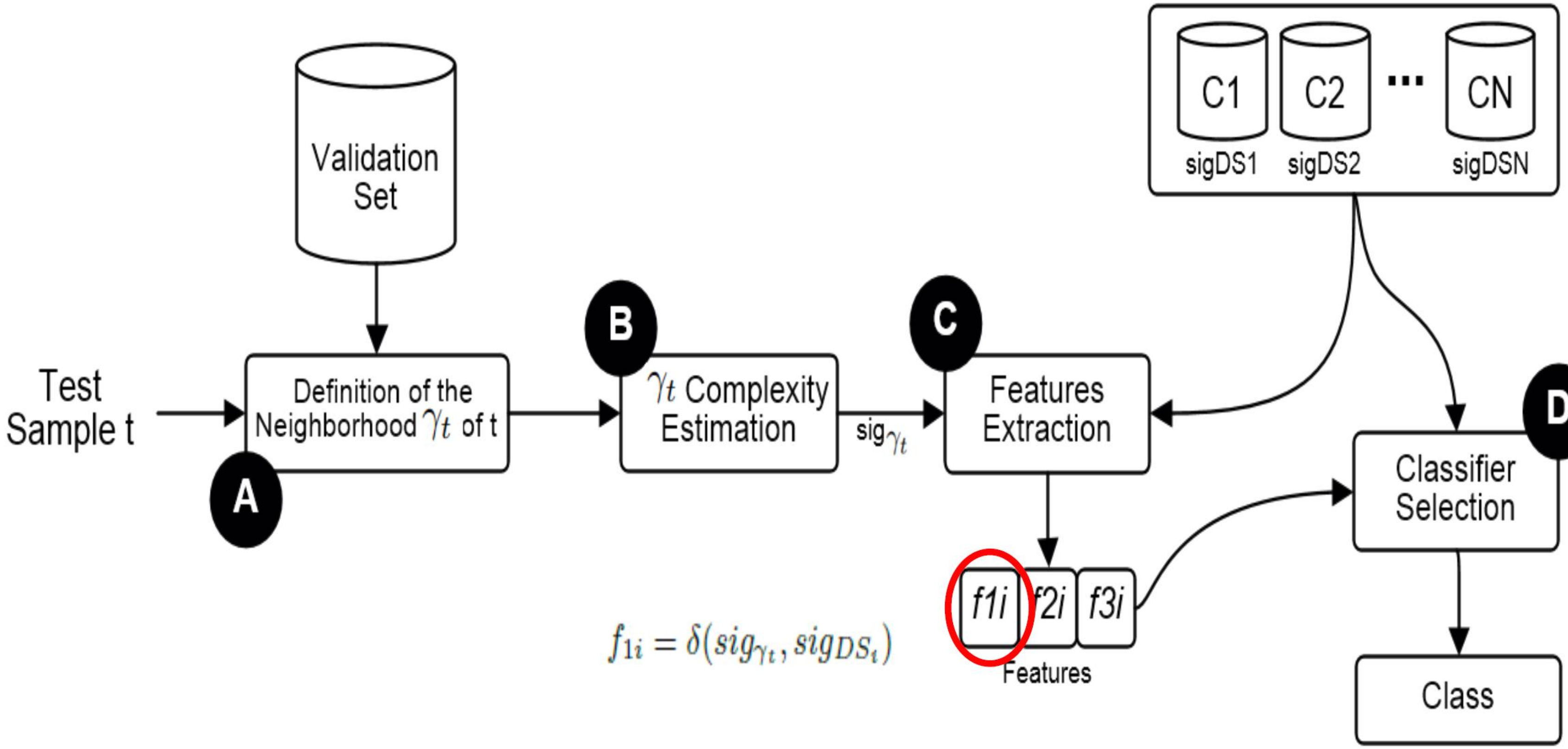
- The base classifier is selected based on three features
  - f1: Similarity in terms of complexity
  - f2: The distance of $\mathbf{x}_j$ and the centroid of the class prediced by c_i
  - f3:The local accuracy of C_i in the region of competence

Dynamic Selection Over Complexity (DSOC)



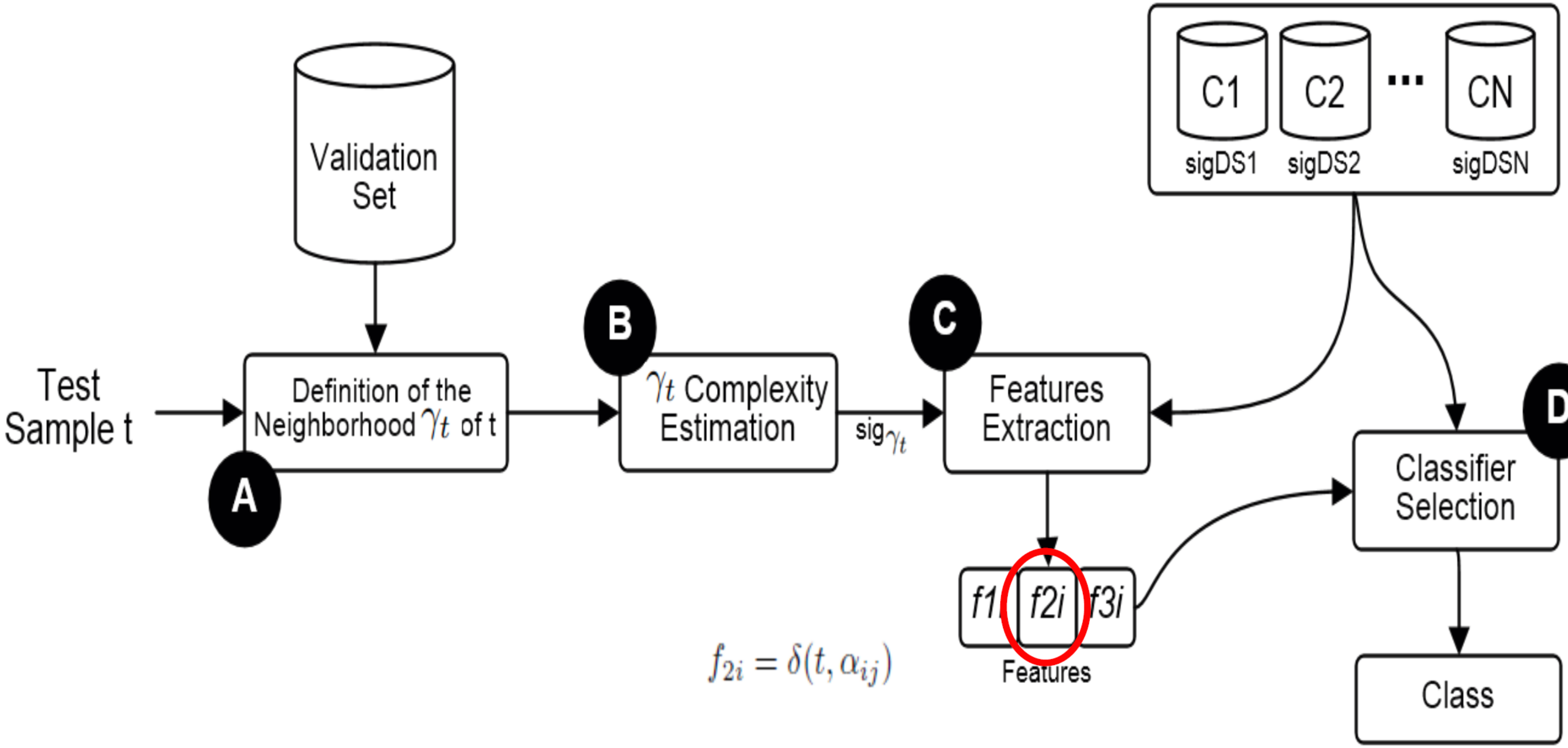
- Evolves the training distribution based on complexity

Dynamic Selection Over Complexity (DSOC)



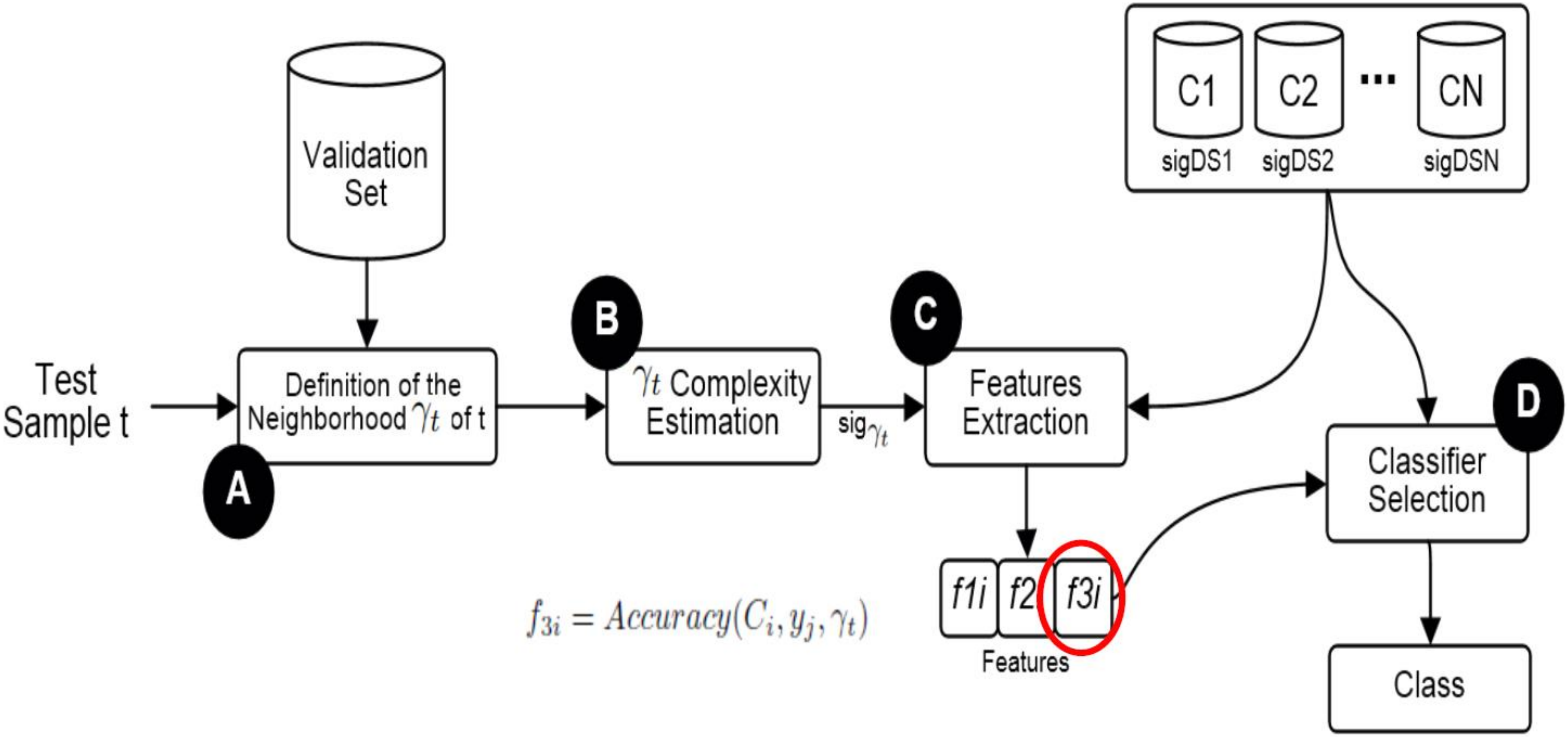
- f1 - Complexity similarity: calculates the similarity between the neighborhood of *t* and all classifiers complexity signatures using the euclidean distance.

Dynamic Selection Over Complexity (DSOC)



- f2 - Centroid distance: based on the class defined by each classifier for *t*, this metric represents the distance (in the feature space) of the test instance to the centroid of the class assigned by the classifier.

Dynamic Selection Over Complexity (DSOC)



- f3 - Local Accuracy: consists on the local class accuracy of each classifier estimated on the neighborhood of the test instance.

Dynamic Selection Over Complexity (DSOC)

- Where f1 and f2 are related to the concept of complexity
  - Can be seen as measures of pertinence of the base classifier
- f3 represents the local competence of the base classifier
- The final competence level of the base classifier is measure using the following formula:

$$\delta_{i,j} = (1 - f_{1i}) + (1 - f_{2i}) + f_{3i}$$

- The base classifier with a highest value of, $\delta_{i,j}$, is selected for the classification of $\mathbf{x}_j$

Perspectives: diversity vs consensus

- One of the most studied aspects in MCS
  - Classifiers that always produce the same decision will not improve the performance of the system

- Often used as a criteria to generate a pool of classifiers

- However, we still do not understand how it should be used to build ensembles
  - An ill-posed problem

- The diversity measure used also depends on the combination scheme employed

Perspectives: diversity vs consensus

- Promoting diversity at the ensemble level did not present good performance in our experiments
  - Promote consensus instead
- Promote diversity in a global level and consensus in the local level instead
  - Have a diverse pool of classifiers so it cover well the feature space
- Distant diversity (Saglam et al., 2016)
  - Selects the most competent classifiers together with the less diverse locally
- Analysis of the impact of diversity on dynamic selection is still an open question
  - At the pool level and at the ensemble level

References

[1] Cruz, Rafael MO, Robert Sabourin, and George DC Cavalcanti. "Dynamic classifier selection: Recent advances and perspectives." *Information Fusion* 41 (2018): 195-216.

[2] Cruz, Rafael MO, Robert Sabourin, George DC Cavalcanti, and Tsang Ing Ren. "META-DES: A dynamic ensemble selection framework using meta-learning." *Pattern recognition* 48, no. 5 (2015): 1925-1935.

[3] Roy, Anandarup, Rafael MO Cruz, Robert Sabourin, and George DC Cavalcanti. "A study on combining dynamic selection and data preprocessing for imbalance learning." *Neurocomputing* 286 (2018): 179-192.

[4] Cruz, Rafael MO, Dayvid VR Oliveira, George DC Cavalcanti, and Robert Sabourin. "FIRE-DES++: Enhanced online pruning of base classifiers for dynamic ensemble selection." *Pattern Recognition* 85 (2019): 149-160.

[5] Cruz, Rafael MO, Luiz G. Hafemann, Robert Sabourin, and George DC Cavalcanti. "DESlib: A Dynamic ensemble selection library in Python." *Journal of Machine Learning Research* 21, no. 8 (2020): 1-5.

[6] Cruz, Rafael MO, Hiba H. Zakane, Robert Sabourin, and George DC Cavalcanti. "Dynamic Ensemble Selection VS K-NN: why and when Dynamic Selection obtains higher classification performance?." In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1-6. IEEE, 2017.

References

[7] Cruz, Rafael MO, Robert Sabourin, and George DC Cavalcanti. "Prototype selection for dynamic classifier and ensemble selection." *Neural Computing and Applications* 29, no. 2 (2018): 447-457.

[8] Cruz, Rafael MO, Robert Sabourin, and George DC Cavalcanti. "META-DES. Oracle: Meta-learning and feature selection for dynamic ensemble selection." *Information fusion* 38 (2017): 84-103.

[9] Brun, André L., Alceu S. Britto Jr, Luiz S. Oliveira, Fabricio Enembreck, and Robert Sabourin. "A framework for dynamic classifier selection oriented by the classification problem difficulty." *Pattern Recognition* 76 (2018): 175-190.

[10] Ko, Albert HR, Robert Sabourin, and Alceu Souza Britto Jr. "From dynamic classifier selection to dynamic ensemble selection." *Pattern recognition* 41, no. 5 (2008): 1718-1731.

[11] Britto Jr, Alceu S., Robert Sabourin, and Luiz ES Oliveira. "Dynamic selection of classifiers—a comprehensive review." *Pattern recognition* 47, no. 11 (2014): 3665-3680.

[12] Woloszynski, Tomasz, and Marek Kurzynski. "A probabilistic model of classifier competence for dynamic ensemble selection." *Pattern Recognition* 44, no. 10-11 (2011): 2656-2668.

Dynamic Classifier Selection: Recent advances and perspectives

Rafael M. O. Cruz[1], Robert Sabourin[1] and George D. C. Cavalcanti[2]
[1] Laboratoire d' imagerie, de vision et d' intelligence artificielle (LIVIA) - École de Technologie Supérieure (ÉTS), Canada
www.livia.etsmtl.ca
[2] Centro de informática – Universidade Federal de Pernambuco, Recife – Brazil
www.cin.ufpe.br/~viisar