Mini Tutorial

Three Hours on Multiple Classifier Systems

Lecturer

Fabio Roli

University of Cagliari

Dept. of Electrical and Electronics Eng., Italy

email roli@diee.unica.it

Terminology

According to Nello's terminology, today we speak of *pattern matching* with multiple classes

also called *pattern classification* [Duda,Hart,Stork]

Terminology

Multiple classifier systems is only one of the "multiple" names used for the topic of this mini tutorial

- ✓combination of multiple classifiers [Lam95,Woods97,Xu92,Kittler98]
- ✓ensemble learning [Jordan95]
- ✓classifier fusion [Cho95,Gader96,Grabisch92,Keller94,Bloch96]
- ✓mixture of experts [Jacobs91,Jacobs95,Jordan95,Nowlan91]
- ✓consensus aggregation [Benediktsson92,Ng92,Benediktsson97]
- ✓voting pool of classifiers [Battiti94]
- ✓composite classifier systems [Dasarathy78]
- ✓classifier ensembles [Drucker94,Filippi94,Sharkey99]
- ✓modular systems [Sharkey99]
- ✓collective recognition [Rastrigin81,Barabash83]
- ✓stacked generalization [Wolpert92]
- ✓divide-and-conquer classifiers [Chiang94]
- ✓pandemonium system of reflective agents [Smieja96]
- ✓etc.

Tutorial Aims and Outline

- An introductive, three hours, tutorial on multiple classifiers

  - Part 1: Motivations and basic concepts
  - Part 2: Main methods for creating multiple classifiers
  - Part 3: Main methods for fusing multiple classifiers

PART I

Motivations and basic concepts

The traditional approach to Pattern Classification

- Unfortunately, no dominant classifier exists for all the data distributions ("no free lunch" theorem), and the data distribution of the task at hand is usually unknown

•CLASSIFIER EVALUATION AND SELECTION: *evaluation* of a set of different classification algorithms (or different "versions" of the same algorithm) against a *representative* pattern sample, and *selection* of the best one

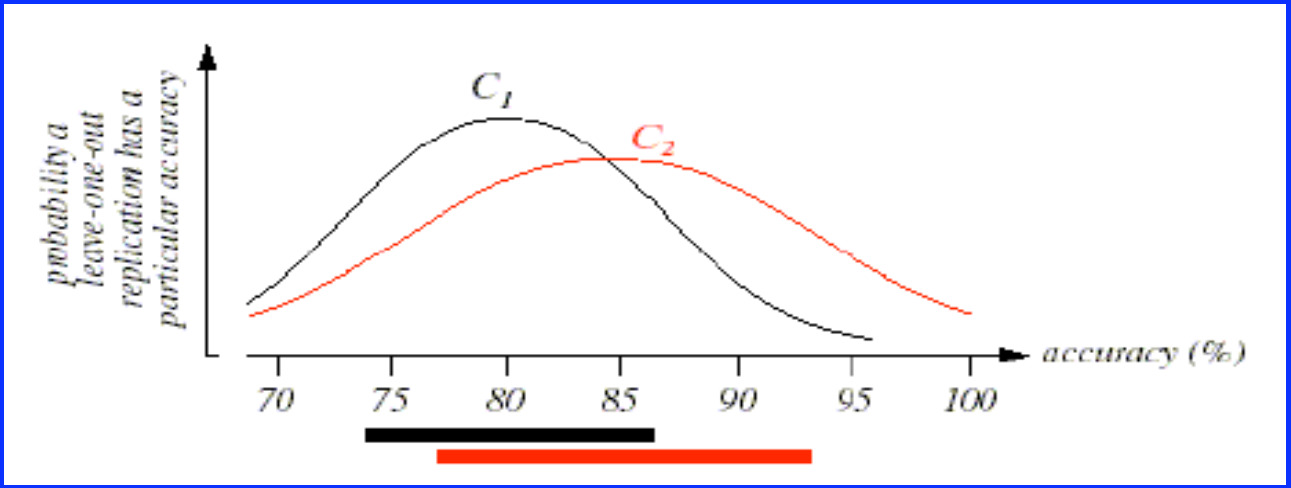
➢I design a set of N classifiers $C_1, C_2, \ldots, C_N$

➢I *evaluate* classifier errors $E_1 < E_2 < E_3 < \ldots < E_N$ (with related confidence intervals) using a validation set

➢I *select* the best classifier C_1 , and consider it the "optimal" one

The traditional approach: Small Sample Size Issue

- The traditional approach works well when a large and representative data set is available *("large" sample size cases)*, so that estimated errors allow to select the best classifier



•However, in many small sample-size real cases, validation set provides just *apparent* errors that differ from true errors E_i:

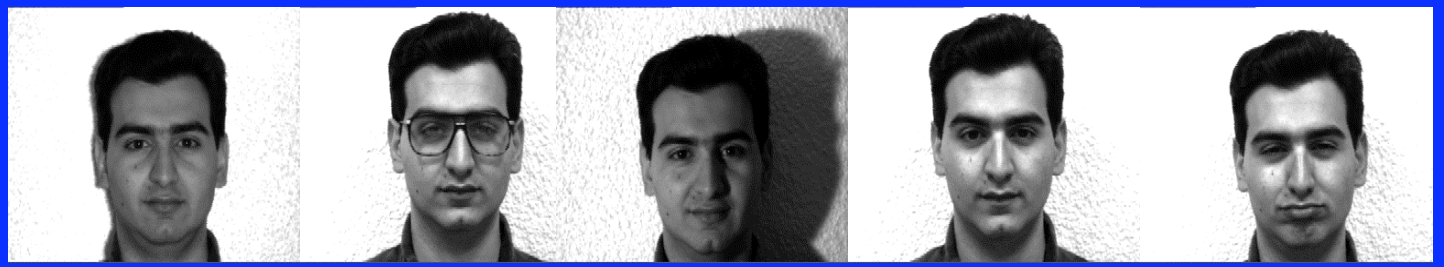
$$\hat{E}_i = E_i \pm \Delta_i$$

This can make impossible the selection of the optimal, if any, classifier, and, in the ***worst case****, I could select the* ***worst*** *classifier*

A practical example

Face recognition using PCA and LDA algorithms

Faces in the **validation set** (*Yale data base*)



High "Variance"

Faces in the **test set**

Apparent error caused from poorly representative validation set can make impossible to select the best one between PCA and LDA

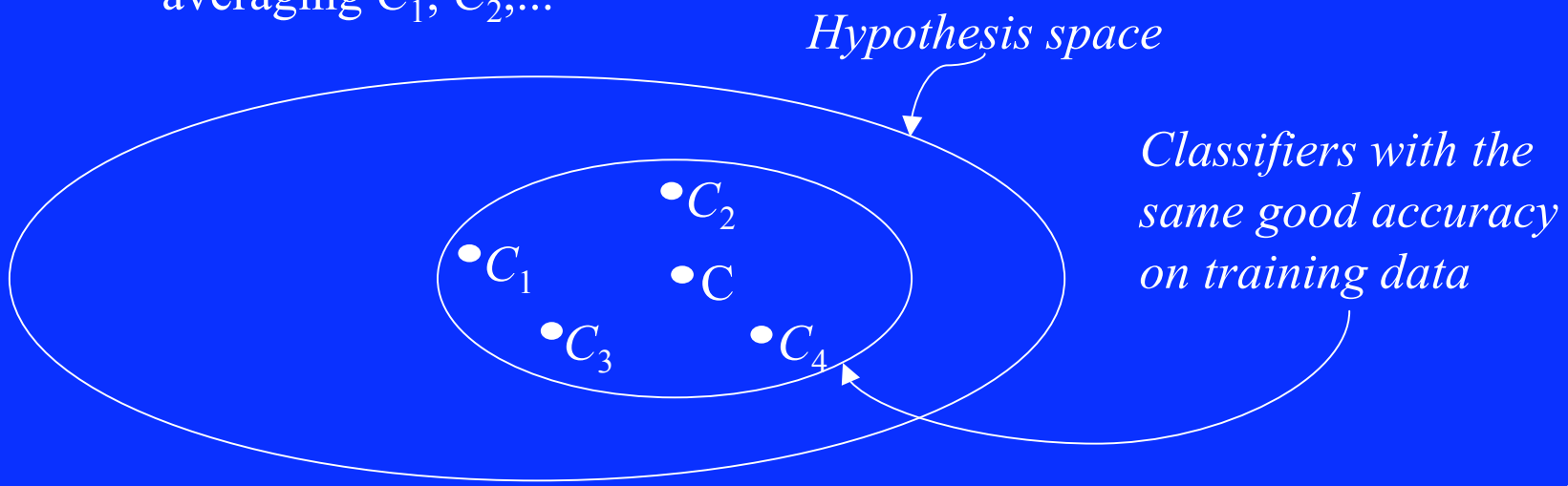
Multiple Classifier Fusion: *Worst Case* Motivation

- In the small sample size case, it is pretty intuitive that I can avoid selection of the worst classifier by, for example, averaging over the individual classifiers

A paradigmatic example (Tom Dietterich, MCS 2000 Workshop)

Few training data with respect to the size of the hypothesis space

- several classifiers (C_1,C_2,...) can provide the same accuracy on validation data
- a good approximation of the optimal classifier C can be found by averaging C_1, C_2,...



A practical example

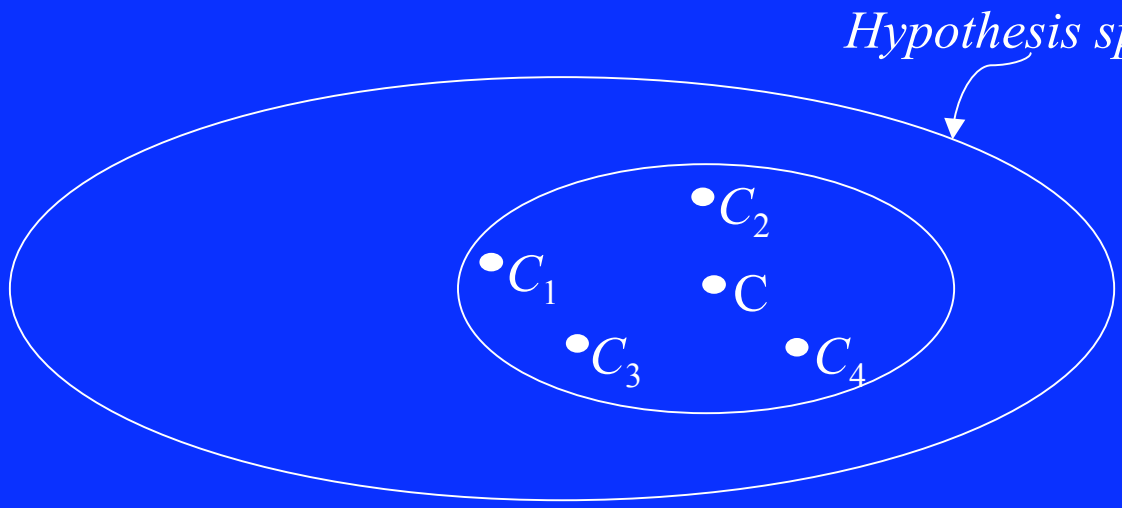
Face recognition using PCA and LDA algorithms (Yale data base)

For different choices of the training set (different "trials"), the best classifier varies. Fusion by averaging avoids to select the worst classifier for some test cases (Marcialis and Roli, *Int. Journal of Image and Graphics,* 2006).

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
PCA	76,7%	87,8%	**92,2%**	84,4%	**88,9%**
LDA	**83,3%**	**90,0%**	85,6%	84,4%	86,7%
Fusion by Average	80,0%	92,2%	88,9%	86,7%	88,9%

Theoretical support for the *worst case* motivation

Tom Dietterich's claim (2000)



Classifiers with the same good accuracy on training data

In 2005, Fumera and Roli confirmed theoretically the claim of Tom Dietterich.

They proved that averaging of classifiers outputs guarantees a better test set performance than the worst classifier of the ensemble (IEEE-T on PAMI, June 2005).

Multiple Classifier Fusion: *Best Case* Motivation

- Beside avoiding the selection of the worst classifier, under particular hypotheses, fusion of multiple classifiers can improve the performance of the best individual classifiers and, in some special cases, provide the optimal Bayes classifier

•This is possible if individual classifiers make "different" errors.

•*Luckily, we have many experimental evidences about that ! !*

•Theoretical support for some classes of fusers (e.g., linear combiners, majority voting)

•For linear combiners, Tumer and Ghosh (1996) showed that averaging the outputs of individual classifiers with unbiased and uncorrelated errors can improve the performance of the best individual classifier and, for an infinite number of classifiers, provides the optimal Bayes classifier

Experimental evidences: Multimodal Biometrics
(Roli et al., Information Fusion Conf., 2002)

- XM2VTS database
  - face images, video sequences, speech recordings
  - 200 training and 25 test clients, 70 test impostors

•Eight classifiers based on different techniques: two speech classifiers, six face classifiers

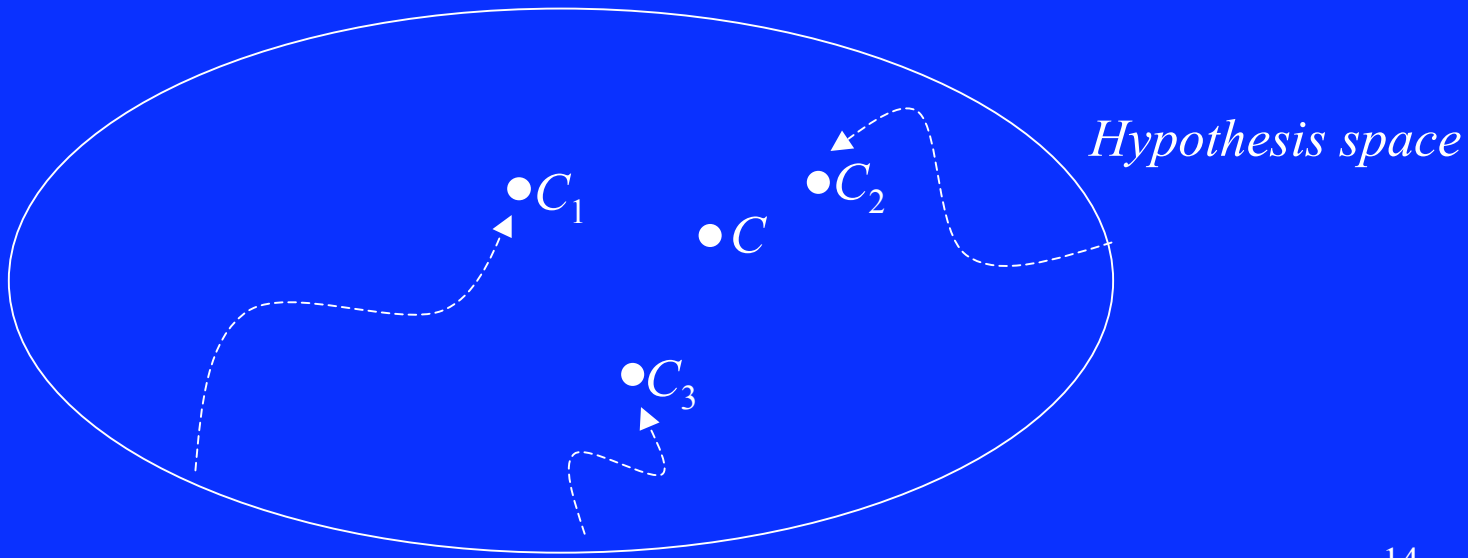
•Simple averaging allows avoiding the selection of the worst classifier for some test cases and, in some experiments, outperformed the best individual classifier

Fusion of multiple classifiers: computational motivation (T.Dietterich, 2000)

Many learning algorithms suffer from the problem of local minima

- Neural Networks, Decision Trees (optimal training is NP-hard!)
- Finding the best classifier *C* can be difficult even with enough training data

➢Fusion of multiple classifiers constructed by running the training algorithm from different starting points can better approximate *C*



Further Motivations for Multiple Classifiers

- In ***sensor fusion***, multiple classifiers are naturally motivated by the application requirements

- The ***"curse" of*** pattern classifier ***designer***

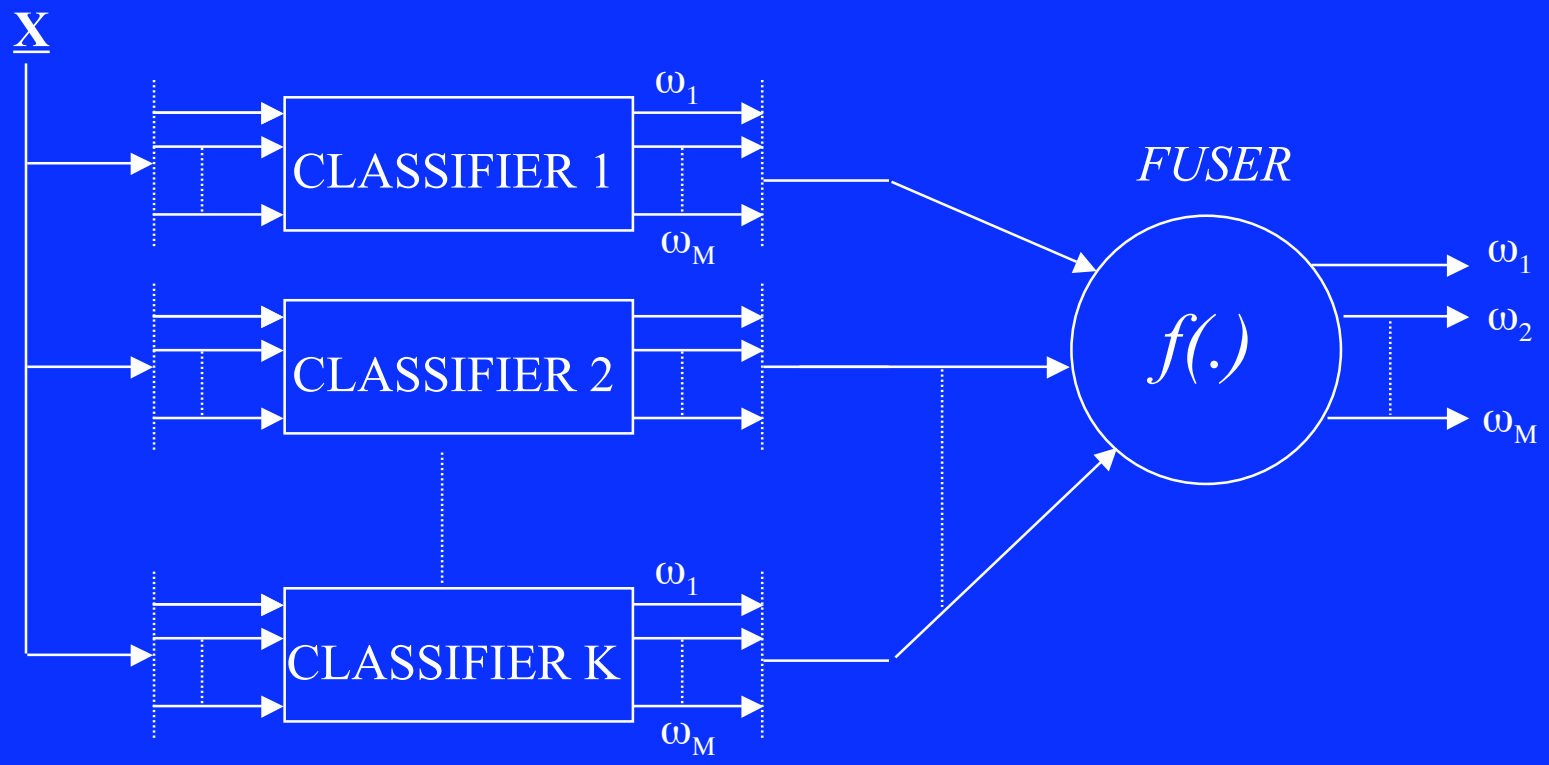
  - The need of avoiding having to make a meaningful choice of some arbitrary initial condition, such as the initial weights for a neural network

  - The intrinsic difficulty of choosing appropriate design parameters

  - "Saturation" of classifier's design *(F. Roli, Pattern Rec. Letters, 2000)*

- **Monolithic** vs. **Modular** classifier systems: different classifiers can have different domains of competence

Basic Architecture of a Multiple Classifier System



Basically, Multiple Classifier System (MCS) consists of an ensemble of different classification algorithms and a "function" f(.) to "fuse" classifiers outputs. The parallel architecture is very natural !

MCS: Basic Concepts

MCS can be characterized by:

- The **Architecture/Topology**
  - The classifier **Ensemble**: type and number of ***base*** classifiers. The ensemble can be subdivided into subsets in the case of non parallel architectures
  - The **Fuser**

MCS Architectures/Topologies

•**Parallel** topology: multiple classifiers operate in parallel. A single combination function merges the outputs of the individual classifiers

•**Serial/Conditional** topology

-Classifiers are applied in succession, with each classifier producing a reduced set of possible classes

-A primary classifier can be used. When it rejects a pattern, a secondary classifier is used, and so on

•**Hybrid** topologies

The Ensemble

•The most common type of MCS, widely used and investigated, includes an ensemble of classifiers, named "base" classifiers, and a function for parallel combination of classifier outputs

•The base classifiers are often algorithms of the same type (e.g., decision trees or neural networks), and statistical classifiers are the most common choice.

•The use of hybrid ensembles containing different types of algorithms has been investigated much less, as well as ensembles of structural, graph-based, classifiers have not attracted much attention, although they could be important for some real applications.

Fuser ("combination" rule)

Two main categories of fuser:

Integration (fusion) functions: for each pattern, all the classifiers contribute to the final decision. Integration assumes **competitive** classifiers

Selection functions: for each pattern, just one classifier, or a subset, is responsible for the final decision. Selection assumes **complementary** classifiers

- Integration and Selection can be "merged" for designing hybrid fuser
- *Multiple* functions for non parallel architecture can be necessary

Focus on Parallel Architecture

- So far research on MCS focused on parallel architectures
- Accordingly, general methodologies and clear foundations are mostly available for parallel architectures
- MCSs based on other architectures (serial, hierarchical, hybrid, etc) were highly specific to the particular application
- In the following, we focus on parallel architectures and briefly discuss the relation between classifier ensemble and combination function. Many of the concepts we discuss also hold for different architectures

Classifiers "Diversity" vs. Fuser Complexity

- *Fusion is obviously useful only if the combined classifiers are not the same classifier…*
  - Intuition: classifiers with high accuracy and high "diversity"

➢ The required degree of error *diversity* depends on the fuser complexity

- Majority vote fuser: the majority should be always correct
- Ideal selector ("oracle"): only one classifier should be correct for each pattern

An example, four diversity levels (A. Sharkey, 1999)

Level 1: no more than one classifier is wrong for each pattern
Level 2: the majority is always correct
Level 3: at least one classifier is correct for each pattern
Level 4: all classifiers are wrong for some patterns

Classifiers Diversity Measures: An Example

- Various measures (classifier outputs correlation, Partridge's diversity measures, Giacinto and Roli compound diversity, etc.) can be used to assess how similar two classifier are.

L. Kuncheva (2000) proposed the use of Q statistics:

$$Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}$$

Q varies between –1 and 1. Classifiers that tend to classify the same patterns correctly will have values of Q close to 1, and those which commit errors on different patterns will render Q negative

Classifiers' diversity is an elusive concept..

[Kuncheva, 03]

- Measures of diversity in classifier ensembles are a matter of on-going research (L.I. Kuncheva book, 2005)

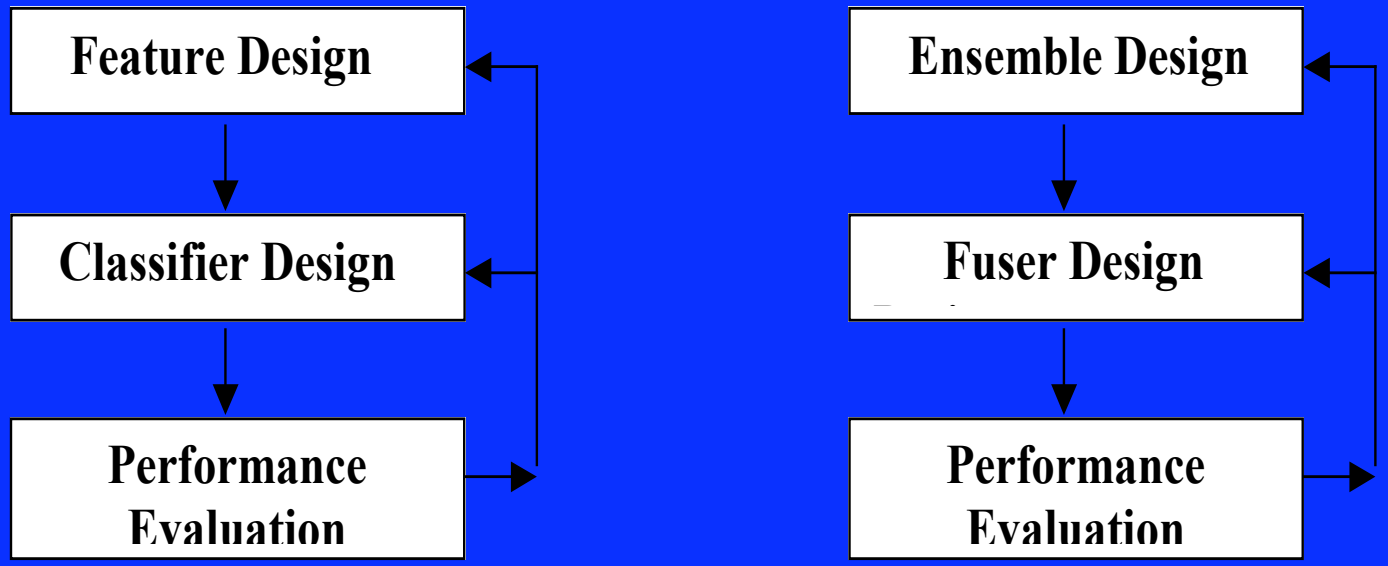
- Key issue: how are the diversity measures related to the accuracy of the ensemble ?

  - Simple fusers can be used for classifiers that exhibit a simple complementary pattern (e.g., majority voting)

  - Complex fusers, for example, a dynamic selector, are necessary for classifiers with a complex dependency model

  - *The required "complexity" of the fuser depends on the degree of classifiers diversity*

Analogy between MCS and Single Classifier Design



Design cycles of single classifier and MCS (Roli and Giacinto, 2002)

Two main methods for MCS design (T.K. Ho, 2000):

- Coverage optimization methods
- Decision optimization methods

MCS Design

- The design of MCS involves two main phases: the design of the classifier ensemble, and the design of the fuser
- The design of the classifier ensemble is aimed to create a set of "complementary/diverse" classifiers
- The design of the combination function/fuser is aimed to create a fusion mechanism that can exploit the complementarity/diversity of classifiers and optimally combine them
- The two above design phases are obviously linked (Roli and Giacinto, Design methods for MCS, Book Chapter, 2002)
- In the following (Parts II and III), we illustrate the main methods for constructing and fusing multiple classifiers

A small homework…

Using any tool in your hands (e.g., Google), do a search and let me know which is the oldest paper dealing with the topic of "multiple classifier systems" that you are able to find, and explain me shortly how you did your search.

Send me via mail (roli@diee.unica.it) the result of your search. Thanks!

Question…

Using independent classifiers is always better than using dependent classifiers for combination purposes?

Send me via mail (roli@diee.unica.it) your justified answer