On the sublinear evolutionary design of Error Correcting Output Codes

Name: Miguel Ángel Bautista Martin, member of the BCN Perceptual Computing Lab.

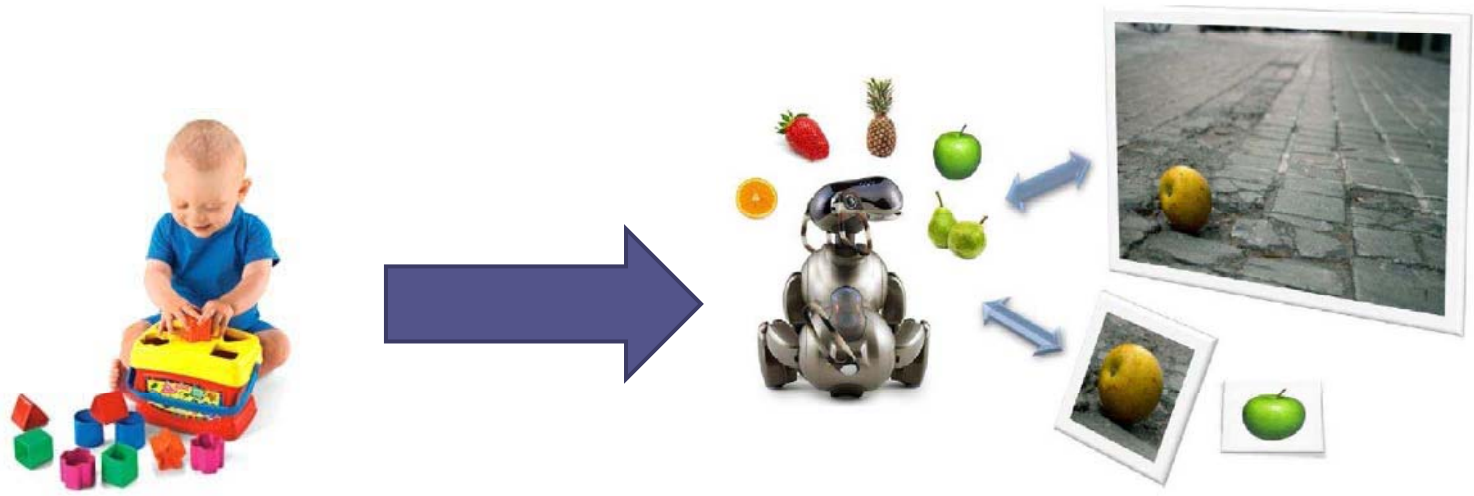
Directors: Dr. Sergio Escalera Guerrero & Dr. Xavier Baró i Solé

Outline

- Categorization problems
- Motivation
- Error Correcting Output Codes
- Support Vector Machines
- Evolutionary optimization
- Results
- Conclusions & Future work

Classification problems

- Humans are involved in classification tasks from their early days.
- Classification is an unavoidable task in intelligent systems.



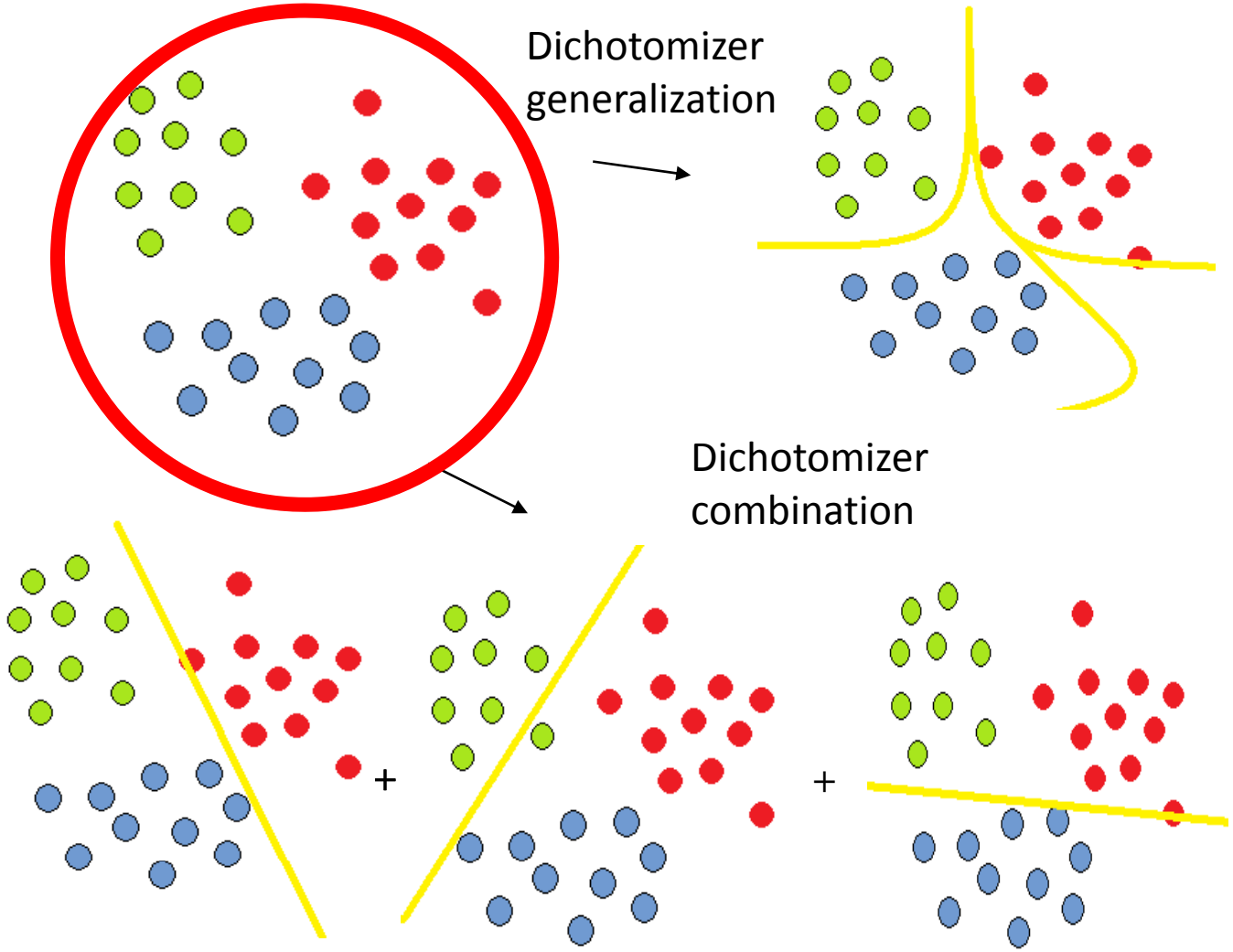
Multiclass categorization problems

- The complexity of the classification task, depends on the number of classes to discriminate.

- Internet and the communication era have brought new problems to the field, called large-scale problems (hundreds of categories).

Treating multiclass problems

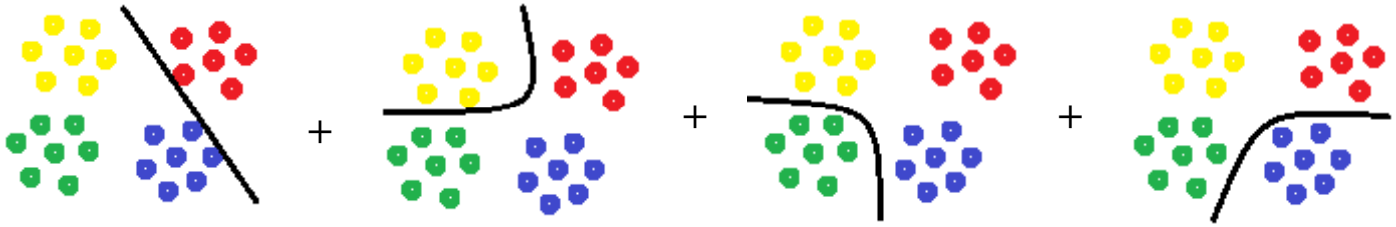
Distinguish the different dots



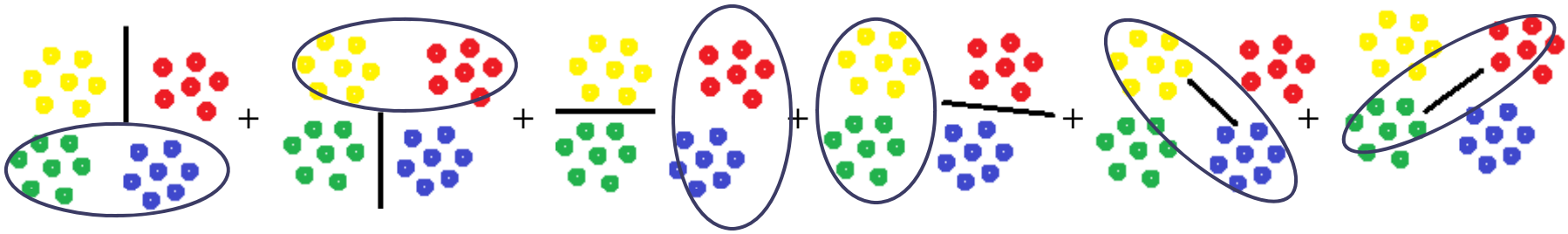
Combination of classifiers

- Several ways to combine binary **classifiers** to treat muticlass problems.

One versus All

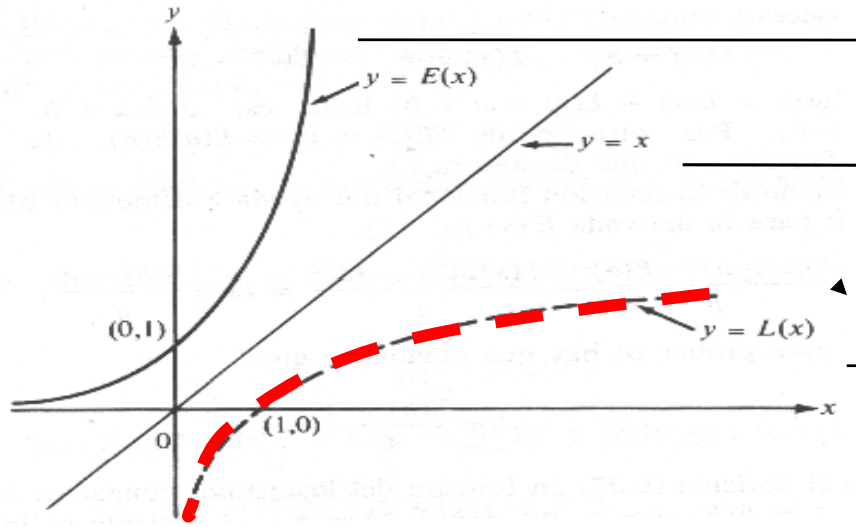


One versus One



Motivation

- The number of classifiers needed by a problem with a big number of classes is huge.
- Can the problem be solved with less classifiers?



One-versus-one

One-versus-all

Theoretical lower bound

Motivation

- Calculating the number of dichotomizers needed.

UCI ML Rep

#Clases	***3***	***8***	***10***	***11***	***184***
One Vs One	3	28	45	55	16386
One Vs All	3	8	10	11	168
Lower bound	2	3	4	4	**8**

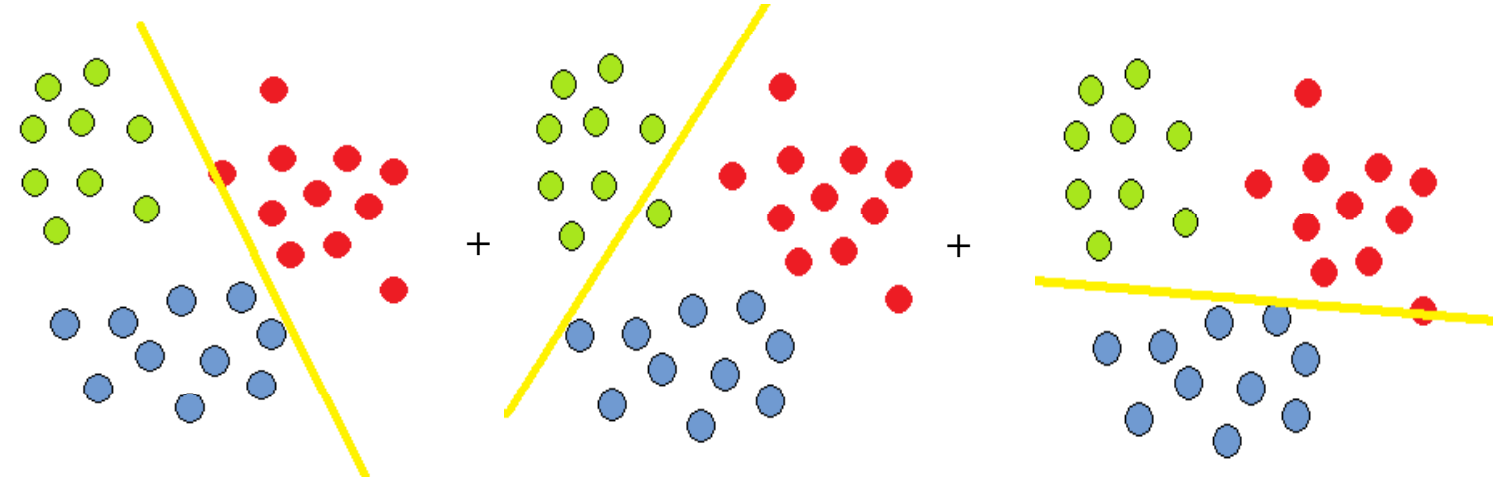
UCI ML Rep : University of California Irvine Machine Learning Repository

Goal

- Treat the problem with the minimal number of classifiers.
- Compensate the lost of generalization of the design by making it problem-dependent.
- This design is recommended when dealing with large-scale problems.

Error Correcting Output Codes (ECOC)

- ECOC are a ensemble learning methodology which allows to combine dichotomizers to treat multiclass problems.

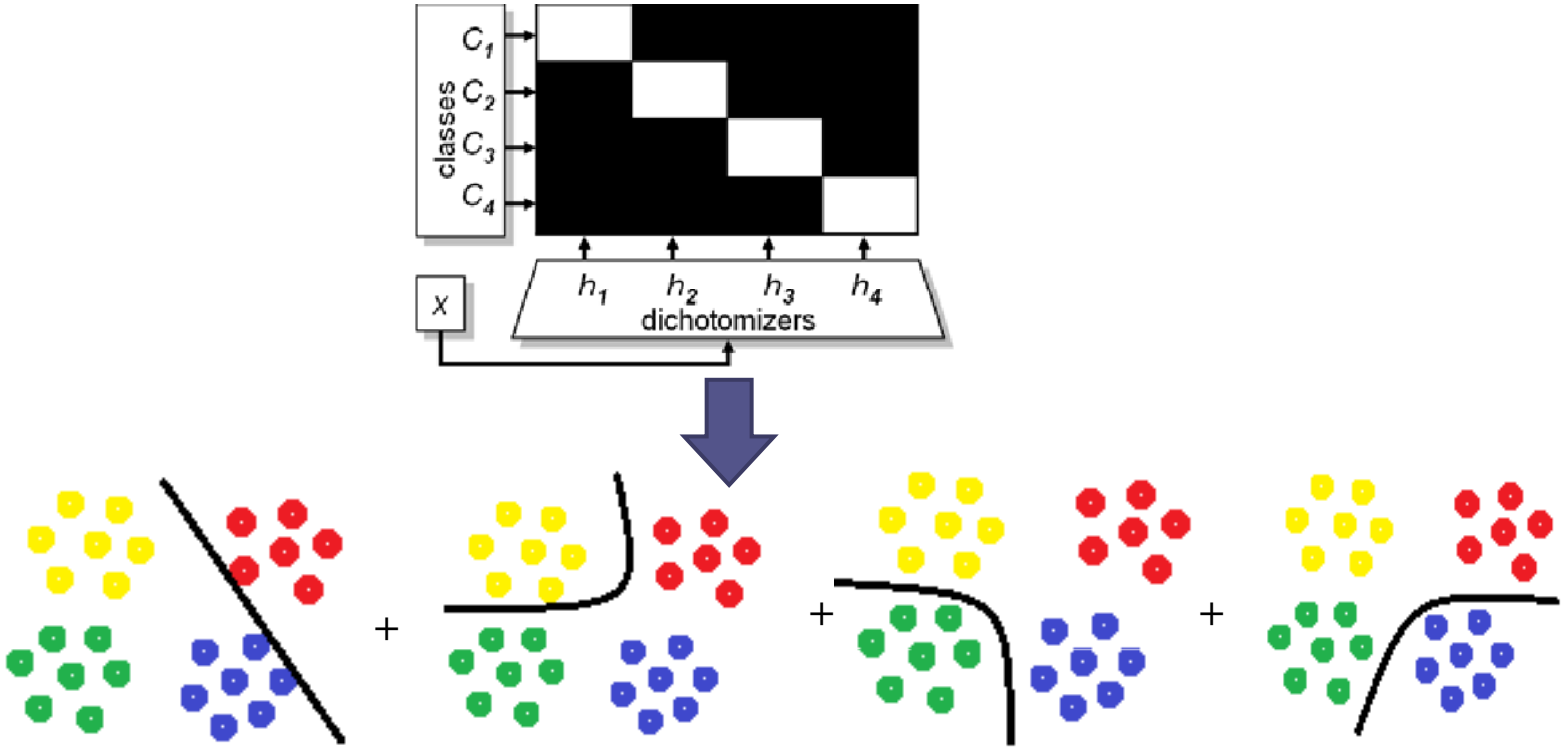


ECOC coding

- ECOCs can be represented as matrices, which columns represent the different sub-problems to treat.
- Each column has values that distinguish categories in two groups.
- In some codings an ignore value is introduced.

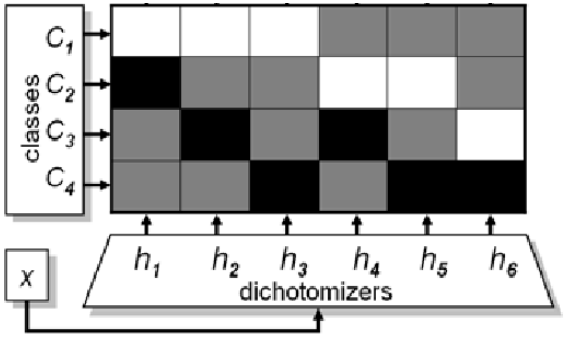
One versus All coding

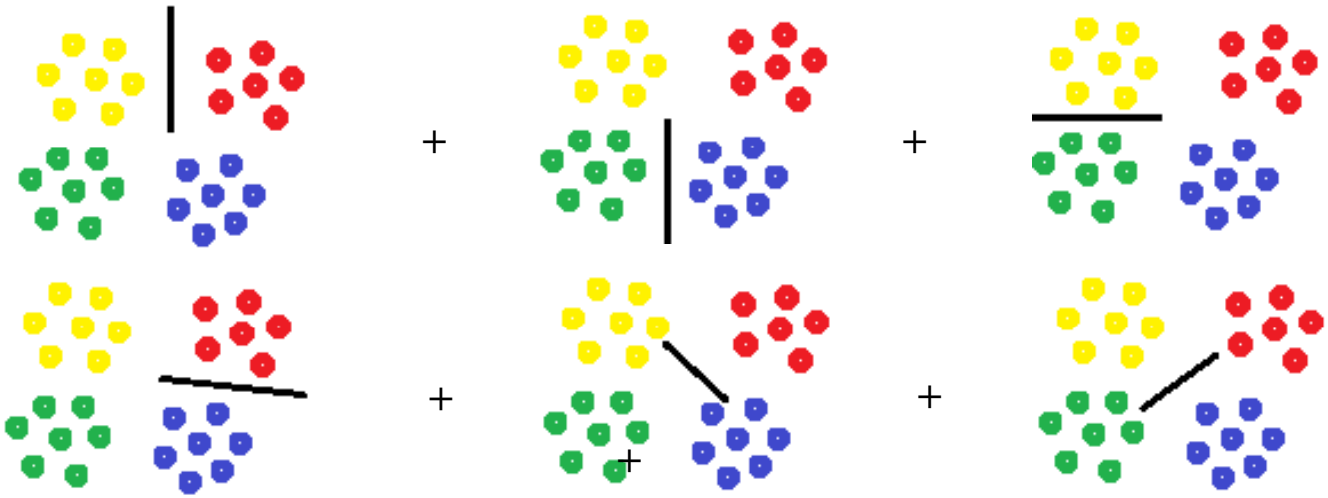
- Each category is discriminated of the rest.
- Only two values are necessary in the coding matrix, therefore this is a binary coding.



One versus One coding

- Each class is distinguish from another, therefore we ignore some classes (third symbol).

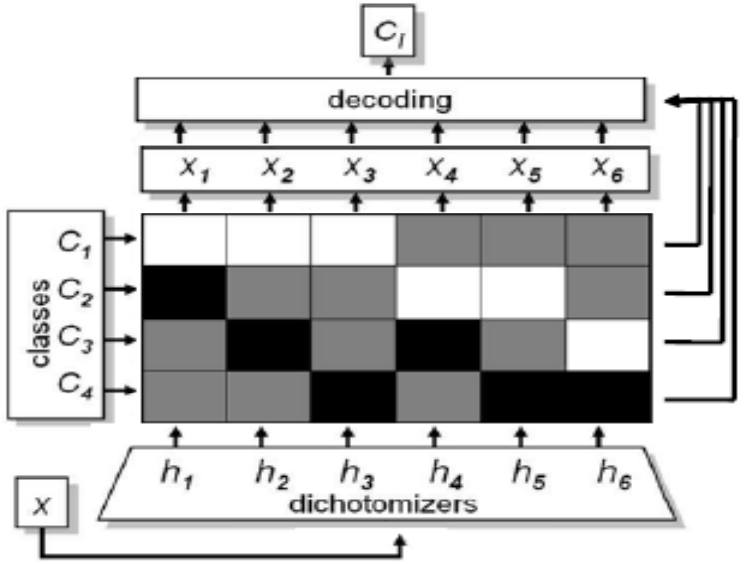


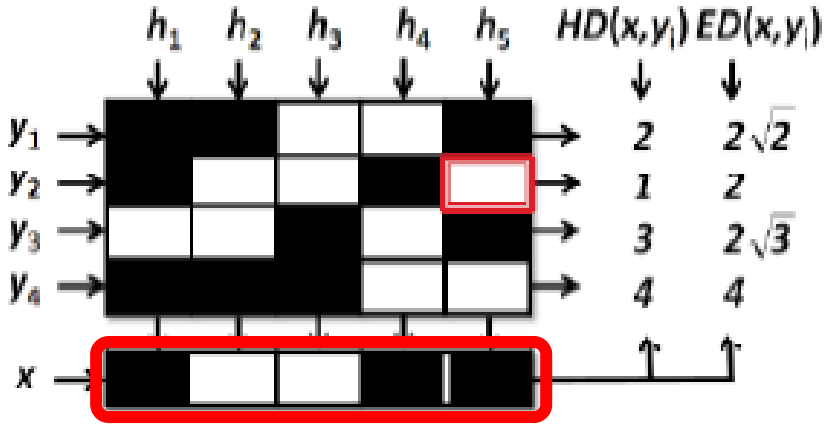


ECOC decoding

- Each base classifier gives its prediction and the set of prediction are compared to the codewords.

- Various types of decoding based on Euclidean and Hamming distances (only binary codings).





ECOC decoding

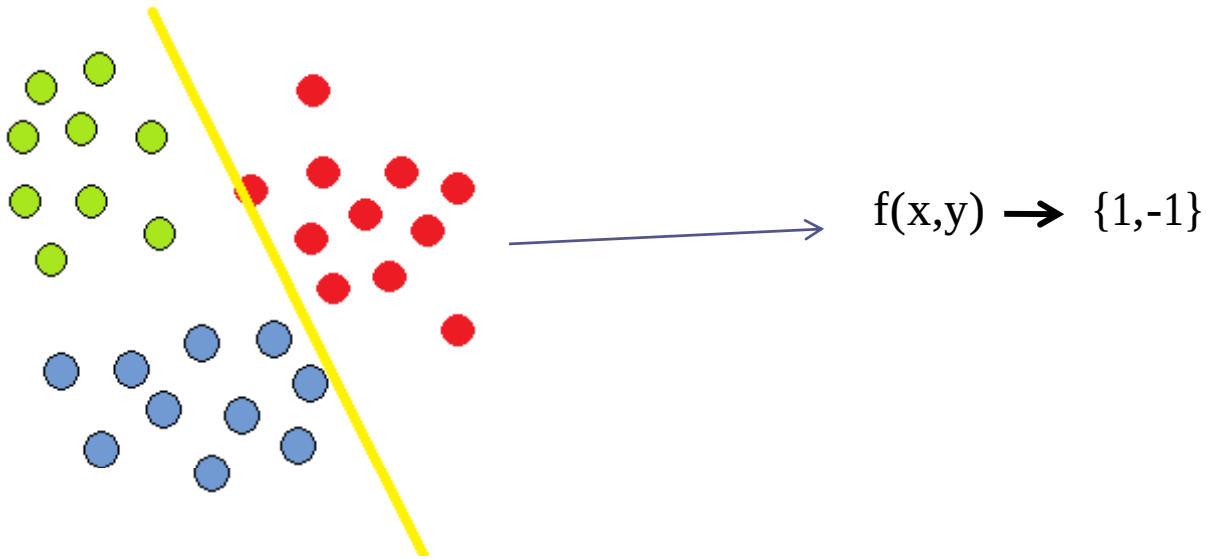
- Recently a new decoding design based on a loss function has appeared.

$$LB(\rho, y_i) = \sum_{j=1}^{n} L(y_i^j \cdot f^j(\rho))$$

- Useful for ternary codings

ECOC base classifiers

- Every sub-problem defined by the columns of the coding matrix is treated by a base classifier.



Support Vector Machines

- Support Vector Machines are classifiers that have shown a good performance on literature.
- Find a linear function that discriminates the two categories and maximizes the distance between the closests point of each class, called the margin.
- Problem: data may not be linearly separable.

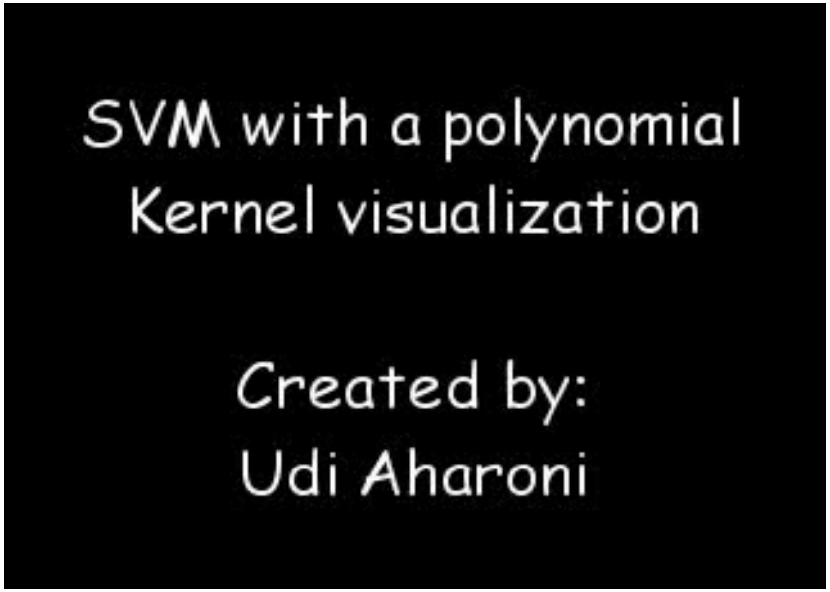
High dimension generalization

- If data is not linearly separable in the input space, the SVM can map the data in a high dimension space.

- This mapping is done by a function called *kernel*, which defined a scalar product over the data points in the high dimension space.

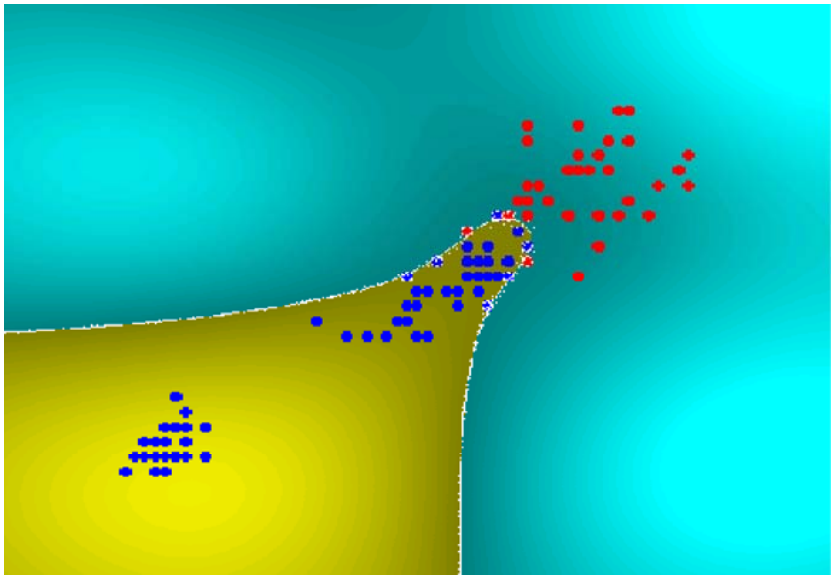
Kernels

- In SVMs a set of kernels can be used in order to map the data.
- Most of them are well-know functions : Polynomios , Radial Basis Functions, Sigmoidal functions



Gaussian RBF kernel

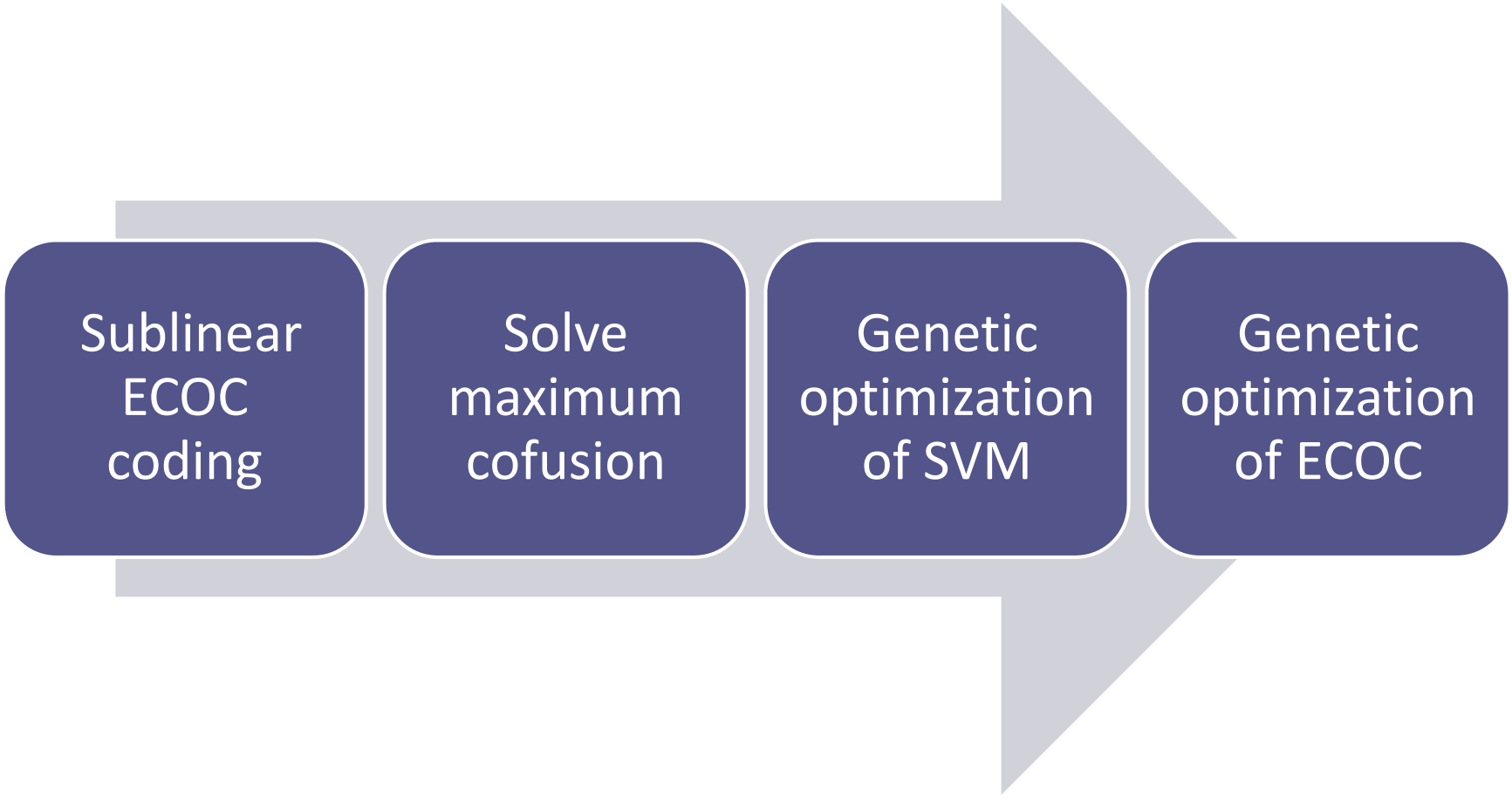
- By using a Gaussian RBF kernel we can obtain very good results when treating binary problems.
- This type of SVM needs the parameters (C & Gamma) to be optimized.



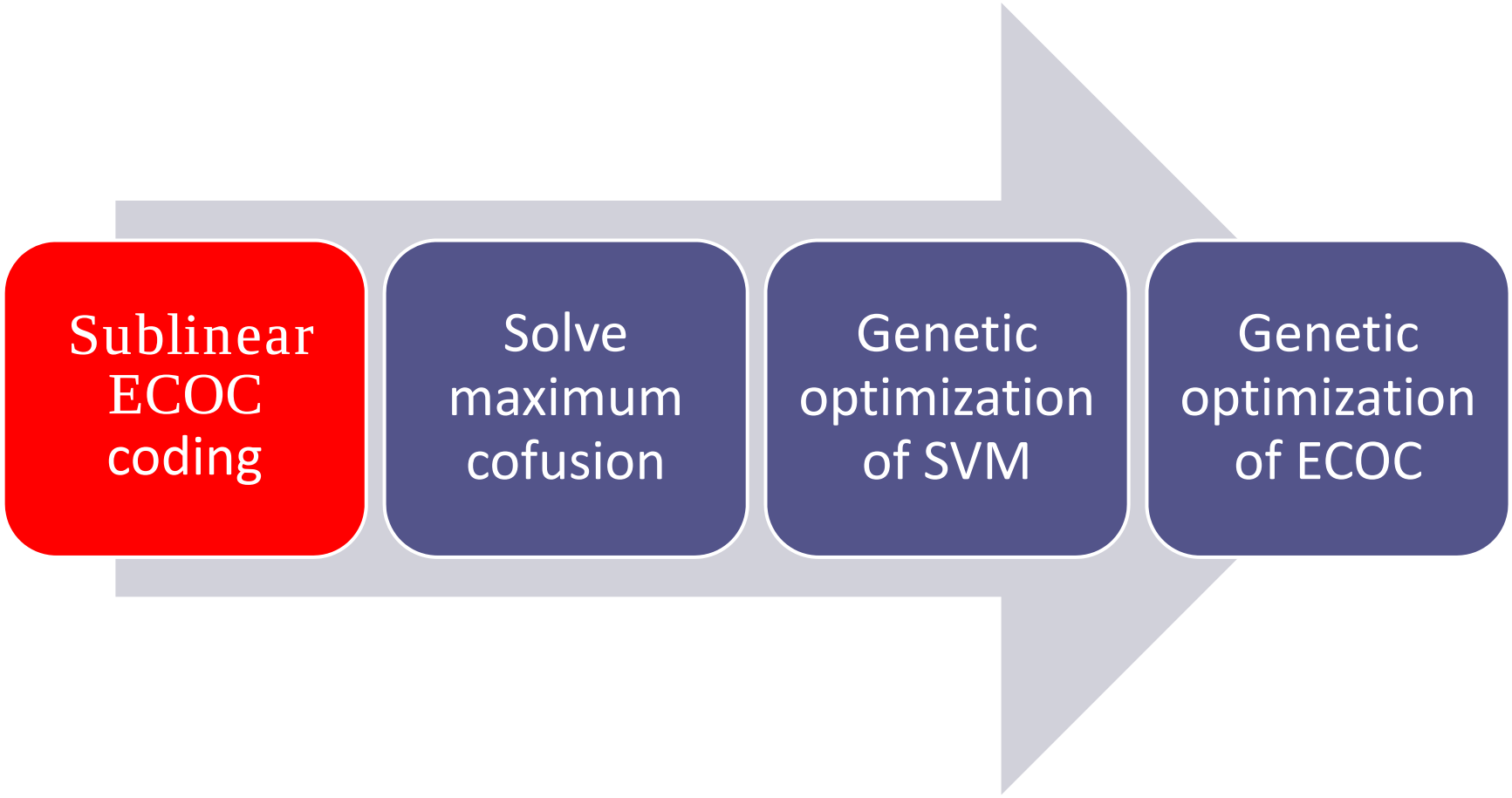
Goals

- Treat the problem with the minimal number of classifiers.
- Compensate the lost of generalization of the design by making it problem-dependent.
- This design is recommended when dealing with large-scale problems.

Global overview



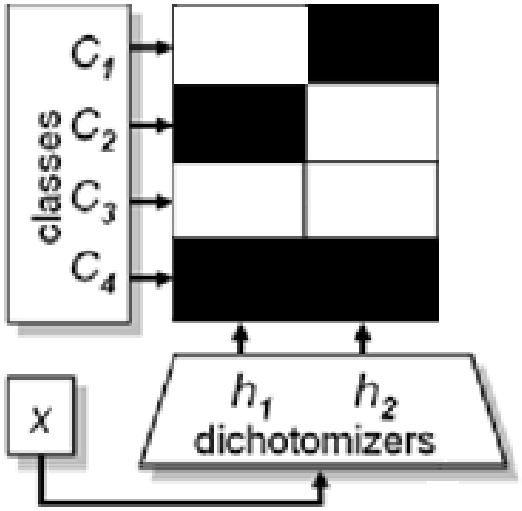
Global overview



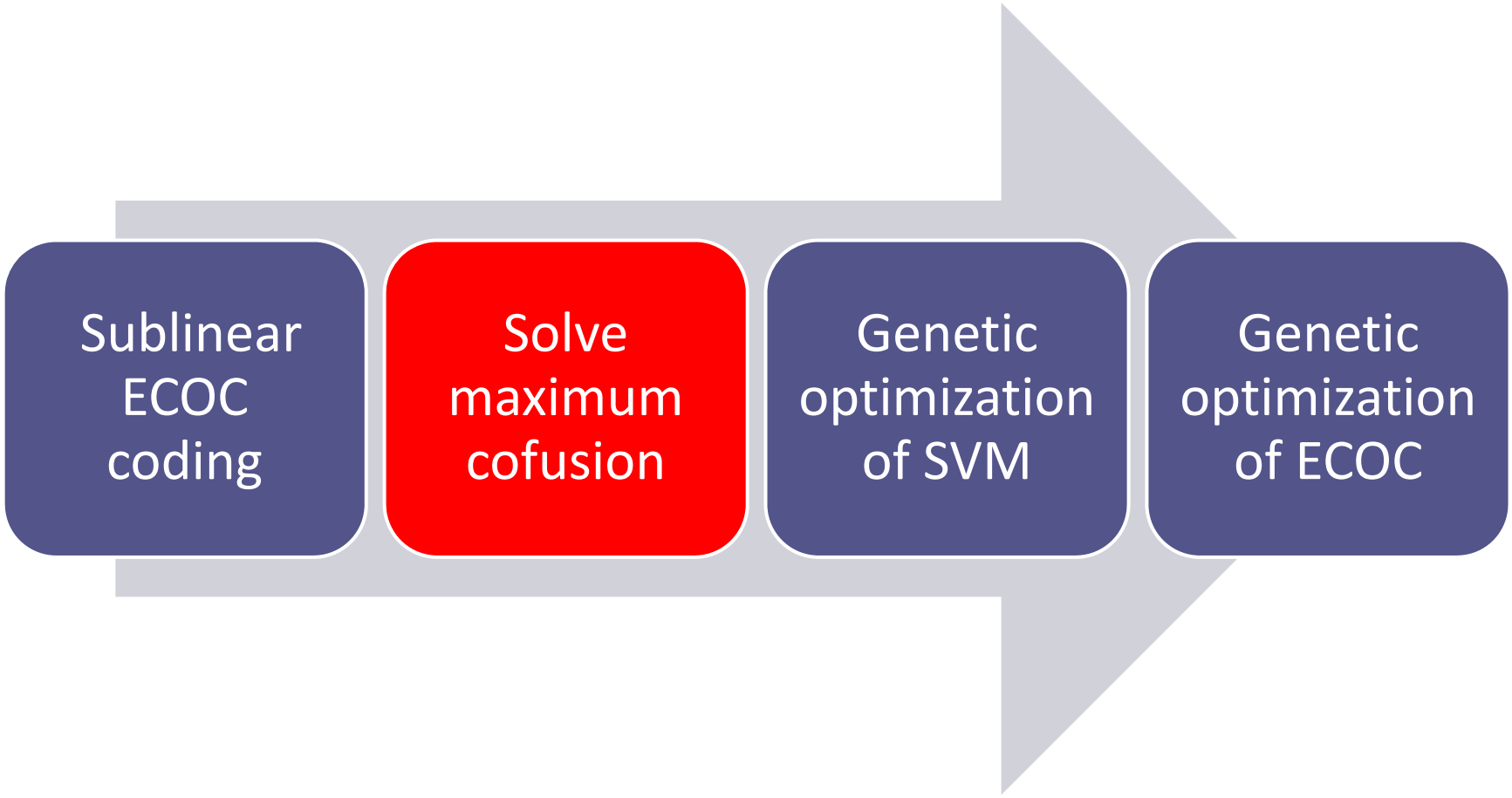
Sublinear incremental coding

- Define the minimal number of base classifiers needed to discriminate *N* categories.

- Sublinear coding

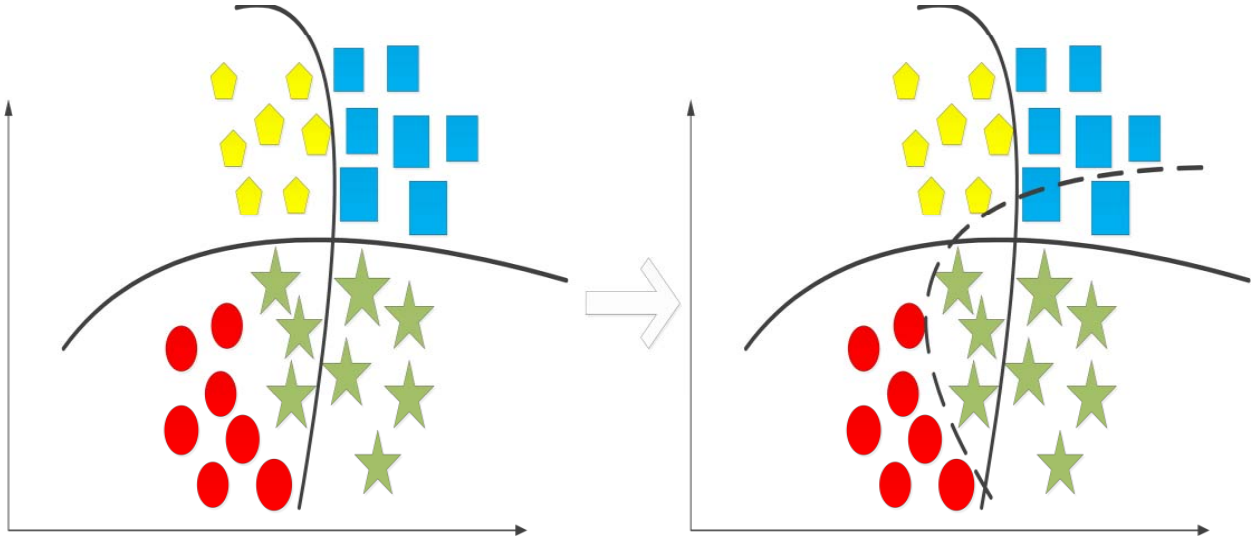


Global overview



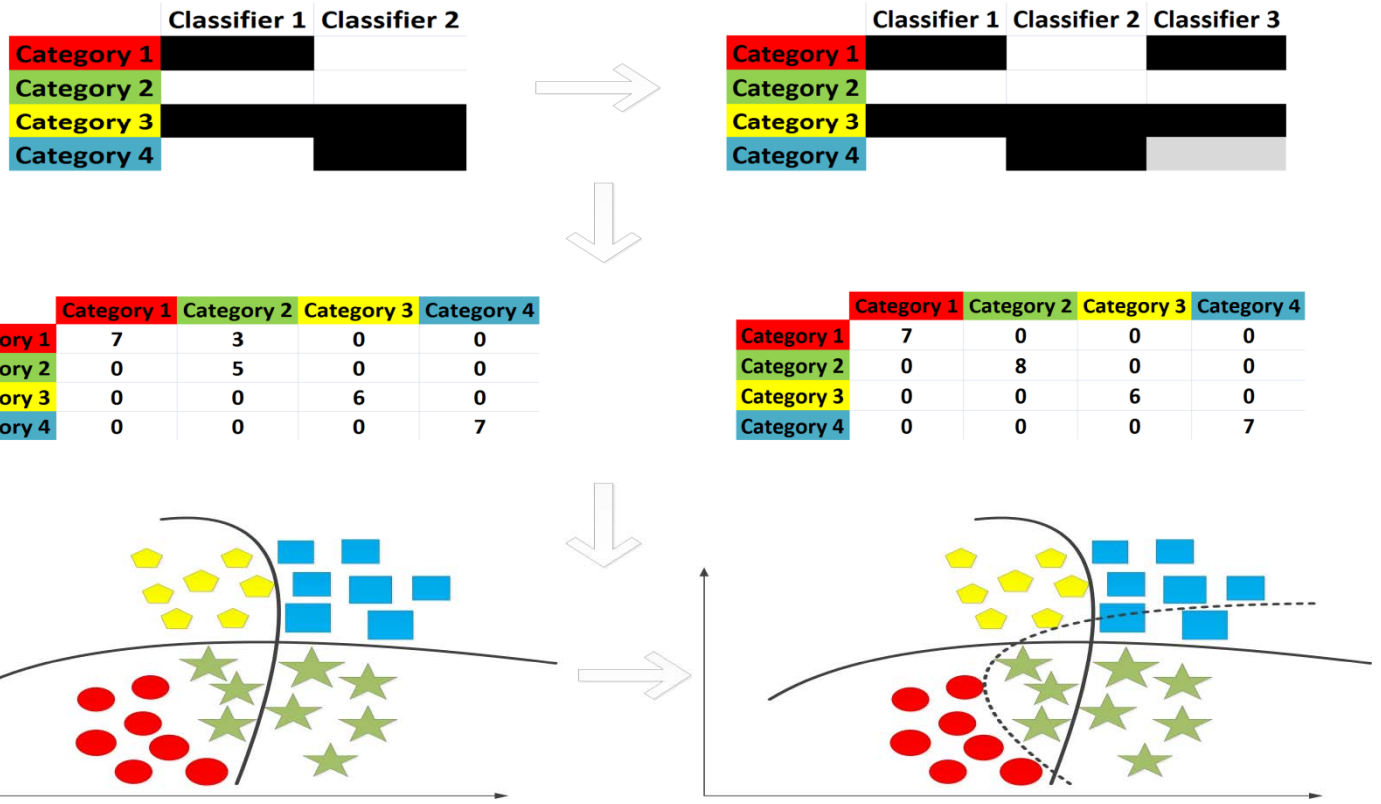
Sublinear incremental coding

- Solve the problem that generates the maximum lost of performance, identify the classes that generate it.
- Add a new classifier focused on those classes, a solver for this confusion.
- Various solvers (One vs One or Confusion-based).



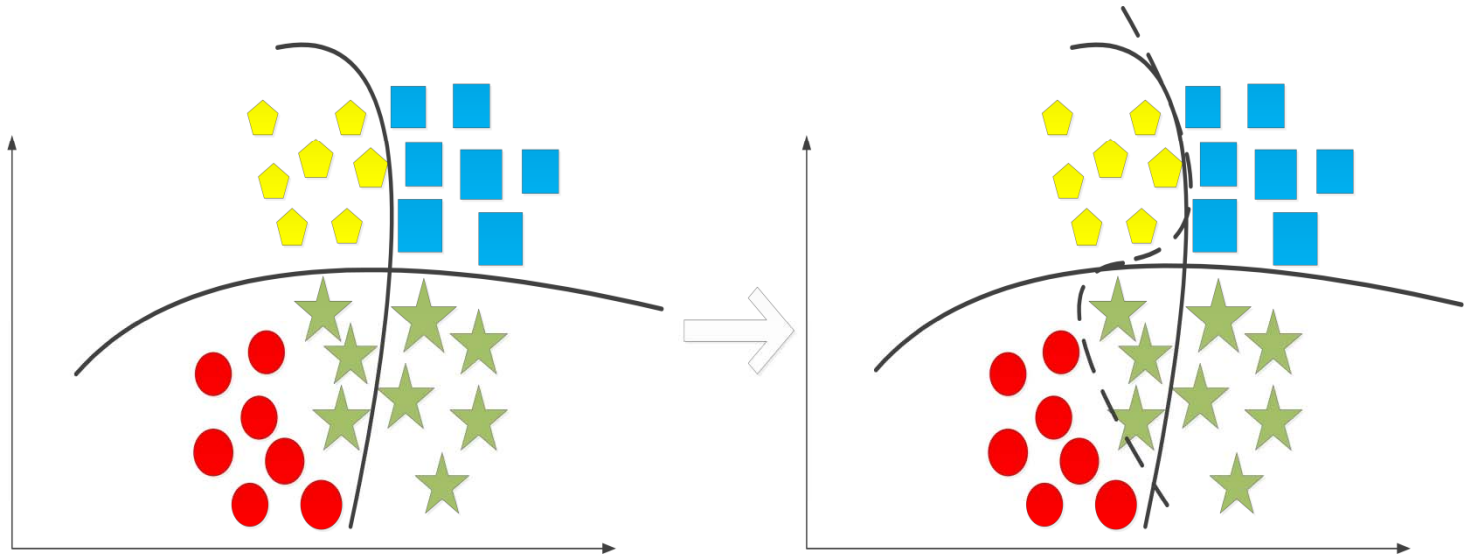
One vs One solver

- Treat only the classes that maximize confusion ignoring the others.

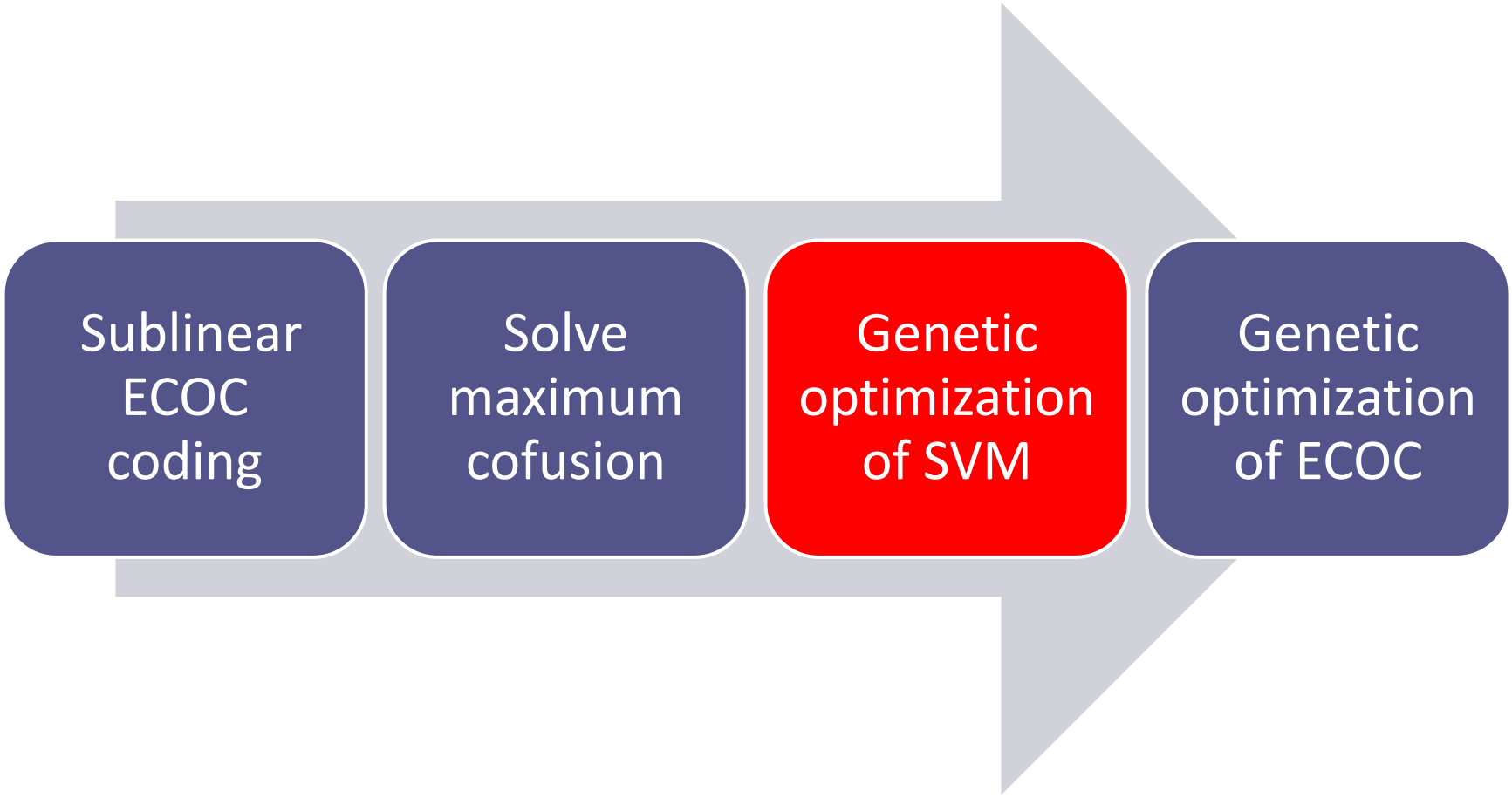


Confusion-based solver

- Support the maximum confused classes with the ones that have less confusion with each one of them.

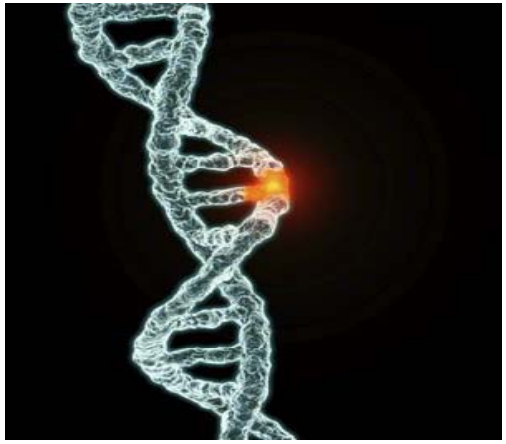


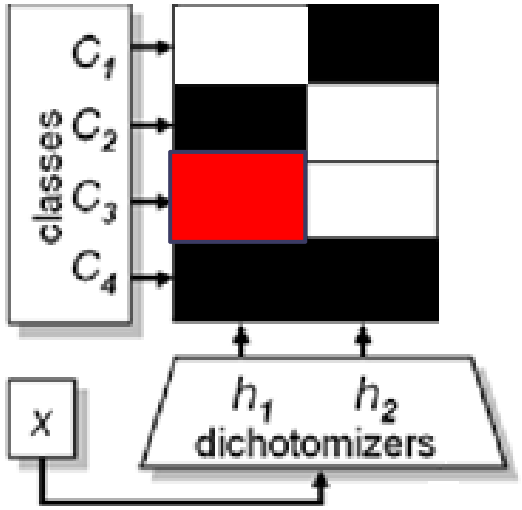
Global overview



Genetic algorithms

- Optimization algorithms based on the evolution theory of Darwin.
- See solutions as individuals and operate with them (crossover & mutation).
- Recommendable method when the space is not continuous neither differentiable.

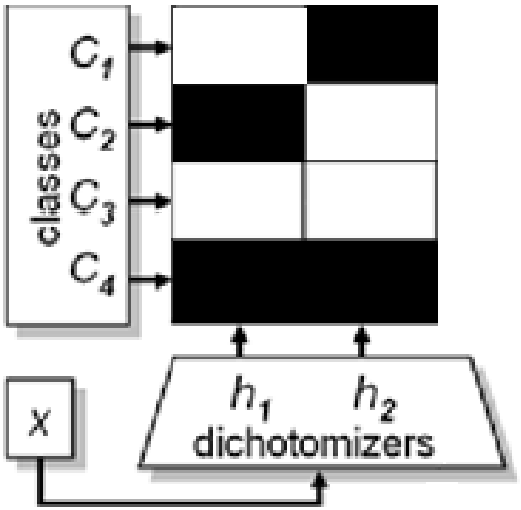




Genetic coding

- Each solution is coded and the result is called a genotype, which represents the individual.

- The standard coding design is the binary one (see the solution as a binary vector)

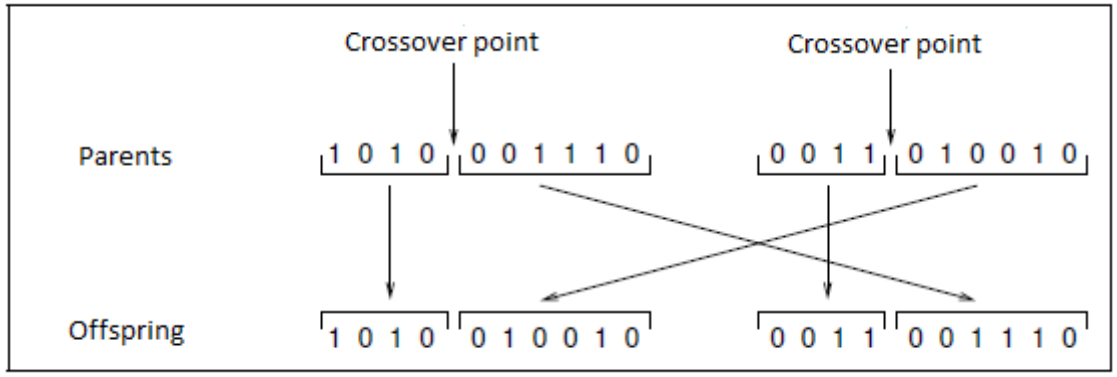


10101001

Genetic operators

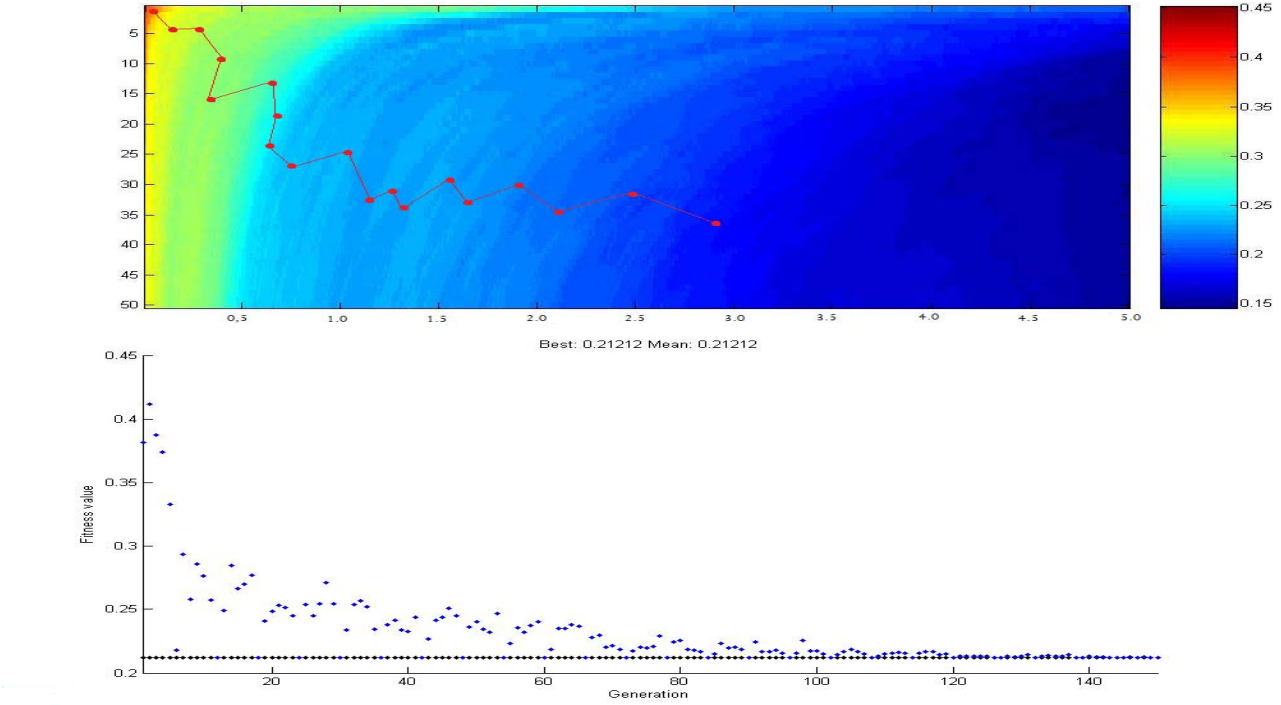
- Seeing solutions as individuals or genotypes, standard operators are used to perform crossover and mutation operations.

- For instance, single point crossover

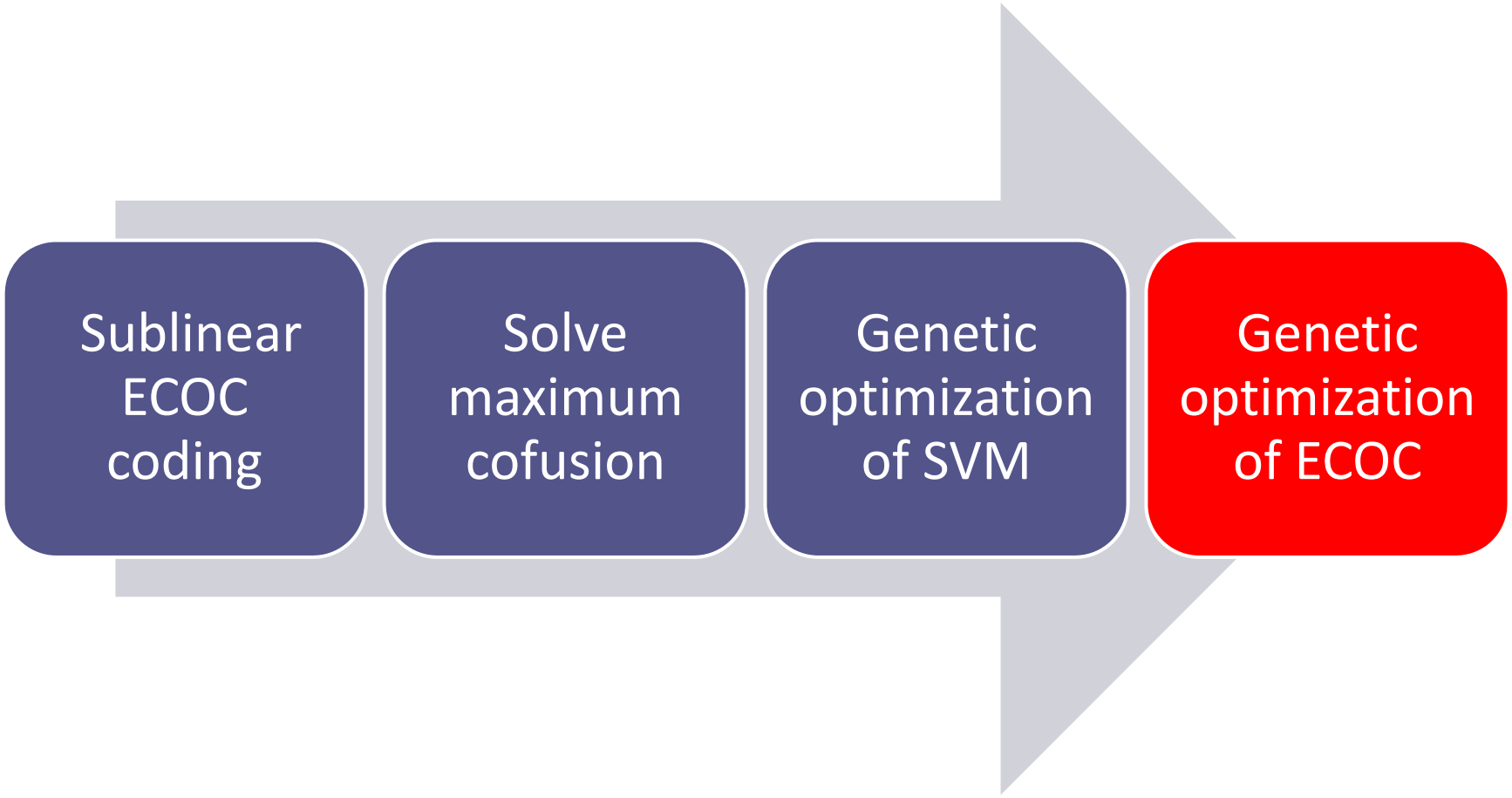


Genetic optimization

- An inner optimization process is also carried out to tune the parameters of the base classifiers.
- SVM-RBF classifiers have mainly 2 parameters (C & Gamma).



Global overview



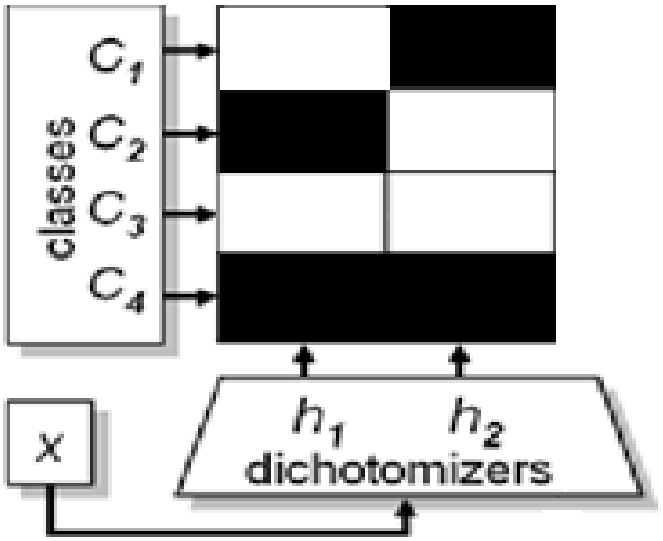
Genetic operators problem

- If the population of individuals is a little complex or have constraints, the generic genetic operators are useless, they can generate individuals that violate constraints.

- Specific genetic operators must be defined in order to have a powerful optimization tool.

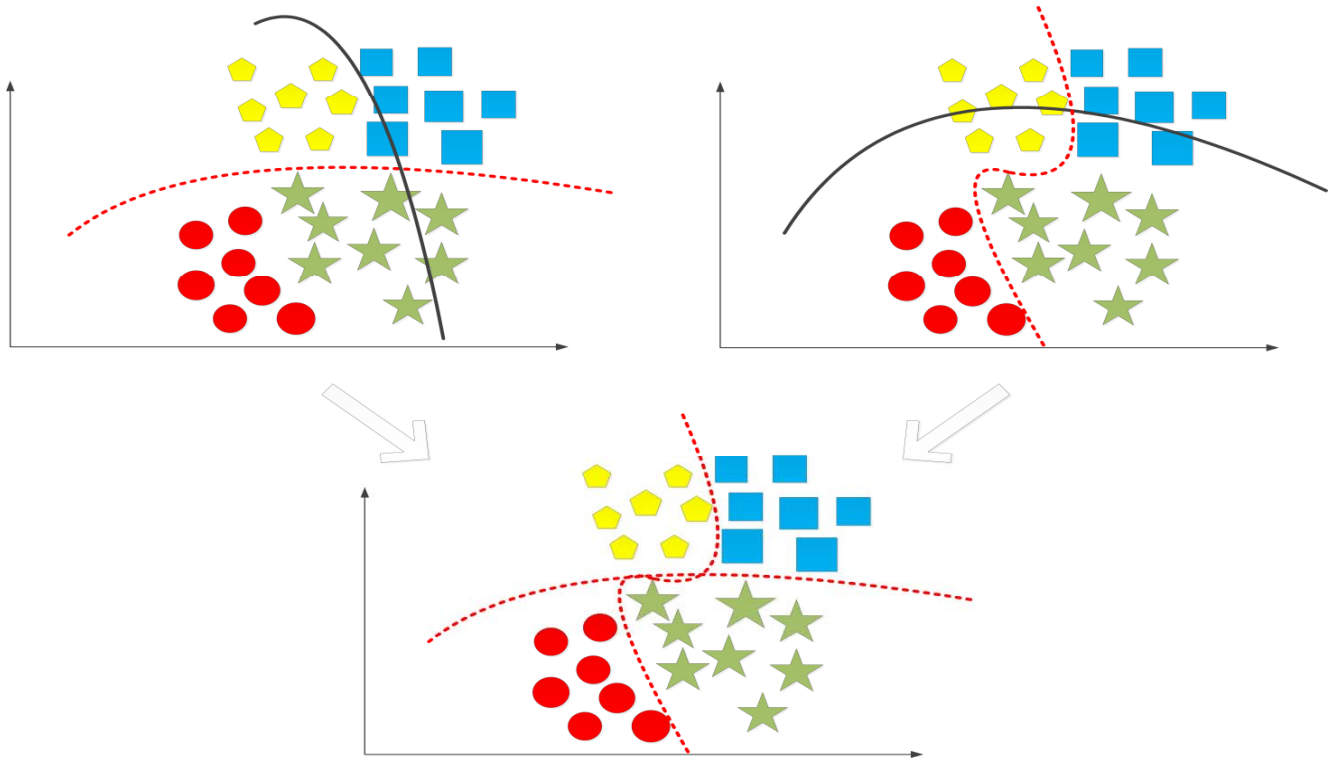
ECOC constraints

- The coding matrix has to univocally distinguish every codeword, which means that every row has to be different.



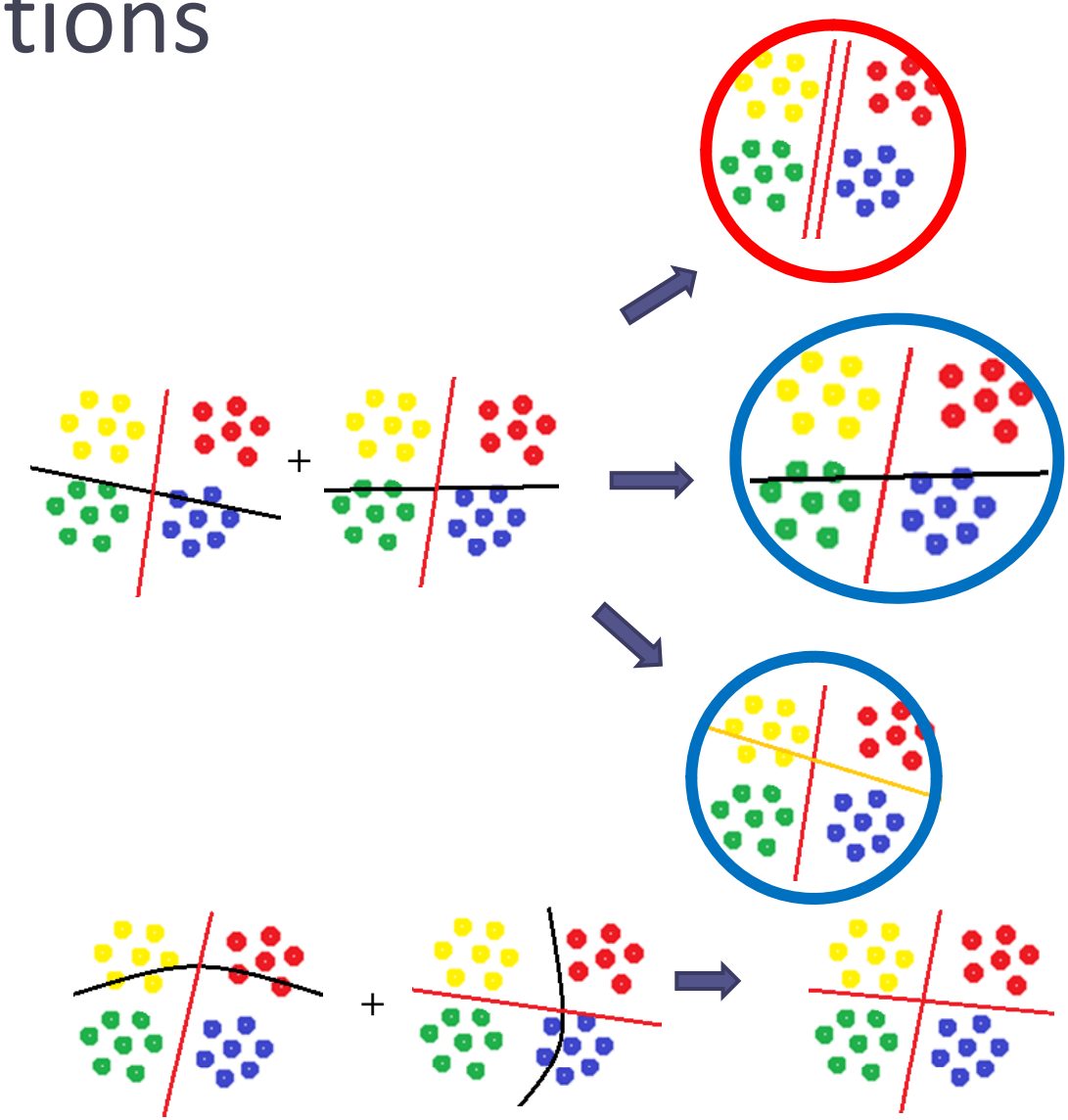
ECOC-specific crossover

- Define a crossover operator for ECOCs which makes sense (always return a ECOC offspring).



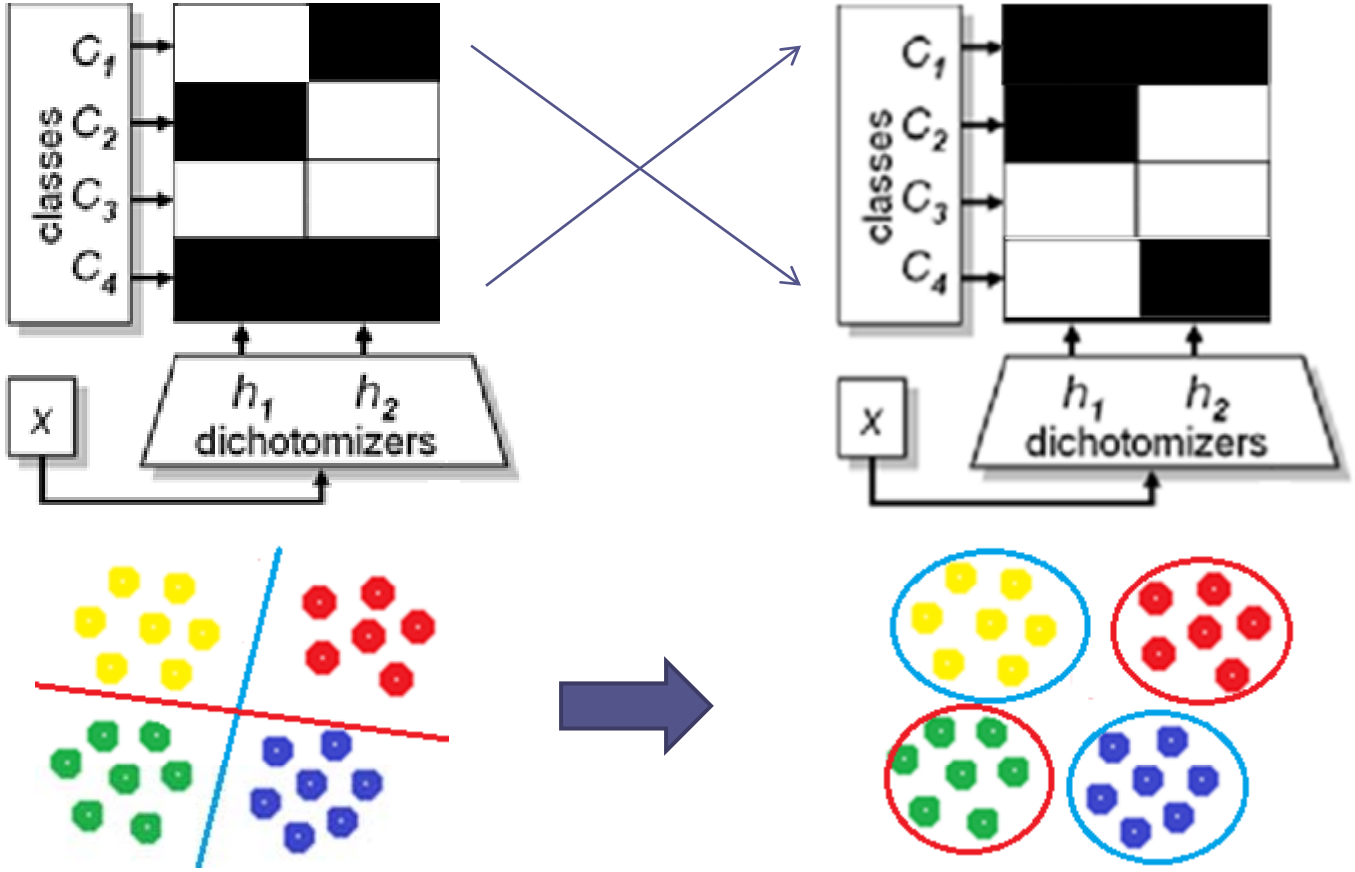
Crossover situations

1. Detect best classifiers.
2. Check if the classifiers are compatible.
3. If there are not compatible
    1. return to step 1 and look for the following best classifier.
    2. If there are not compatible classifiers left , one classifier has to be created.
4. If they are compatible,aggregate them in the field.



ECOC-specific mutation

- Randomly swap to rows of the coding matrix.



Experiments

- The first experiment was performed over a set of 12 dataset from the UCI Machine Learning Repository.

- The second experiment was performed over a set of 5 challenging computer vision problems.

Metrics

- Each experiment is validated through a Stratified 10-fold cross-validation.
- The results shown are the mean of the 10-fold performances.
- The Minimal coding proposal is compared with the standard ECOC codings.

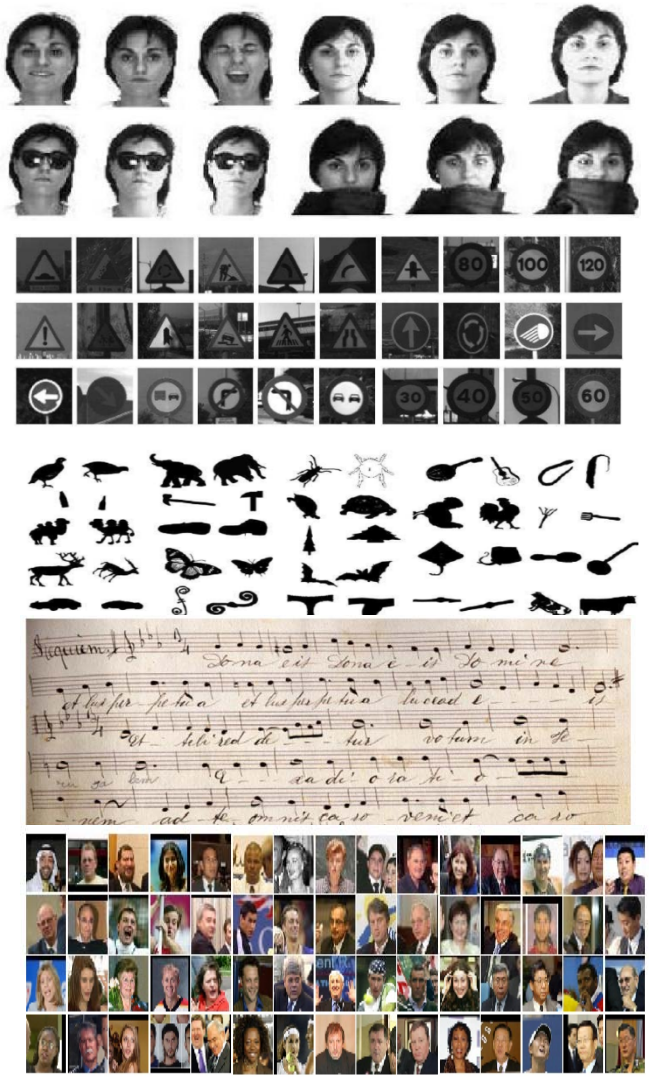
UCI datasets Characteristics

- UCI dataset characteristics

Problem	#Training samples	#Features	#Classes
Dermathology	366	34	6
Iris	150	4	3
Ecoli	336	8	8
Vehicle	846	18	4
Wine	178	13	3
Segmentation	2310	19	7
Glass	214	9	7
Thyroid	215	5	3
Vowel	990	10	11
Balance	625	4	3
Shuttle	14500	9	7
Yeast	1484	8	10

CV Dataset Characteristics

- ARFace:520 x 120, 20 classes.
- Traffic: 3481 x 100, 36 classes.
- MPEG: 1400 x 70, 20 classes.
- Cleafs: 4098x65, 7 classes.
- LFW: 6144x50, 184 classes.



Results on UCI problems

- As we can see the evolutive minimal performs similar to the standard codings.

	Binary Minimal ECOC		Evol. Minimal ECOC		one-vs-all ECOC		one-vs-one ECOC	
Data set	Perf.	Classif.	Perf.	Classif.	Perf.	Classif.	Perf.	Classif.
Derma	96.0±2.9	3	**97.2±2.1**	3	95.1±3.3	6	94.7±4.3	15
Iris	96.4±6.3	2	**98.2±1.9**	2	96.9±6.0	3	96.3±3.1	3
Ecoli	80.5±10.9	3	**81.3±10.8**	3	79.5±12.2	8	79.2±13.8	28
Vehicle	72.5±14.3	2	82.60±12.4	2	74.2±13.4	4	**83.6±10.5**	6
Wine	95.5±4.3	2	**98.3±2.3**	2	95.5±4.3	3	97.2±2.4	3
Segment	96.6±2.3	3	96.7±1.5	3	96.1±1.8	7	**97.18±1.3**	21
Glass	56.7±23.5	3	51.24±29.7	3	53.85±25.8	6	**60.5±26.9**	15
Thyroid	**96.4±5.3**	2	94.7±5.1	2	95.6±7.4	3	96.1±5.4	3
Vowel	57.7±29.4	3	80.3±11.1	3	**80.7±11.9**	8	78.9±14.2	28
Balance	80.9±11.2	2	87.1±9.2	2	78.9±8.4	3	**92.8±6.4**	3
Shuttle	80.9±29.1	3	83.4±15.9	3	**90.6±11.3**	7	86.3±18.1	21
Yeast	50.2±18.2	4	**53.7±11.8**	4	51.1±18.0	10	52.4±20.8	45
Rank & #	2.8	**2.7**	**1.9**	**2.7**	2.9	5.7	2.3	15.9

Results on CV problems

- In this set of experiments we can see how althought the Minimal ECOC is slightly worst it uses far less number of classifiers.

	Binary M. ECOC		GA M. ECOC		one-vs-all		one-vs-one	
Data set	Perf.	#	Perf.	#	Perf.	#	Perf.	#
FacesWild	26.4±2.1	8	**31.7±2.3**	8	25.0±3.1	184	-	16836
Traffic	90.8±4.1	6	90.6±3.4	6	**91.8±4.6**	36	90.6±4.1	630
ARFaces	76.0±7.2	5	85.84.0±5.2	5	84.0±6.3	20	**96.0±2.5**	190
Clefs	81.2±4.2	3	**95.6±9.3**	3	80.8±11.2	7	84.2±6.8	21
MPEG7	89.29±5.1	7	90.4±4.5	7	87.8±6.4	70	**92.8±3.7**	2415
Rank & #	3.6	**6.2**	2.0	**6.2**	3.8	148.6	**1.75**	4018.4

Conclusions

- We propose a novel methodology to deal with large-scale problems.
- The ECOC framework is used.
- Each base classifier is a SVM with an RBF kernel.
- Optimization of SVM parameters is performed through a genetic algorithm.
- A coding matrix with high generalization capability is obtained via a genetic algorithm.
- New ECOC-specific crossover and mutation algorithms are designed.

Conclusions

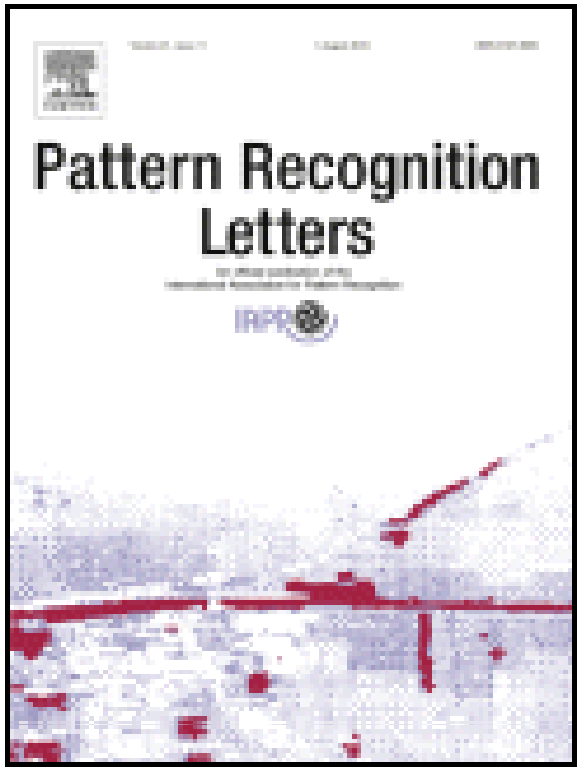
- The use of minimal coding, reaches same performances (or better) as other coding with far less cost.
- The computational time of evolution is reduced due to the use of the ECOC-specific operators.

Future work

- Introduce online techniques to support a great number of categories and samples by performing incremental learning.

- Redesign crossover and mutation operators in order to have a complete functional optimization process.

Submission to Pattern Recognition Letters Journal



Minimal Design of Error-Correcting Output Codes

Miguel Ángel Bautista, Xavier Baró, Oriol Pujol, Petia Radeva, Jordi Vitrià, Sergio Escalera

Centre de Visió per Computador, Campus UAB, Edifici O, 08193 Bellaterra (Barcelona), Spain.
Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via 585, 08007, Barcelona, Spain.

Abstract

A challenging trend in Pattern Recognition consists of dealing with the classification of large number of object categories. In literature, this problem is often addressed using an ensemble of classifiers. In this scope, the Error-Correcting Output Codes framework has demonstrated to be a powerful tool for the combination of classifiers. However, most of the state-of-the-art ECOC approaches use a linear or exponential number of classifiers, making the discrimination of a large number of classes unfeasible. In this paper, we propose a minimal design of ECOC in terms of the number of classifiers. Evolutionary computation is used for tuning the parameters of the classifiers and looking for the best Minimal ECOC code configuration. The results over several public UCI data sets and different multi-class Computer Vision problems show that the proposed methodology obtains comparable (even better) results than state-of-the-art ECOC methodologies with far less number of dichotomizers.

1 Introduction

Nowadays challenging applications of Pattern Recognition deal with changing environments, online adaptations, contextual information, etc. In order to deal with all these problems, efficient ways for processing huge amount of data are often required. One clear example is the case of general Pattern Recognition algorithms for classification, especially when the number of categories, namely objects, people, brands, etc. is arbitrarily large. Usual machine learning strategies are effective for dealing with small number of classes. The choices are limited when the number of classes becomes large. In that case, the natural algorithms to consider are those that model classes in an implicit way, such as instance based learning (i.e. nearest neighbors). However, this choice is not necessarily the most adequate for a given problem. Moreover, we are forgetting many algorithms of the literature such as ensemble learning (i.e. Adaboost [1]) or kernel based discriminant classifiers (i.e. support vector machines [2]) that have been proven to be very powerful tools.

Thank you!

QUESTIONS?