Classifier Combination

Kuncheva Ch. 3

Combination: The Statistical Reason

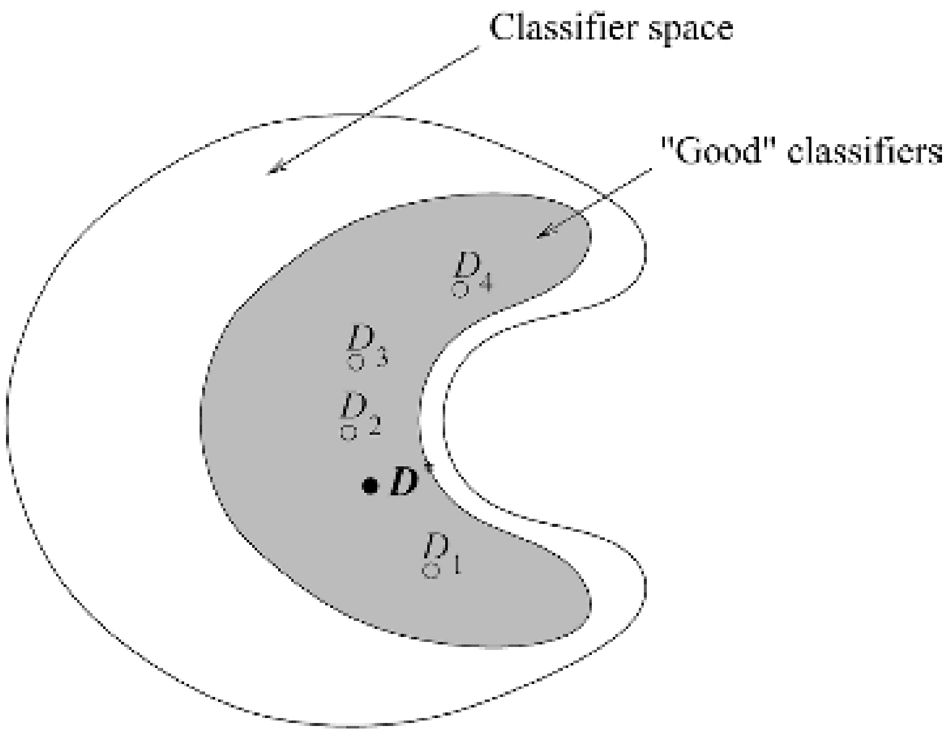

Fig. 3.1 *The statistical reason for combining classifiers. D^* is the best classifier for the problem, the outer curve shows the space of all classifiers; the shaded area is the space of classifiers with good performances on the data set.*

R·I·T

Combination: The Computational Reason

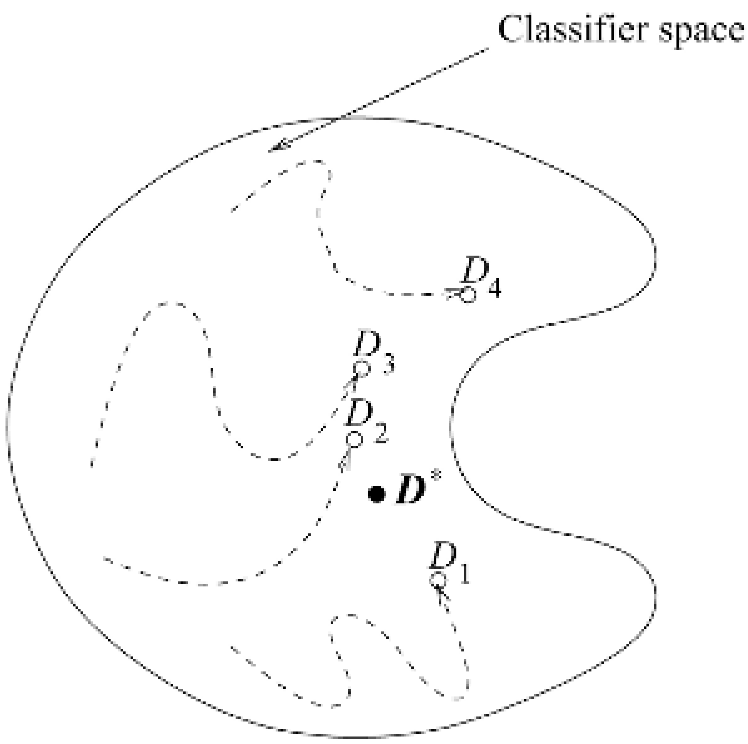

Fig. 3.2 *The computational reason for combining classifiers. D^* is the best classifier for the problem, the closed space shows the space of all classifiers, the dashed lines are the hypothetical trajectories for the classifiers during training.*

R·I·T

Combination: The Representational Reason

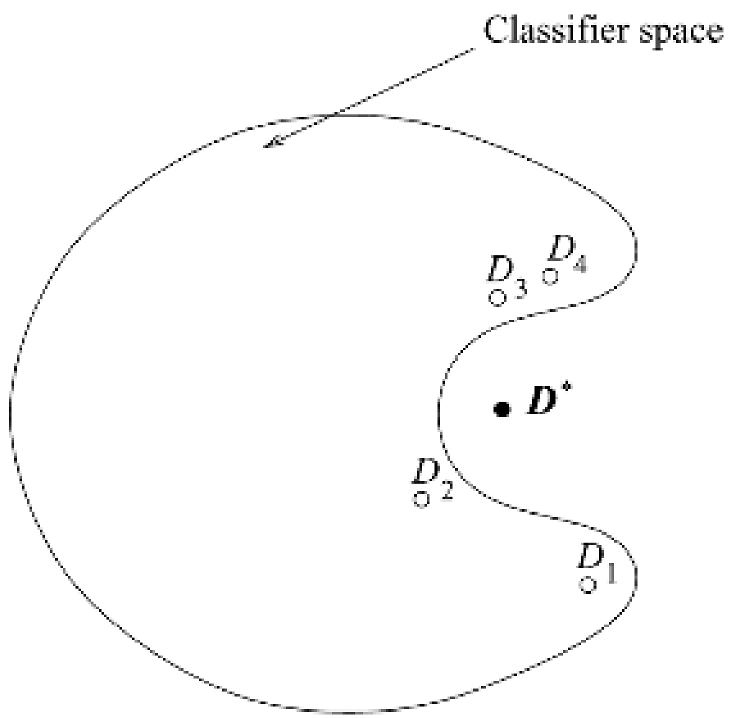

Fig. 3.3 *The representational reason for combining classifiers. D^* is the best classifier for the problem; the closed shape shows the chosen space of classifiers.*

Classifier Ensembles: Types of Combination

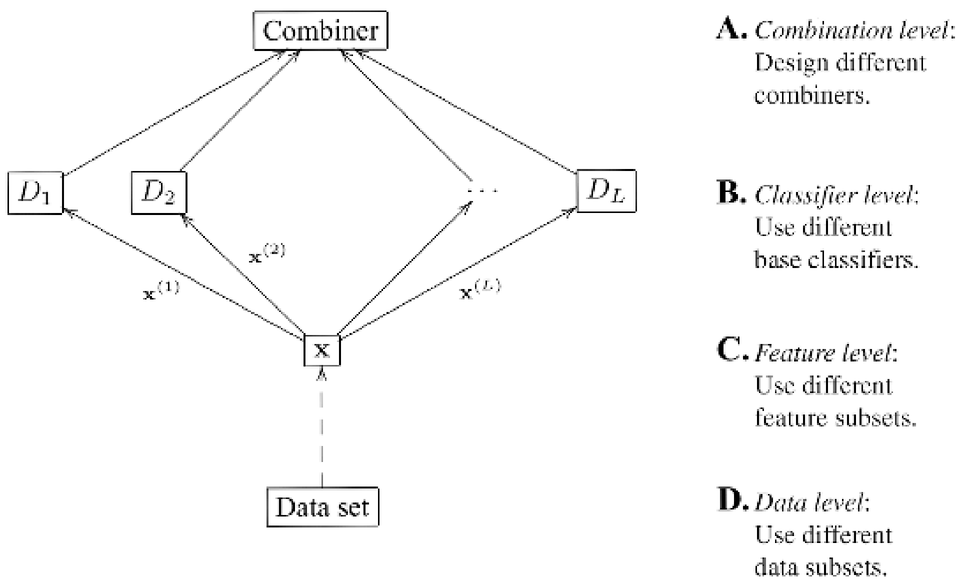

Fig. 3.4 *Approaches to building classifier ensembles.*

Training: Stacked Generalization

	A	B	C	D
Training	B	C	D	A
	C	D	A	B
Testing	D	A	B	C

Fig. 3.5 *Standard four-fold cross-validation set-up.*

Protocol:

Train base classifiers using cross-fold validation

Then train combiner on all N points by using the class labels output by the base classifiers for each fold (train/test partition)

Other Issues

Fusion vs. Selection

Trainable vs. Non-trainable Combiners

Decision Optimization vs. Coverage Optimization

- Tuning a fixed set of base classifiers *vs.* creating diverse base classifiers using a fixed combiner

R·I·T

Fusion of Class Labels

Kuncheva, Ch. 4

Let's Start at the End...

4.8 CONCLUSIONS

This chapter describes various methods for combining class label outputs. Formally speaking, we transformed the problem to find a mapping

$$\boxed{D : \Re^n \to \Omega} \tag{4.82}$$

into a problem of finding a mapping

$$\boxed{\mathcal{F} : \Omega^L \to \Omega} \tag{4.83}$$

We can call Ω^L an *intermediate feature space*, which in this case is discrete. The methods described here varied on the assumptions and also on the complexity of their implementation. Table 4.11 summarizes the methods in this chapter.

R·I·T

Methods for Fusing Class (Label) Outputs

TABLE 4.11 A Summary of the Label Fusion Methods.

Method	Assumptions (for Optimality)	Memory Requirements (in Number of Parameters Needed for Its Operation)
Majority vote	None	None
Weighted majority vote	None	L
Naive Bayes	Conditional independence	$L \times c^2$
Behavior knowledge space (BKS)	None	c^L
Wernecke	None	c^L
First-order dependence tree	Conditional first-order dependence	$c \times [(L-1) \times c^2 + L]$
SVD combination	None	$[L \times c + (c+1)]\delta$ $(\delta \leq \min\{N, (L \cdot c)\})$

SVD, singular value decomposition.

R·I·T

Classifier Output Types
(Xu, Krzyzak, Suen; ad. by Kuncheva)

Type 0: Oracle Level (Kuncheva)

Outputs: 0/1 (incorrect/correct)

Type 1: Abstract Level

Outputs: Chosen class label

Type 2: Rank Level

List of ranked class labels

Type 3: Measurement Level

Output: (Usually) values in [0,1] for each .class (discriminant function outputs)

Simple Consensus Models

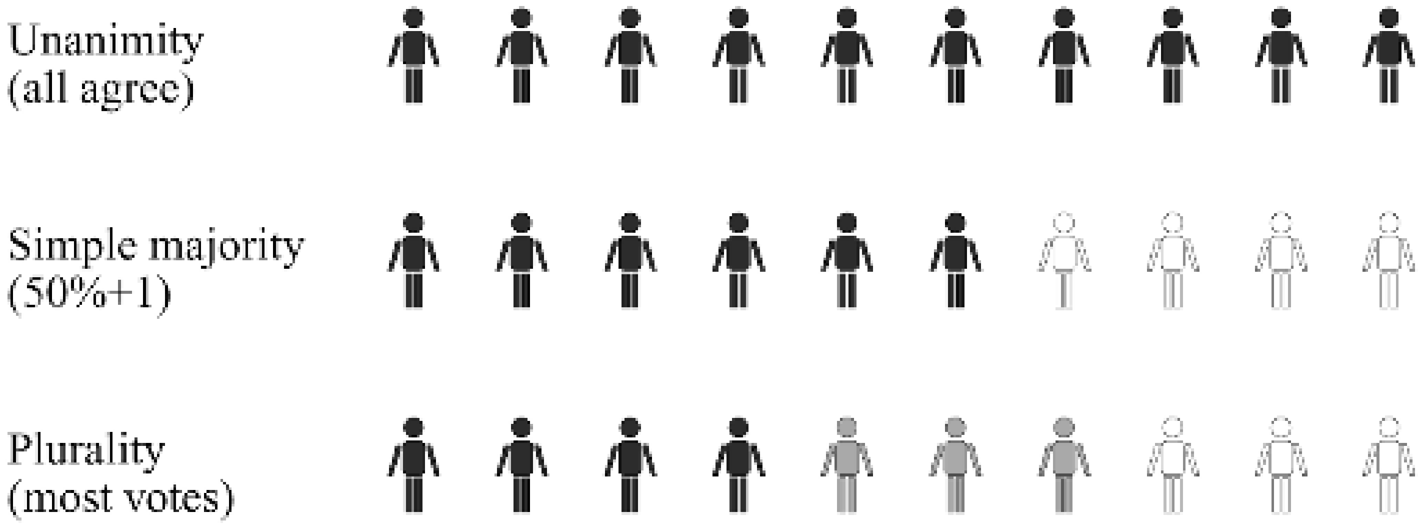

Fig. 4.1 *Consensus patterns in a group of 10 decision makers: unanimity, simple majority, and plurality. In all three cases the final decision of the group is "black."*

R·I·T