The Combining Classifier: to Train or Not to Train?

Robert P.W.Duin

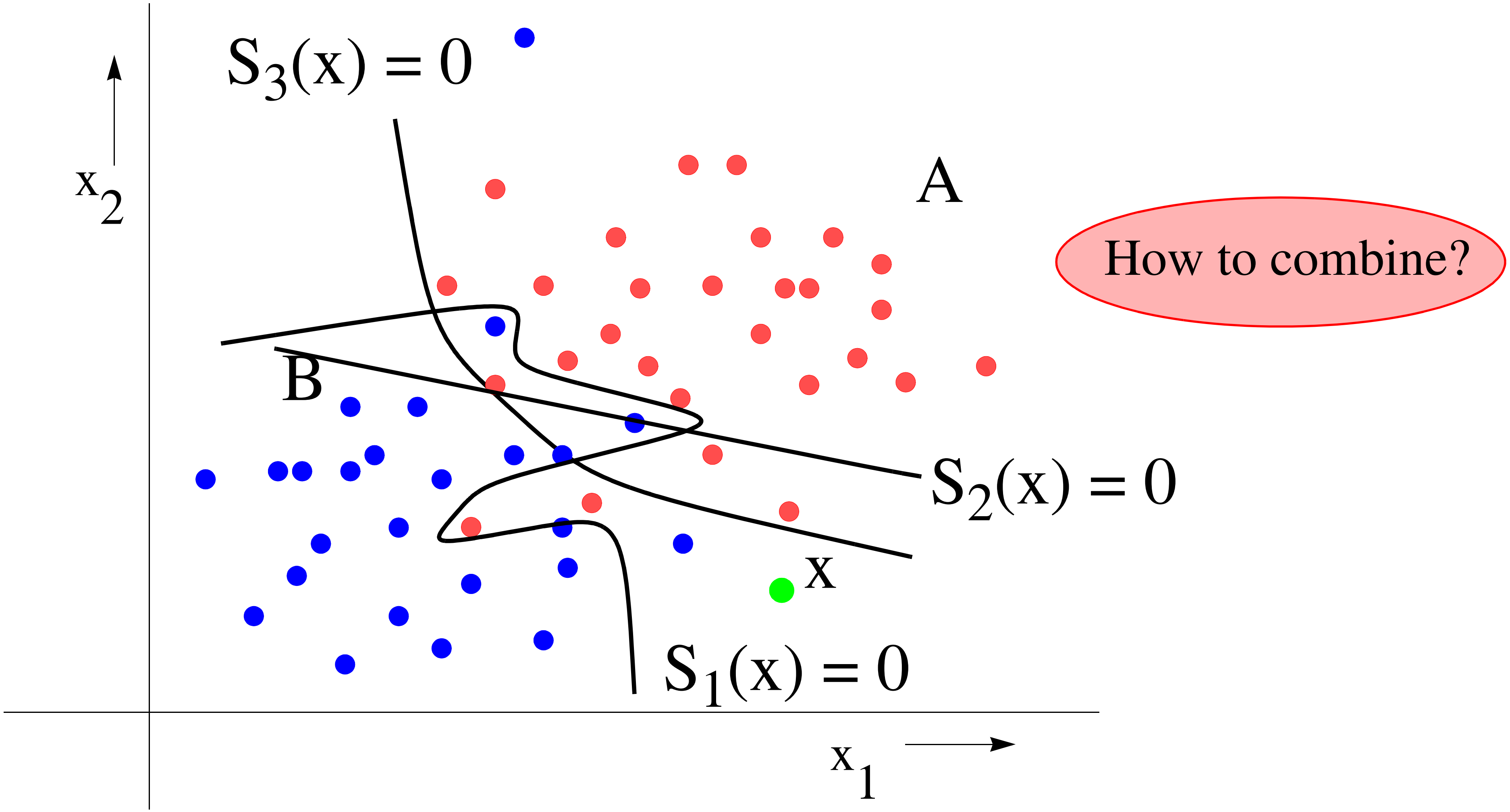
Pattern Recognition Group, Faculty of Applied Sciences

Delft University of Technology, The Netherlands

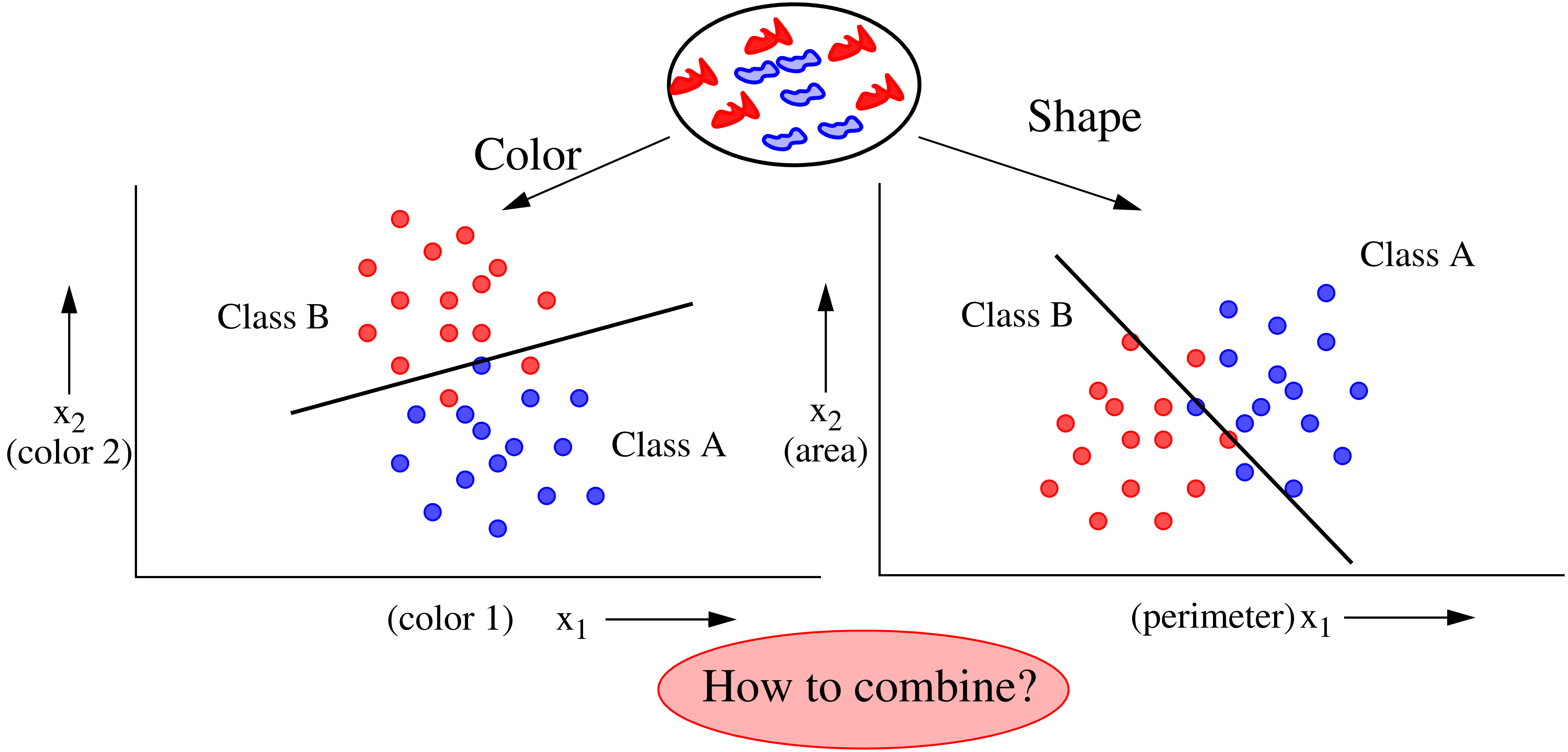
Quebec City, August 2002

P.O. Box 5046, 2600GA Delft, The Netherlands.
Phone: +(31) 15 2786143, FAX: +(31) 15 2786740,
E-mail: duin@tnw.tudelft.nl

Several Classifiers in Same Feature Space



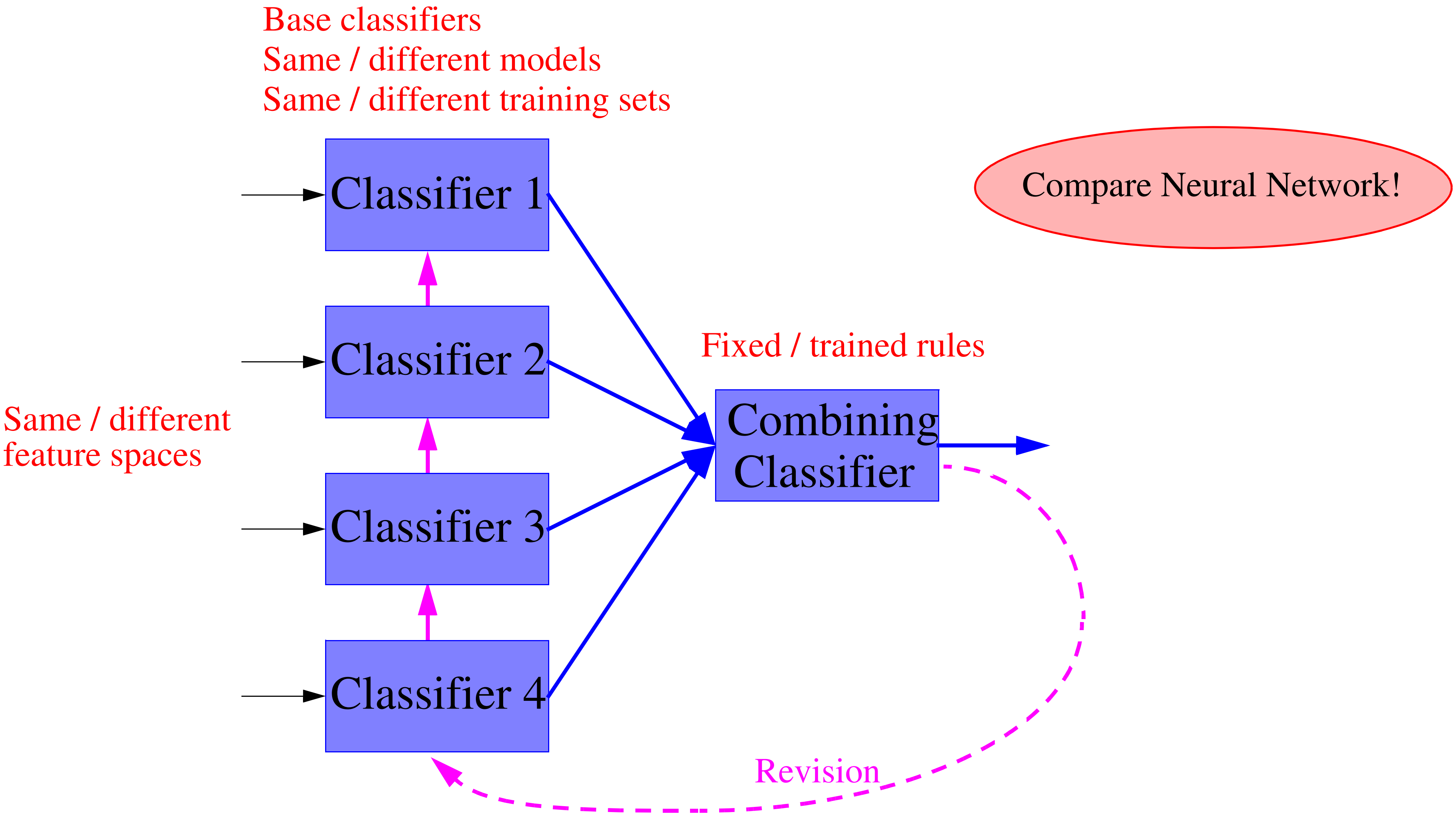
Several Classifiers in Different Feature Spaces



Multiple Classifier Sources

Different feature spaces:	Face, Voice, Fingerprint
Different training sets:	Sampling, Bootstrapping
Different classifiers:	k_NN, Bayes Normal, Dec. Tree, SVC, Neural Net
Different architectures:	Neural Net: #Layers, #Units, Transfer function.
Different parameter values:	k in k_NN, kernel in SVC, pruning in Dec. Tree
Different initializations:	Neural Net

Combining Classifiers Architecture



Strategic Reasons for Multiple Classifiers

Multiple Sensors :	Different feature spaces
No Clear Data Model:	Multiple Classifiers
Unstable Classification:	Multiple Bootstrapped Training Sets
Classifier Economy:	Multiple Initializations

Combining Classifiers Examples

Multi-layer perceptrons by voting (majority or veto), (Nilsson, 1965)

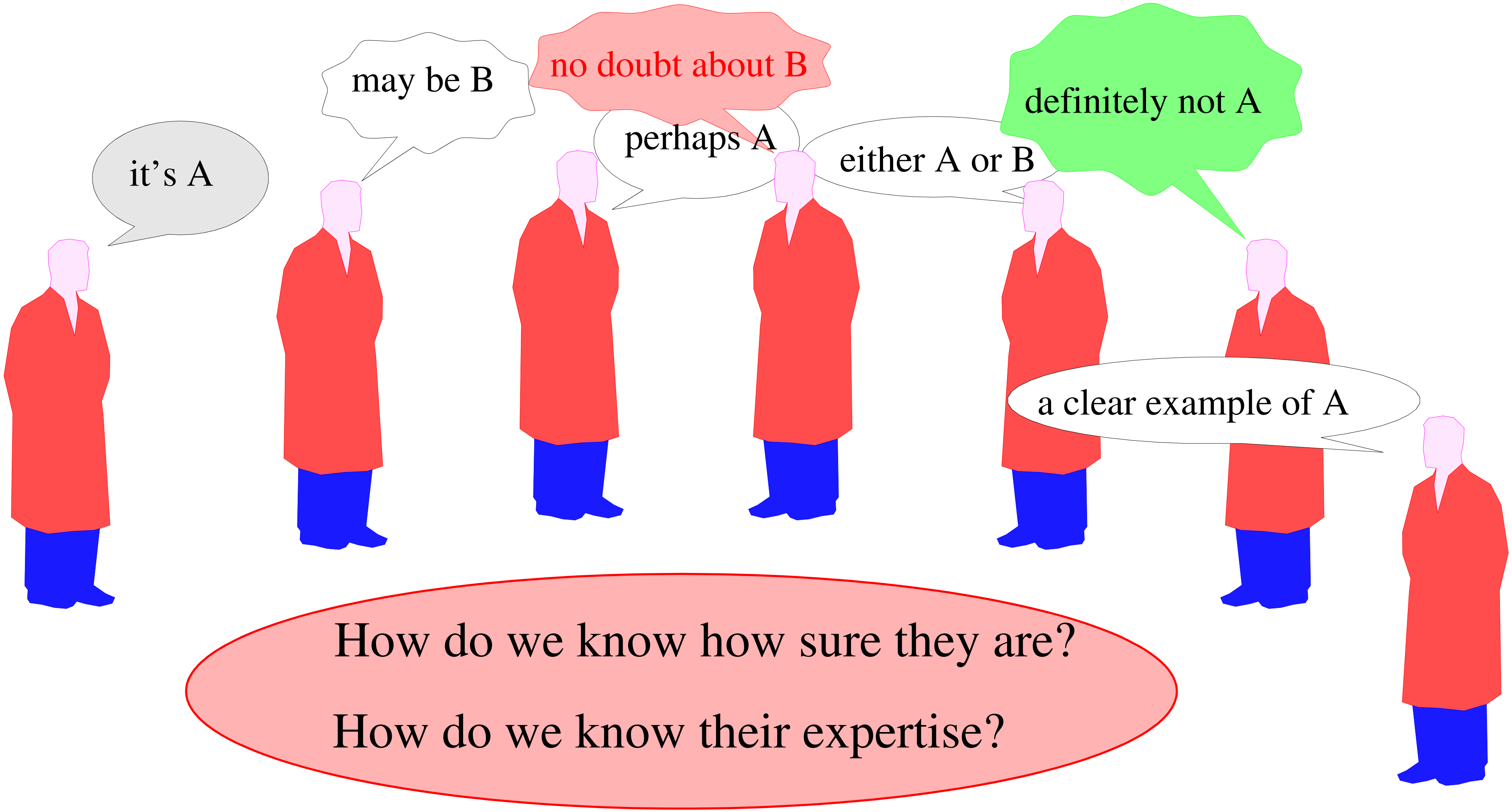
Multi class classification by multiple 2-class discriminants

Combining fingerprint classifiers, (Prabhakar & Jain, PR, 1999, 2002)

Combining OCR feature spaces, (Mao PRL-1997, Duin, MCS-2000)

Combining biometric sensors (Jain PAMI 1998, Kittler PAMI-1998)

Experts



How do we know how sure they are?

How do we know their expertise?

Two Opposite Multiple Classifier Strategies

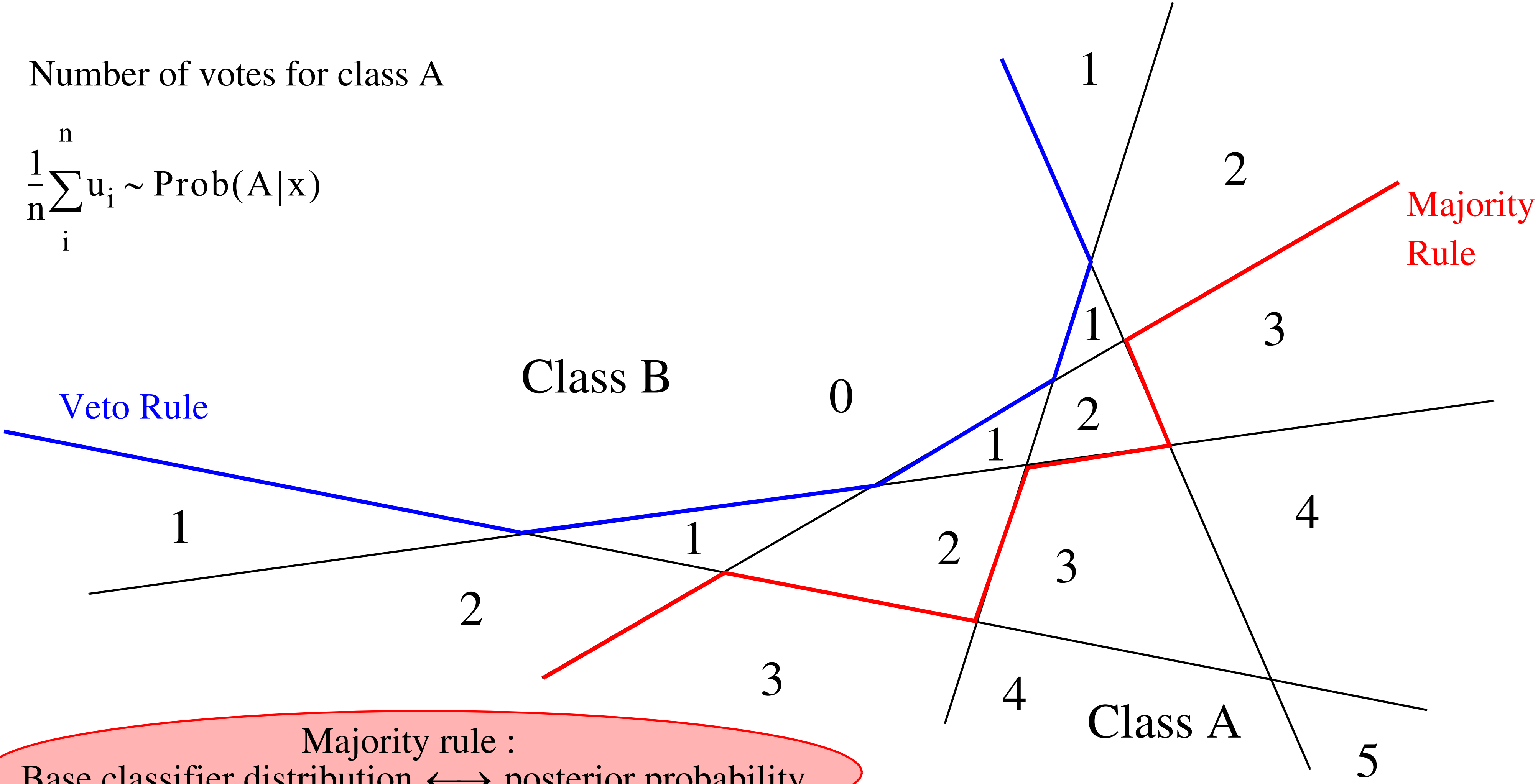
How to generate base classifiers, given how to combine them
(e.g. bagging, random subspace method)

Given base classifiers; how to combine them ?

Voting

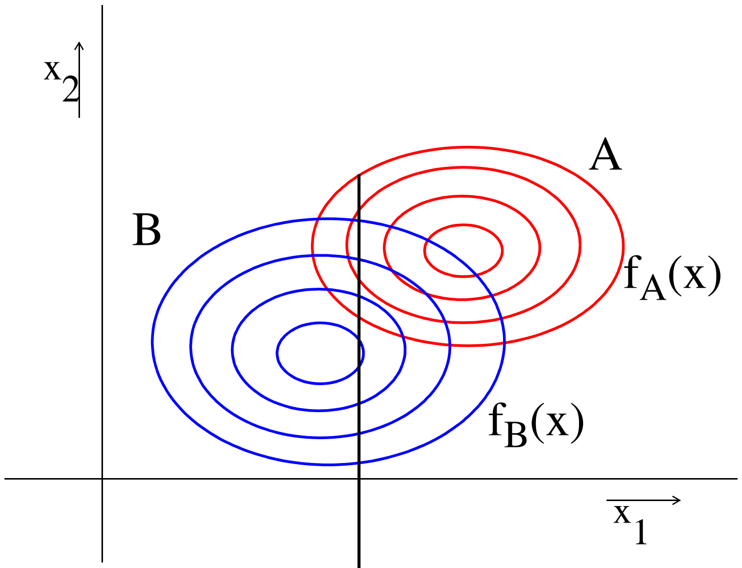
Number of votes for class A

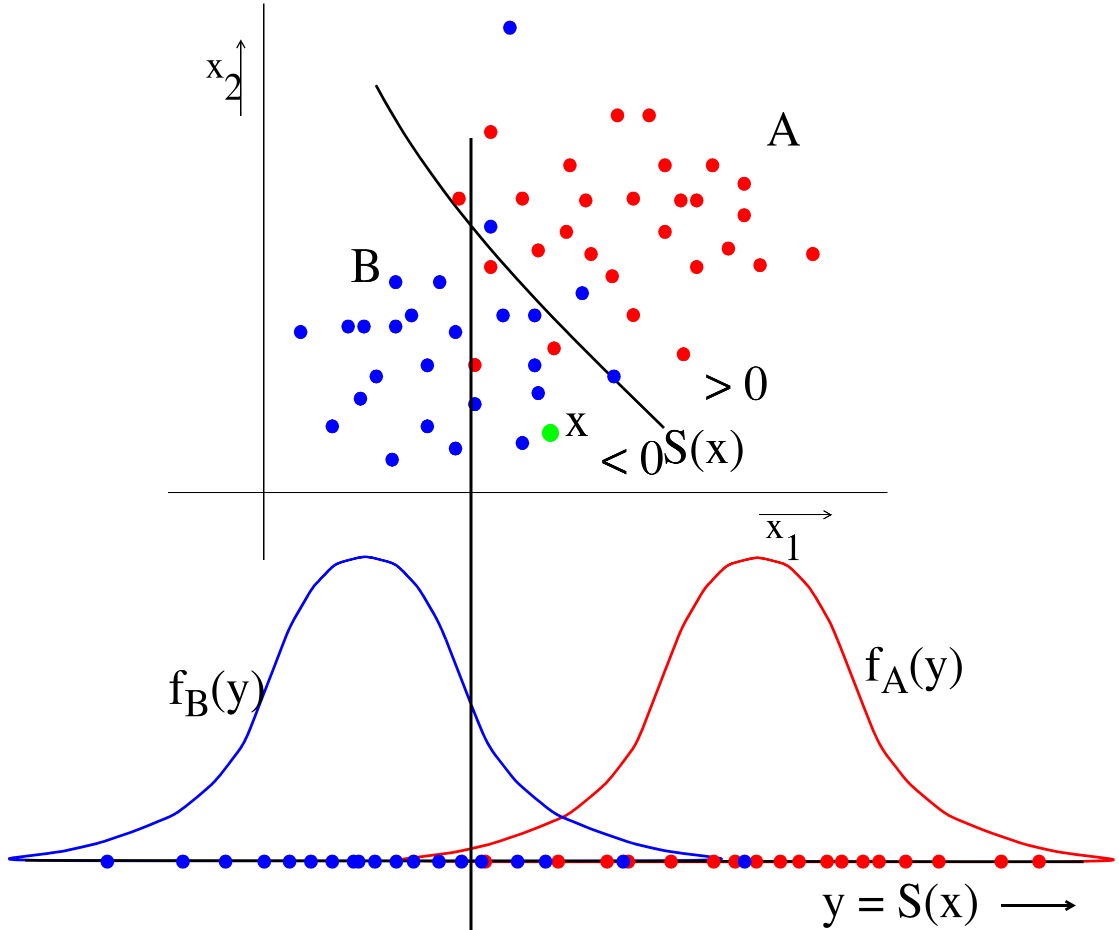
$$\frac{1}{n}\sum_{i}^{n} u_i \sim \mathrm{Prob}(A|x)$$



Majority rule :
Base classifier distribution $\longleftrightarrow$ posterior probability

Posterior Probabilities



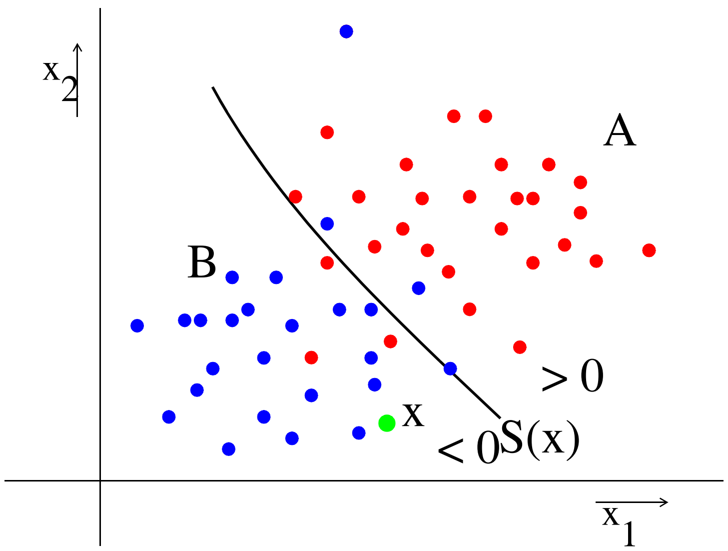


Posterior Probabilities

Classifier Conditional Posterior Probabilities

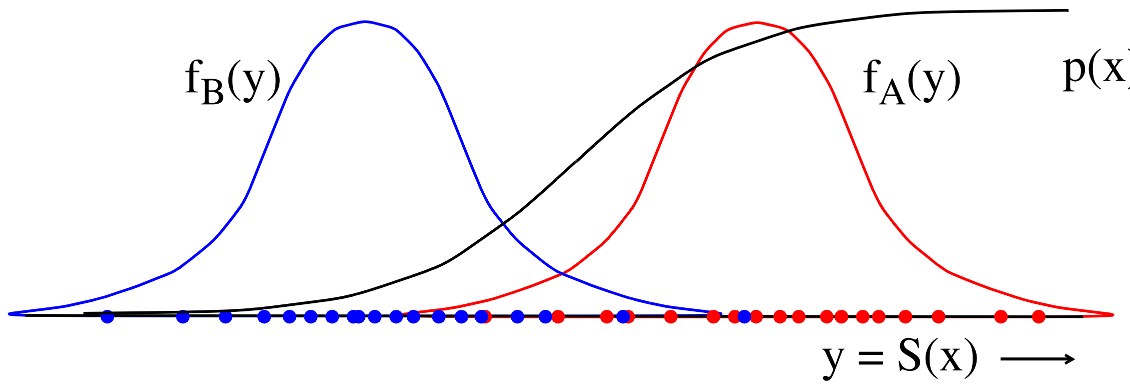
$$\text{Prob}\,(\,x \in A \mid x\,) = \frac{p_A f_A(x)}{f(x)} \qquad \neq \qquad p(x) = \text{Prob}\,(\,x \in A \mid y = S(x)\,) = \frac{p_A f_A(y)}{f(y)}$$

Posterior Probabilities for Arbitrary Classifiers: Normalization



Classifier Conditional Posterior Probabilities

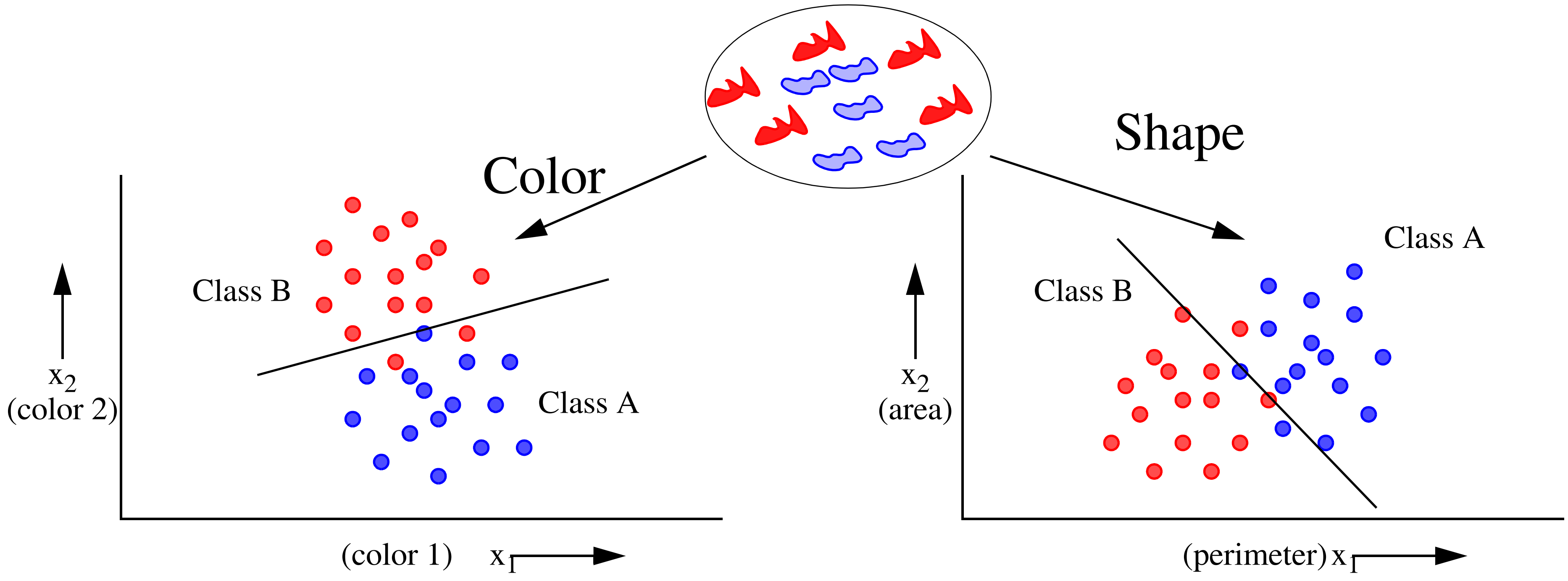
$$p(x) = \text{Prob}\,(\,x \in A \mid y = S(x)\,) = \frac{p_A f_A(y)}{f(y)}$$



Fit a sigmoid, or a logistic function to the data y = S(x),

such that $\prod_i p(x_i)$ is maximized restricted to p(x) = 0.5 for S(x) = 0.

Combining Different Representations - Different Areas of Expertise



Base classifier j posterior probabilities for class A : $y_{Aj} = \text{Prob}_j(A|x_j)$

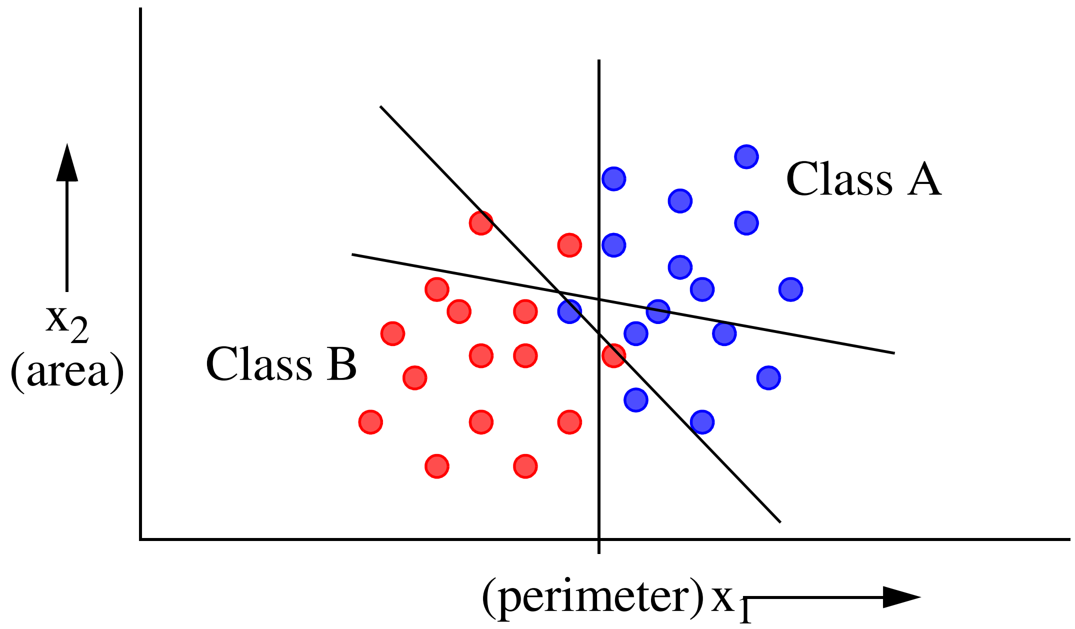
Product Rule: $y_A = \prod \text{Prob}_j(A|x_j)$, $y_B = \prod \text{Prob}_j(B|x_j)$,

Useful for 'independent' feature spaces (logical 'AND', experts should agree)

Minimum Rule: $y_A = \text{Min}\{\text{Prob}_j(A|x_j)\}$, $y_B = \text{Min}\{\text{Prob}_j(B|x_j)\}$

Assign according to 'least objecting expert'

Combining Different Estimates - Differently Trained Experts



Base classifier j posterior probabilities for class A : $y_{Aj} = \mathrm{Prob}_j(A|\mathbf{x})$

Sum (Mean) Rule: $y_A = \sum \mathrm{Prob}_j(A|\mathbf{x})$, $y_B = \sum \mathrm{Prob}_j(B|\mathbf{x})$,

Useful for improved estimates of posterior probabilities

Also: Median and Majority Voting

Improvement by averaging out mistakes of experts

The Product and the Minimum Rule

Base classifier j posterior probabilities for class A : $y_{Aj} = \text{Prob}_j(A|x_j)$

Product Rule: $y_A = \prod \text{Prob}_j(A|x_j)$, $y_B = \prod \text{Prob}_j(B|x_j)$,

Useful for 'independent' feature spaces, see *Kittler, IEEE-PAMI-20(3),1998*

Minimum Rule: $y_A = \text{Min}\{\text{Prob}_j(A|x_j)\}$, $y_B = \text{Min}\{\text{Prob}_j(B|x_j)\}$

Assign according to 'least objecting classifier'

objects	Classifier 1		Classifier 2		Product		Minimum	
	Class A	Class B	Class A	Class B	Class A	Class B	Class A	Class B
1	0.4	0.6	0.2	0.8	0.08	0.48	0.2	0.6
2	0.1	0.9	0.7	0.3	0.07	0.27	0.1	0.3
3	0.3	0.7	0.4	0.6	0.12	0.42	0.3	0.6
4	0.5	0.5	0.2	0.8	0.10	0.40	0.2	0.5
5	0.0	1	0.9	0.1	0.00	0.10	0.0	0.1
6	0.8	0.2	0.2	0.8	0.16	0.16	0.2	0.2

Fixed combining rules

Product, Minimum

Independent feature spaces

Different areas of expertise

Error free posterior probability estimates

Ever optimal?

Sum (Mean), Median, Majority Vote

Equal posterior-estimation distributions in same feature space

Differently trained classifiers, but drawn from the same distribution

Bad if some classifiers (experts) are very good or very bad

Maximum

Trust the most confident classifier / expert

Bad if some classifiers (experts) are badly trained

Fixed combining rules are sub-optimal

Base classifiers are never really independent (product)

Base classifiers are never really equally imperfectly trained (sum, median, majority)

Sensitivity to over-confident base classifiers (product, min, max)

Fixed combining rules are never optimal

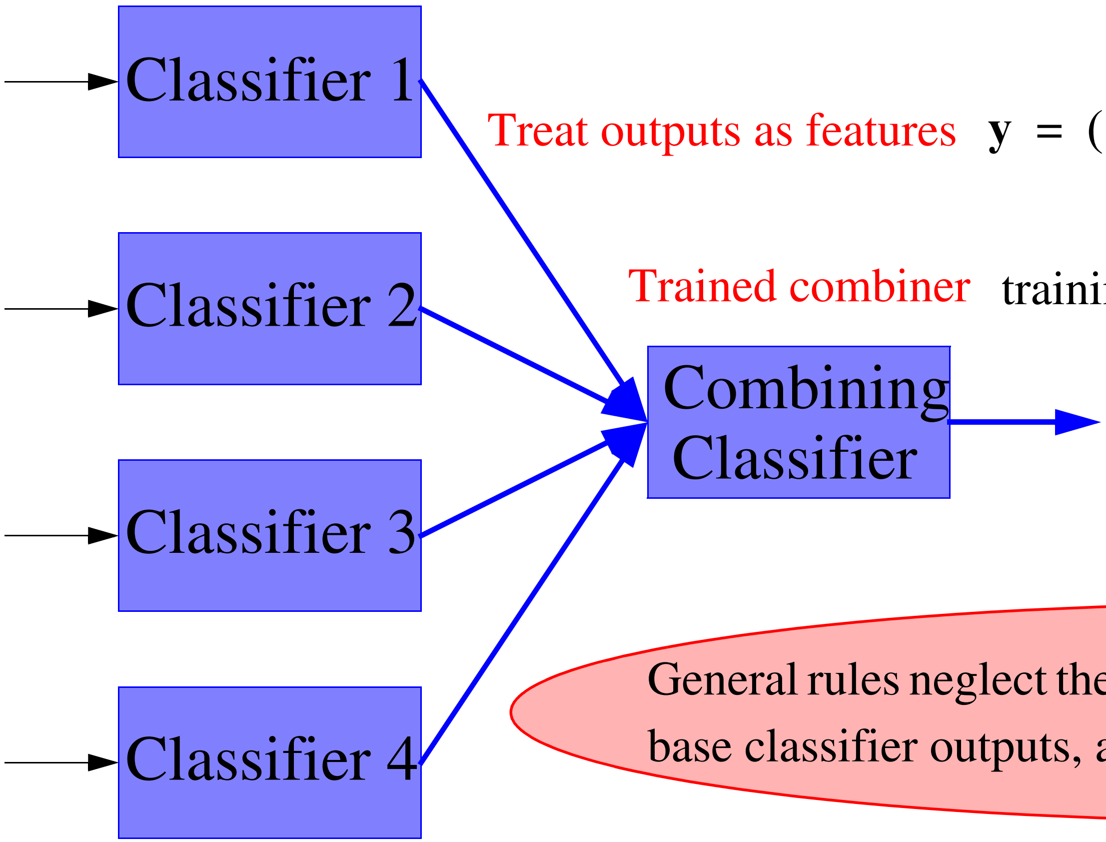
Larger training sets do not really improve this (except max?)

Trained Combining Classifier

Base classifiers $y_{Aj} = \mathrm{Prob}_j(A|x)$, $y_{Bj} = \mathrm{Prob}_j(B|x)$

Treat outputs as features $\mathbf{y} = (y_{A1}, y_{A2}, y_{A3}, y_{A4}, y_{B1}, y_{B2}, y_{B3}, y_{B4})$

Trained combiner training set: $\{\mathbf{y}_1, \mathbf{y}_2, \ldots\ldots, \mathbf{y}_m\}$

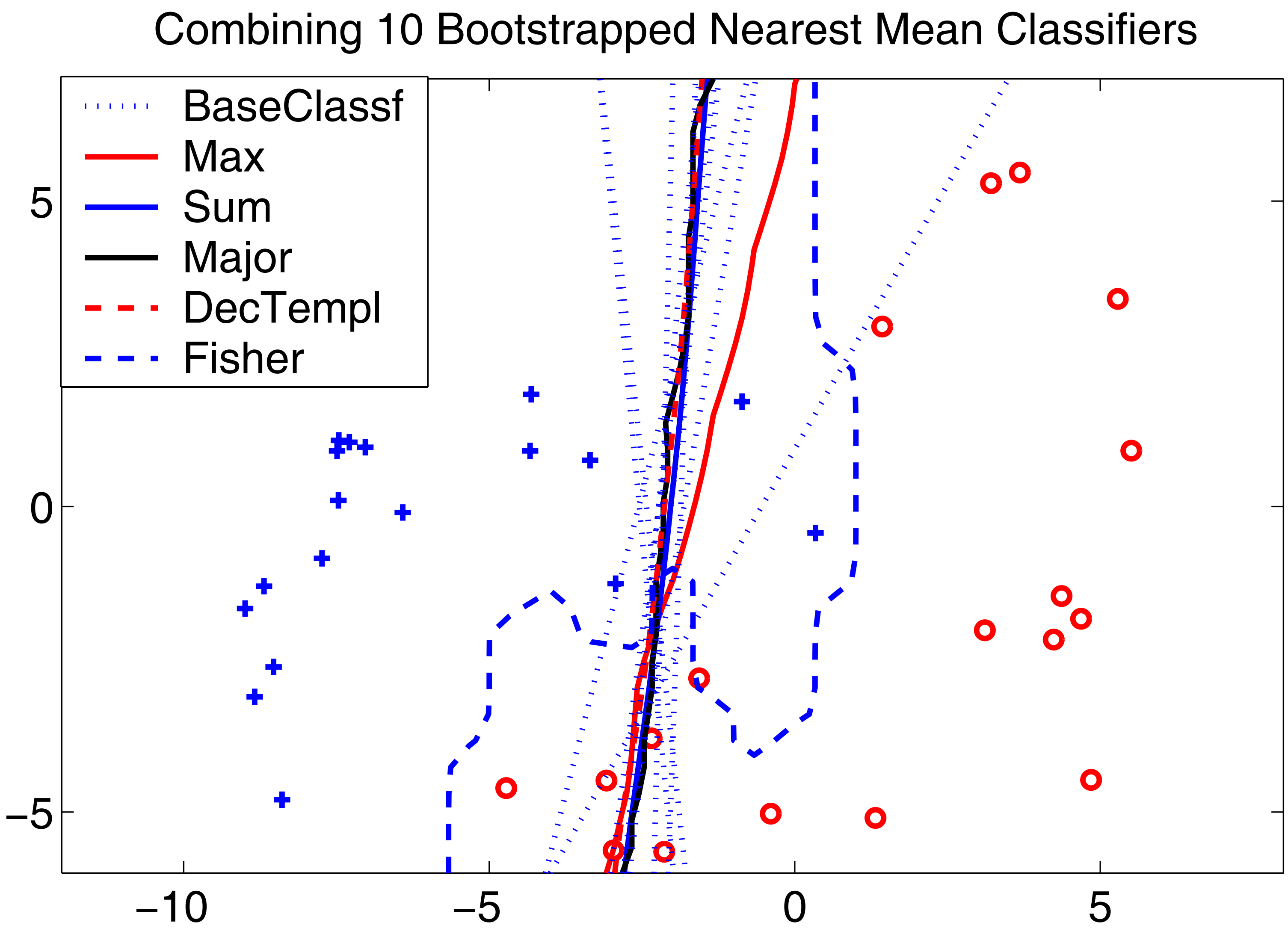


- Decision Templates*
- Fisher
- Bayes Normal
- Decision Tree
- ....

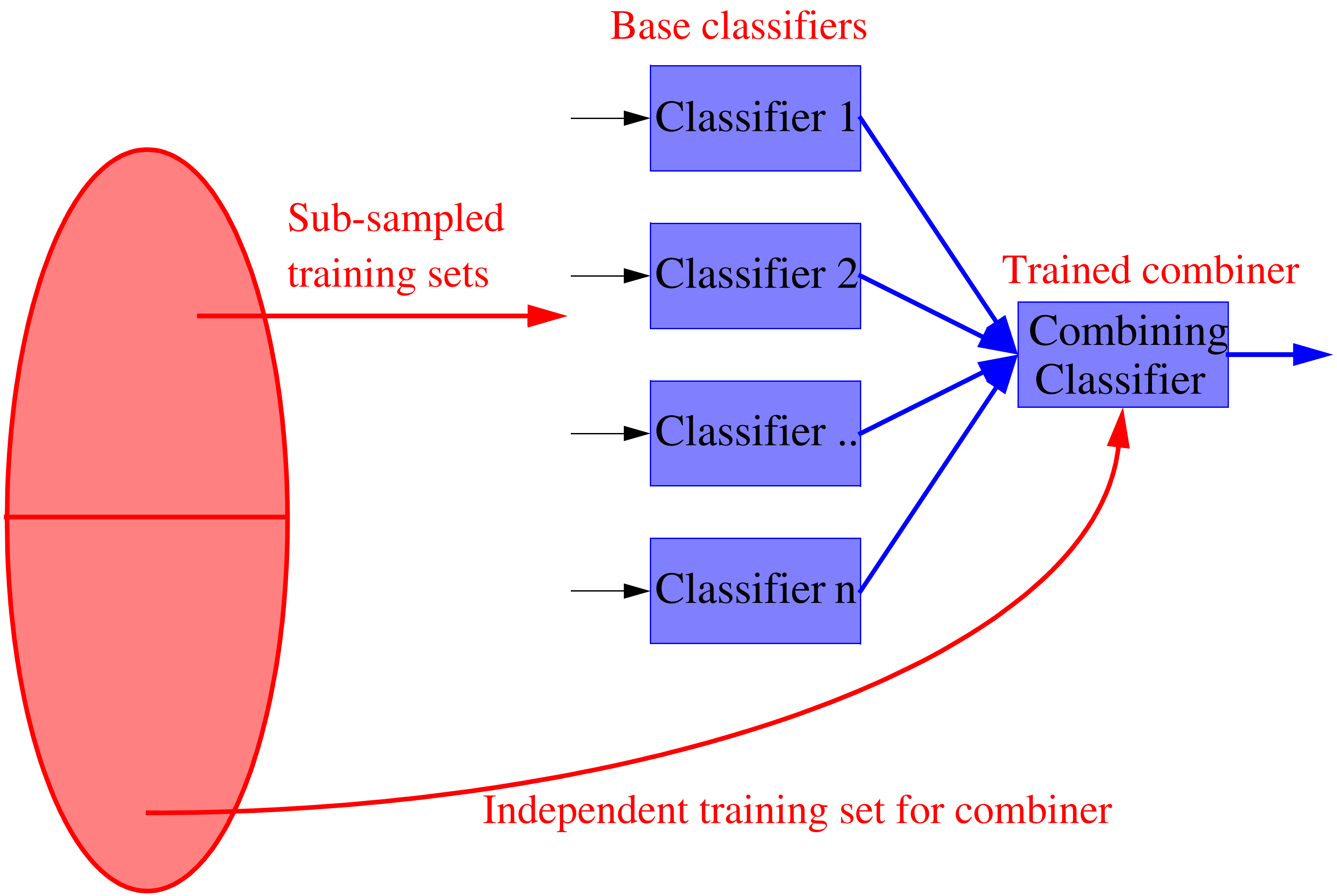
General rules neglect the classification-confidence characteristic of the base classifier outputs, as they are treated as general feature values.

* *Kuncheva, PR-23(2), 2001*

Example

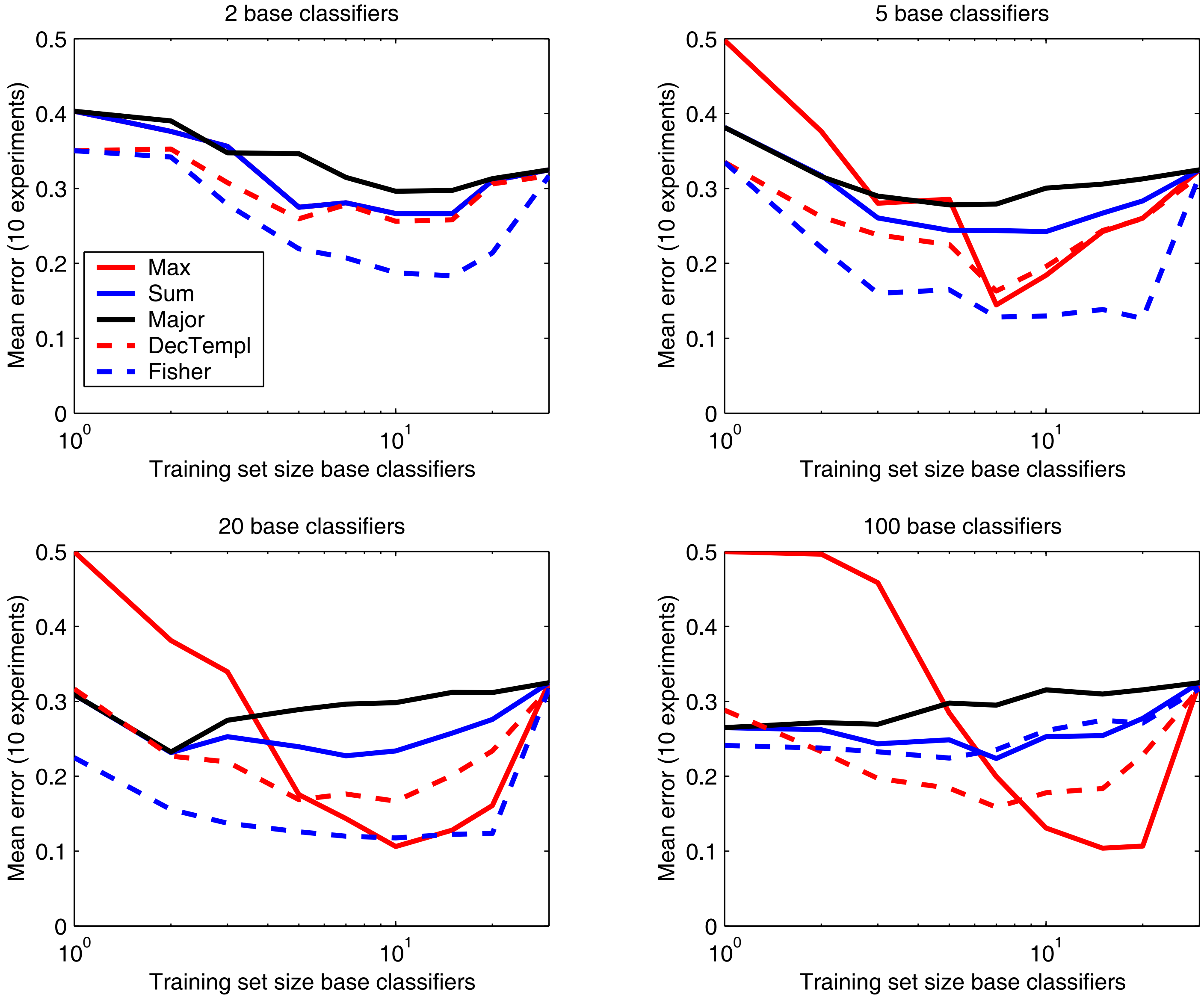

Combining 10 Bootstrapped Nearest Mean Classifiers
BaseClassf
Max
Sum
Major
DecTempl
Fisher
5
0
−5
−10
−5
0
5

Experiment


Base classifiers
Classifier 1
Classifier 2
Classifier ..
Classifier n
Sub-sampled
training sets
Trained combiner
Combining
Classifier
Independent training set for combiner

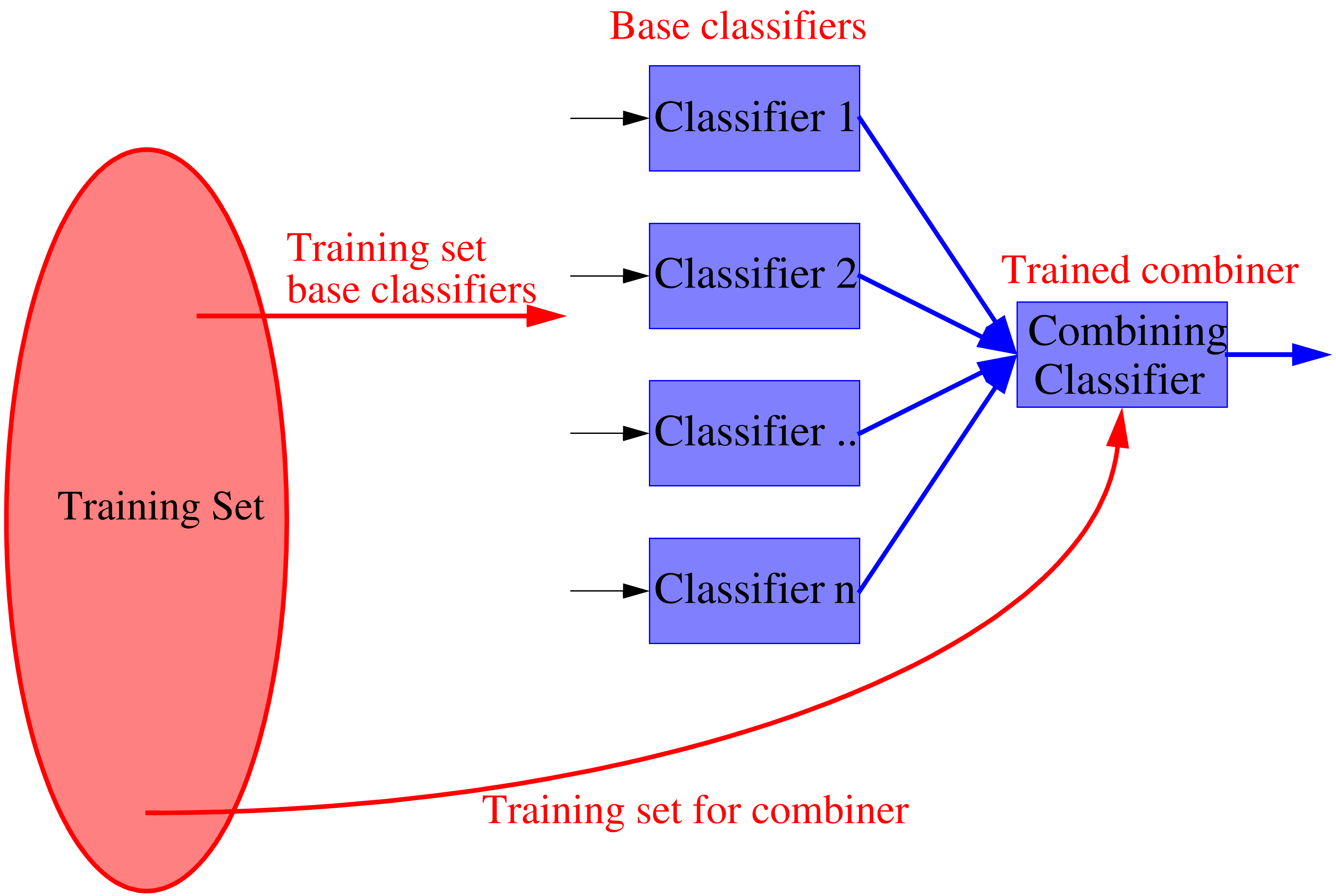
2-Class 10D Gaussian Experiment

30 samples per class for base classifiers, 30 samples per class for combining classifiers

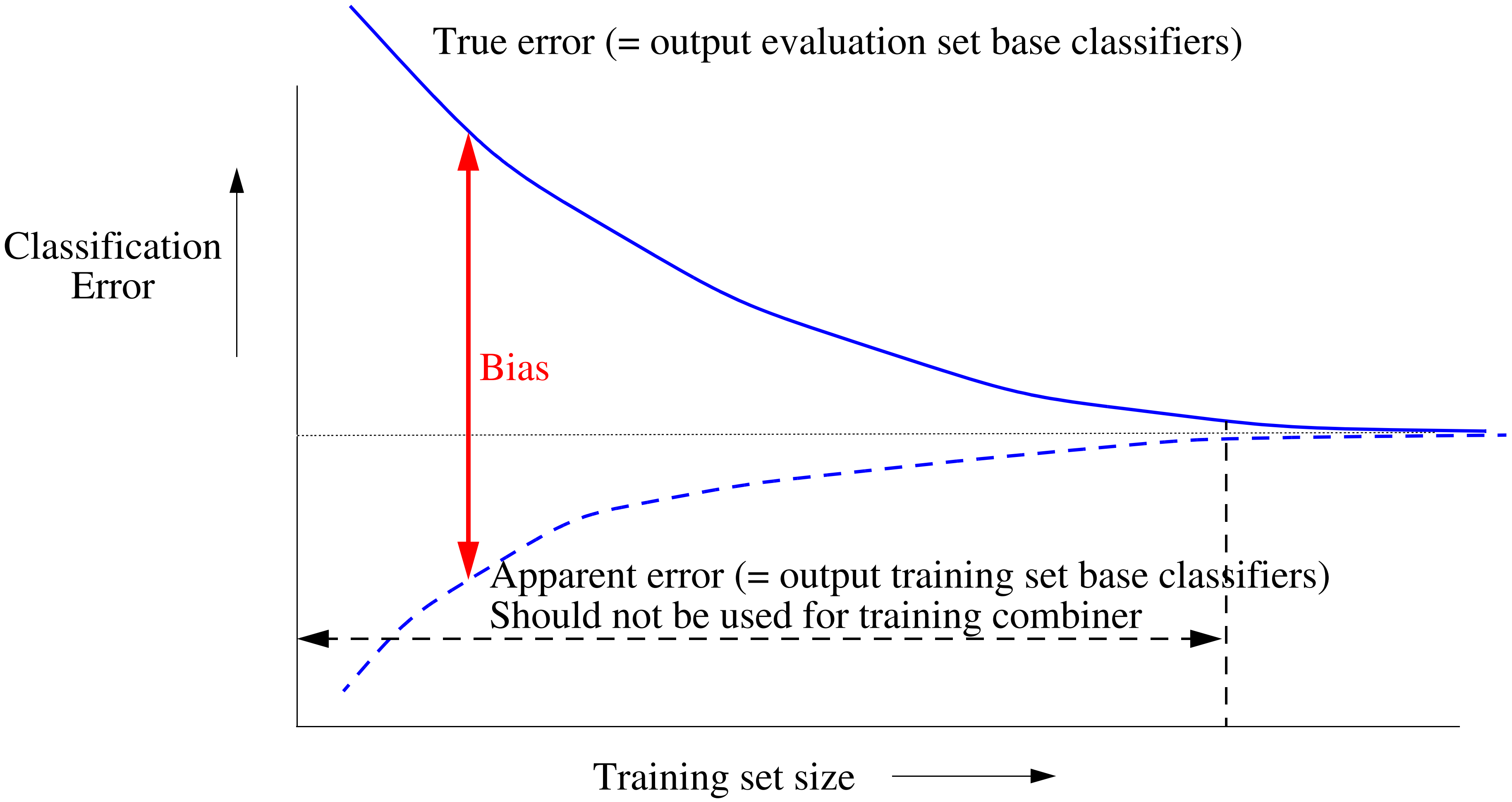


Trained rules are better for less base classifiers

Trained Combiners, a Single Training Set

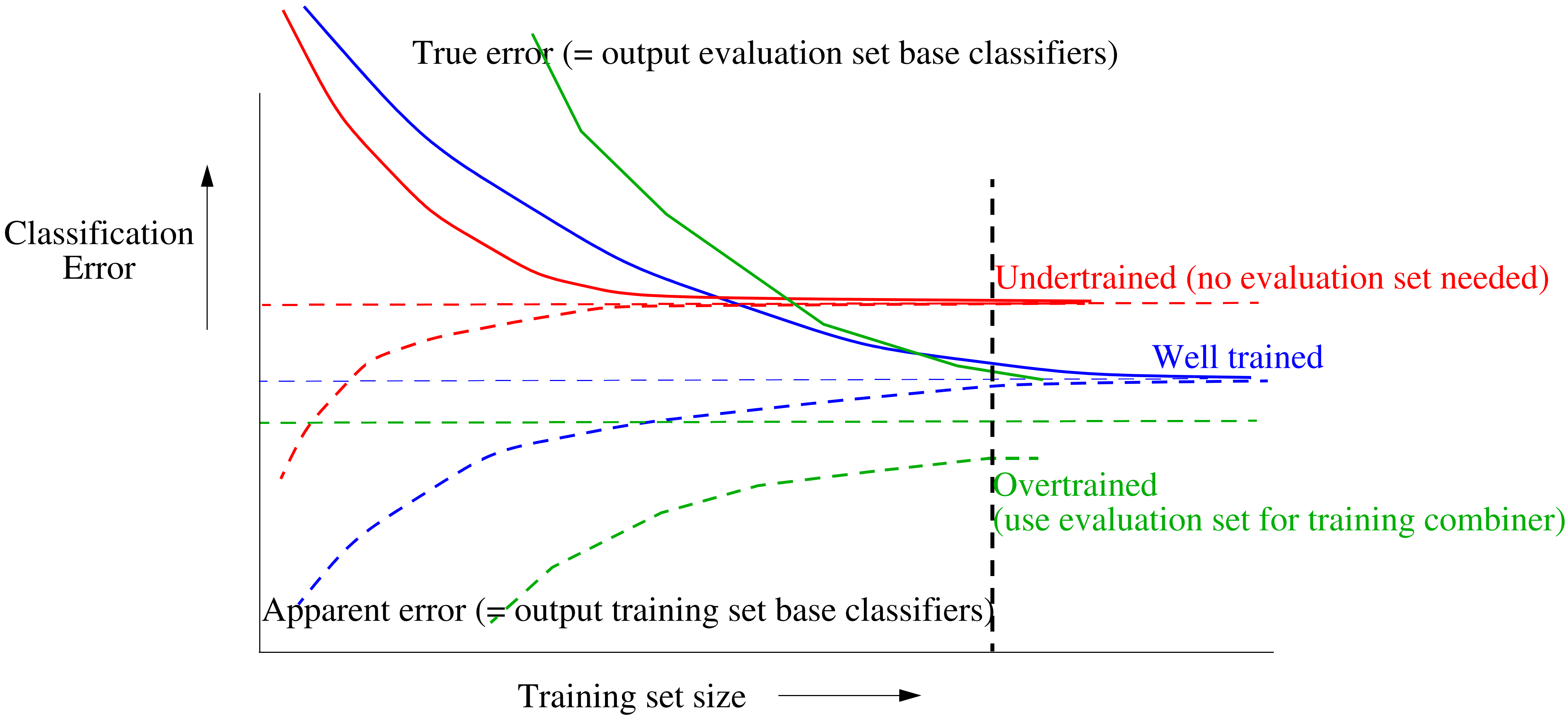


Biased Outputs Base Classifiers

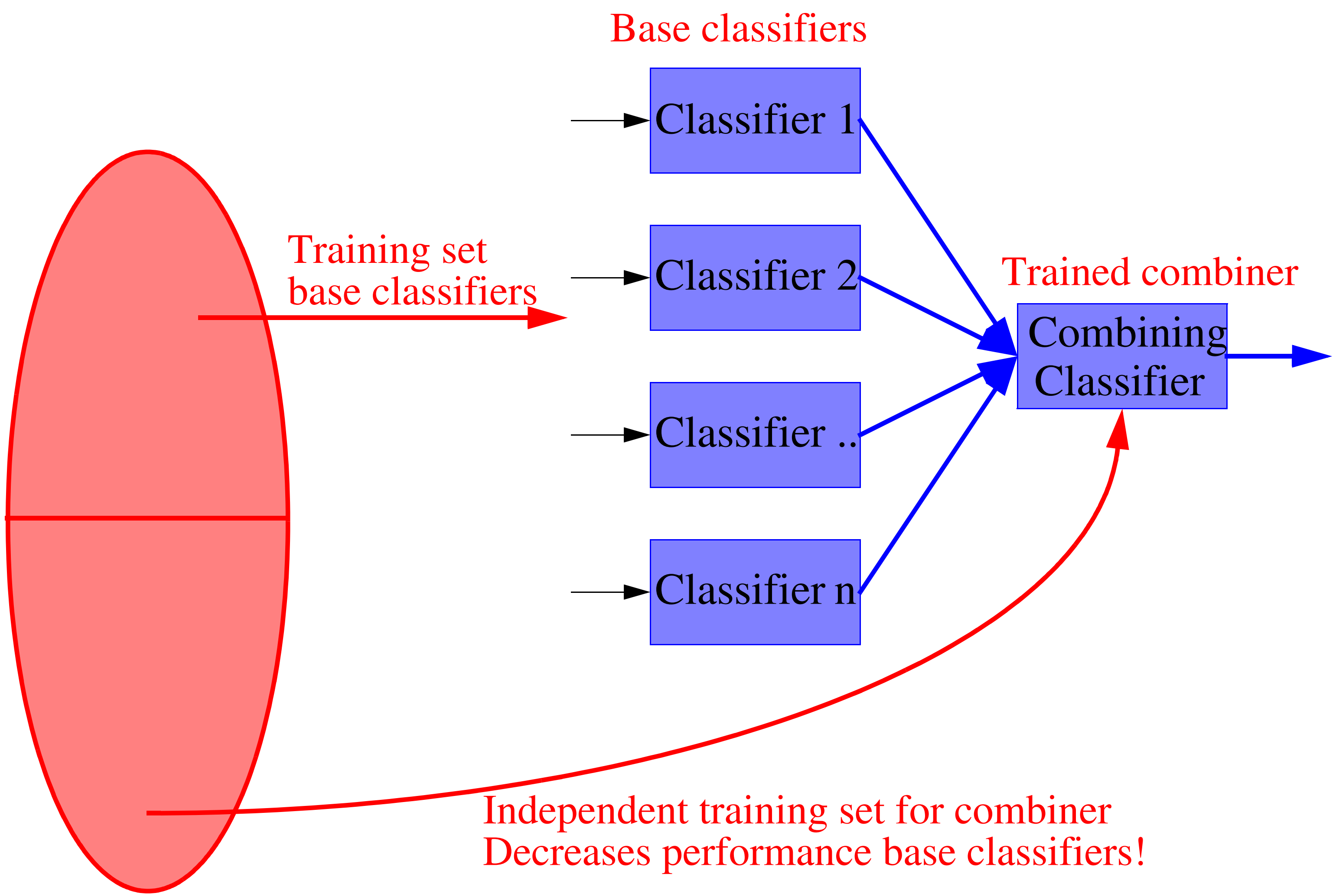


See also S. Raudys, MCS2002

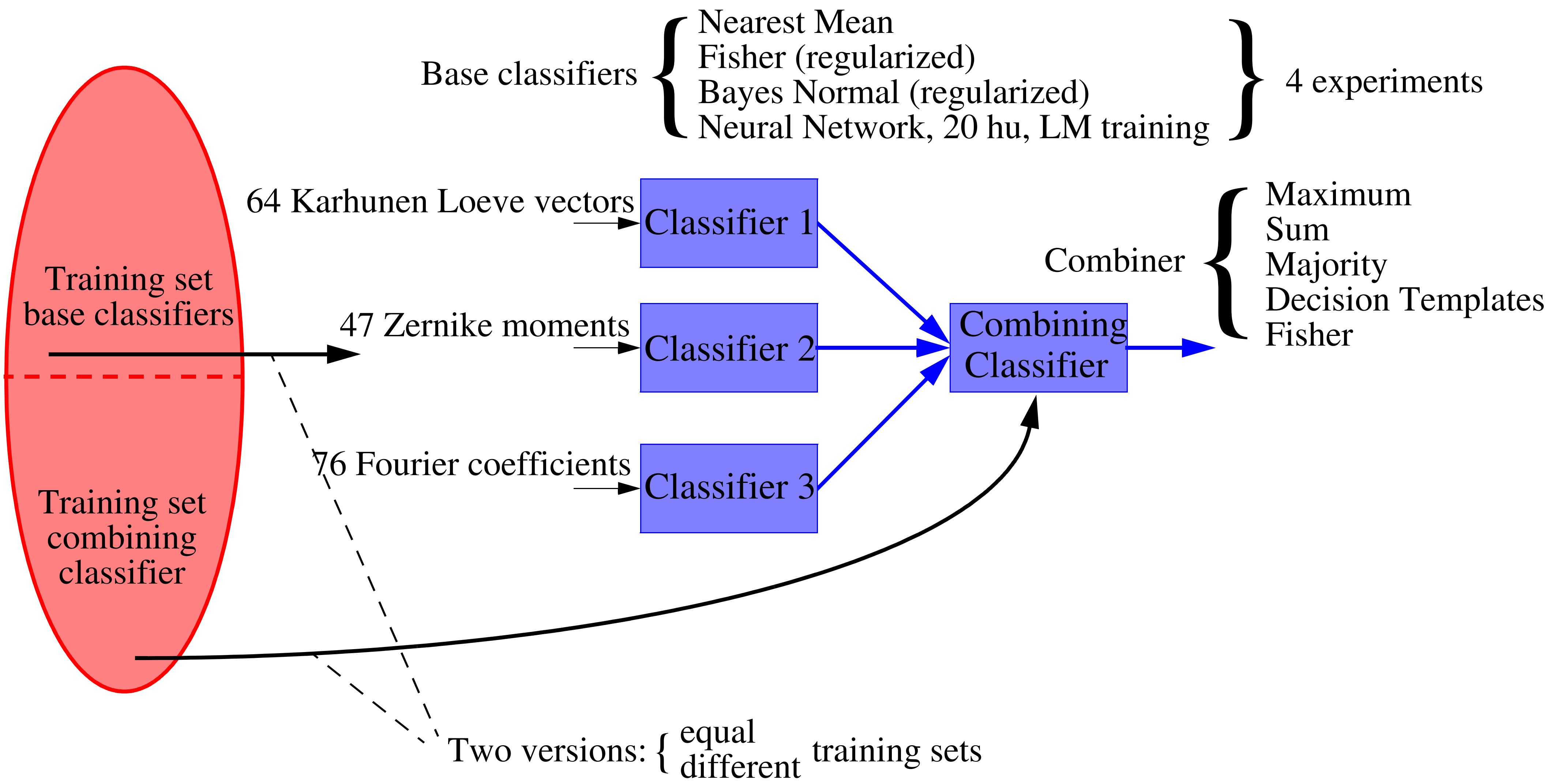
Combining Differently Trained Classifiers



Independent Training Set Combining Classifier



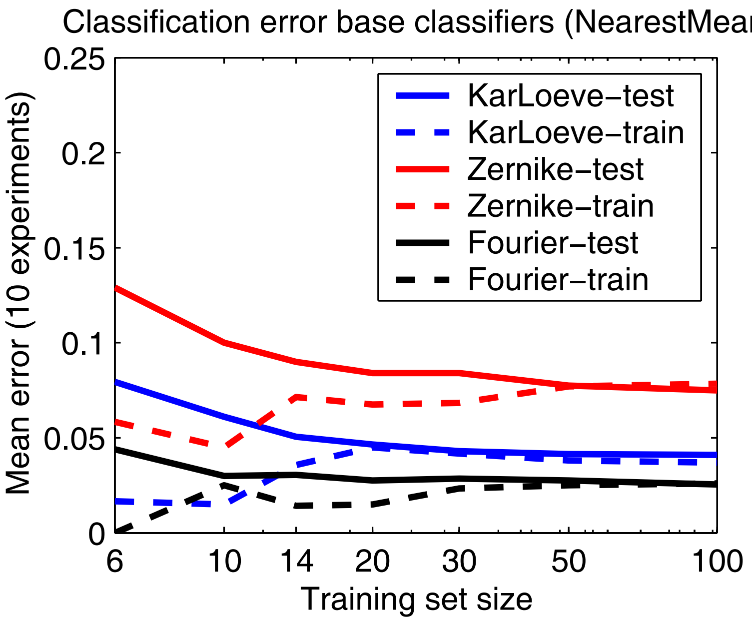
Example: Digit Classification: ‘3’ and ‘5’

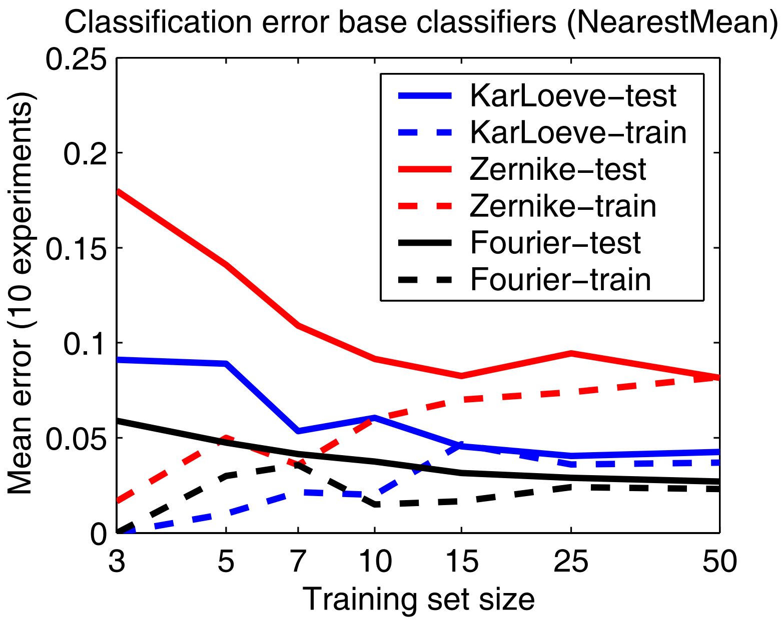


Base Classifiers: Nearest Mean

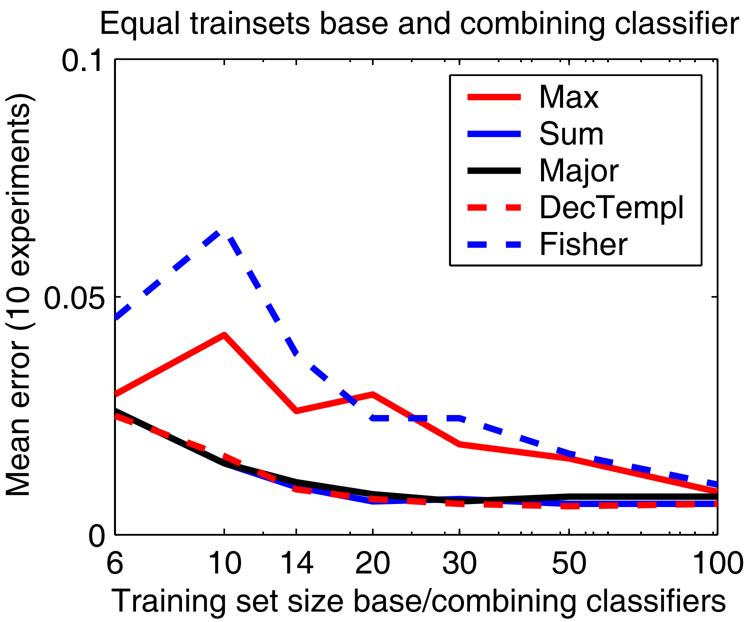
equal training sets | different training sets

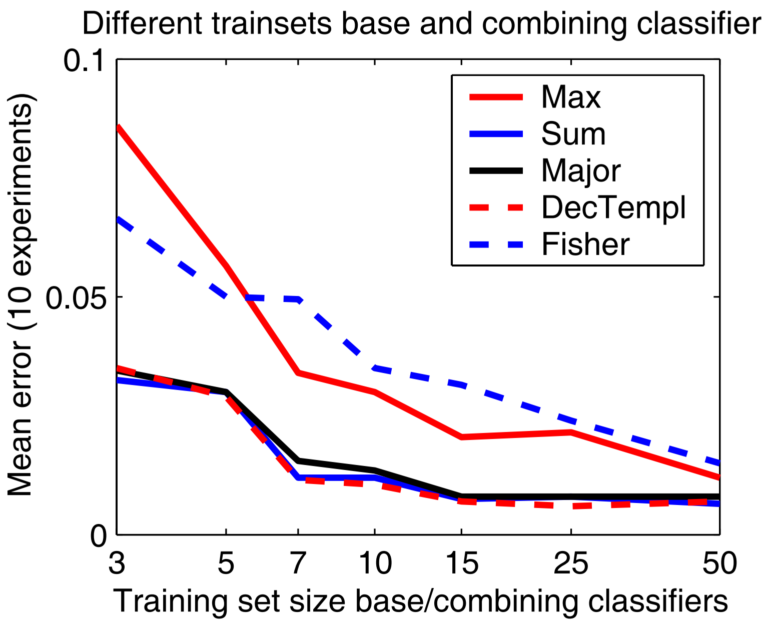
error base classifiers





error combining classifiers



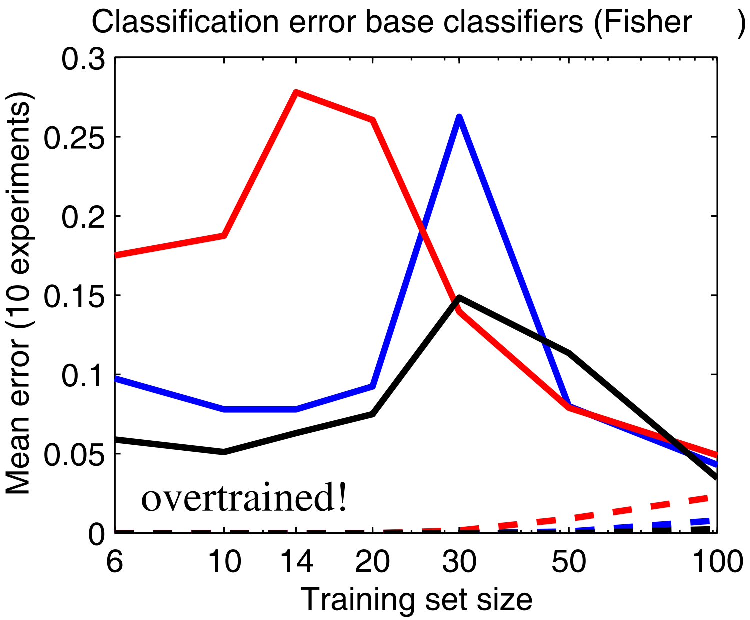


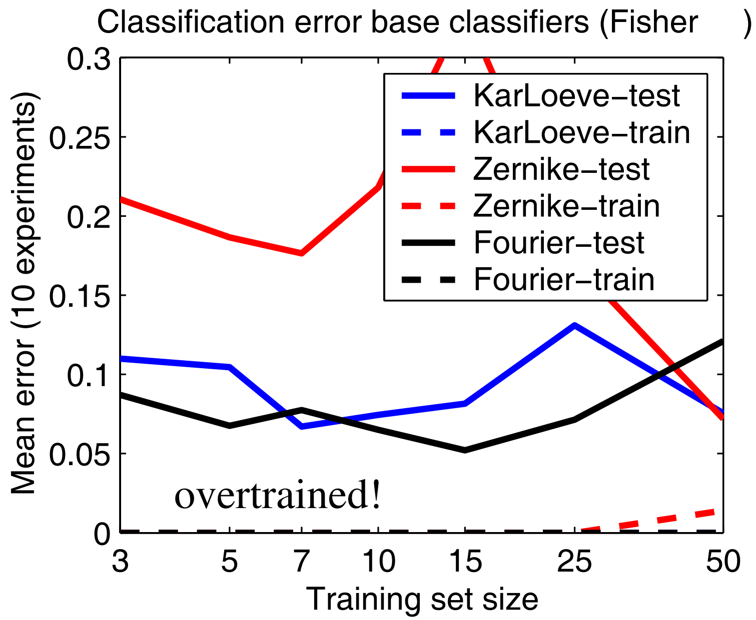
Base Classifiers: Fisher

equal training sets

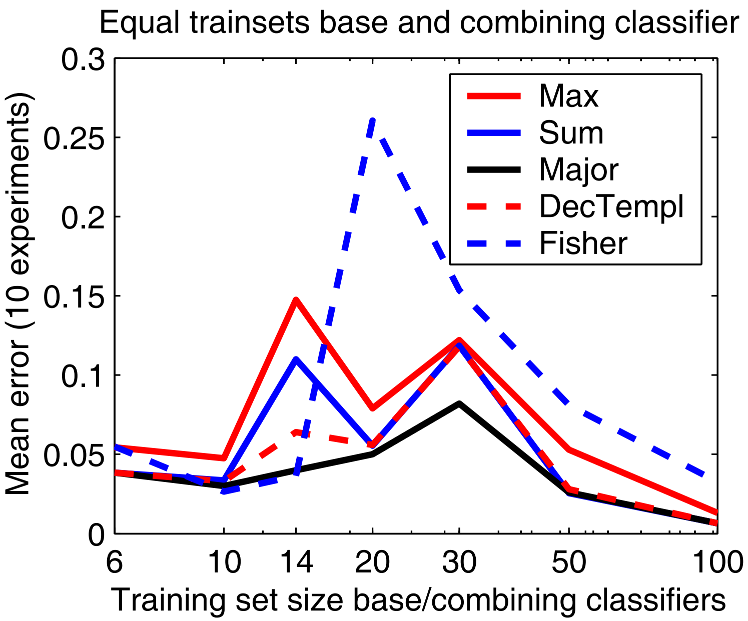
different training sets

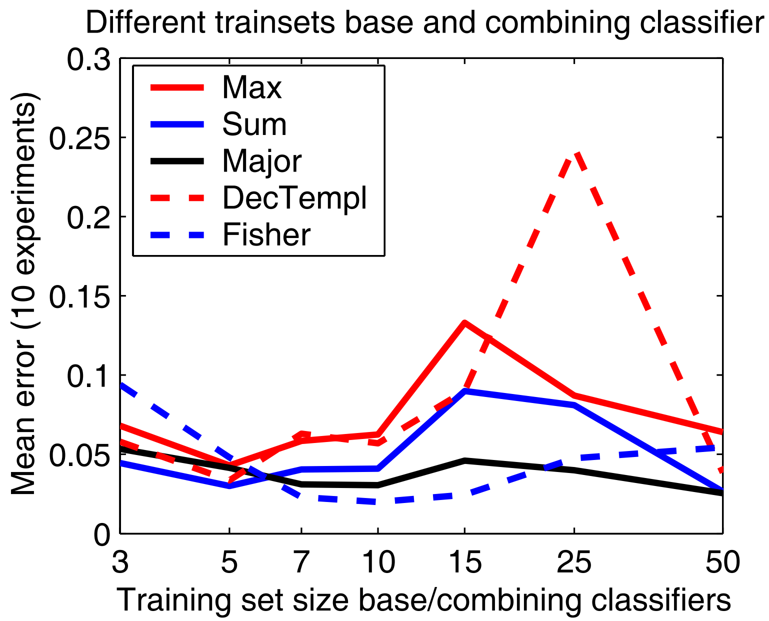
error base classifiers



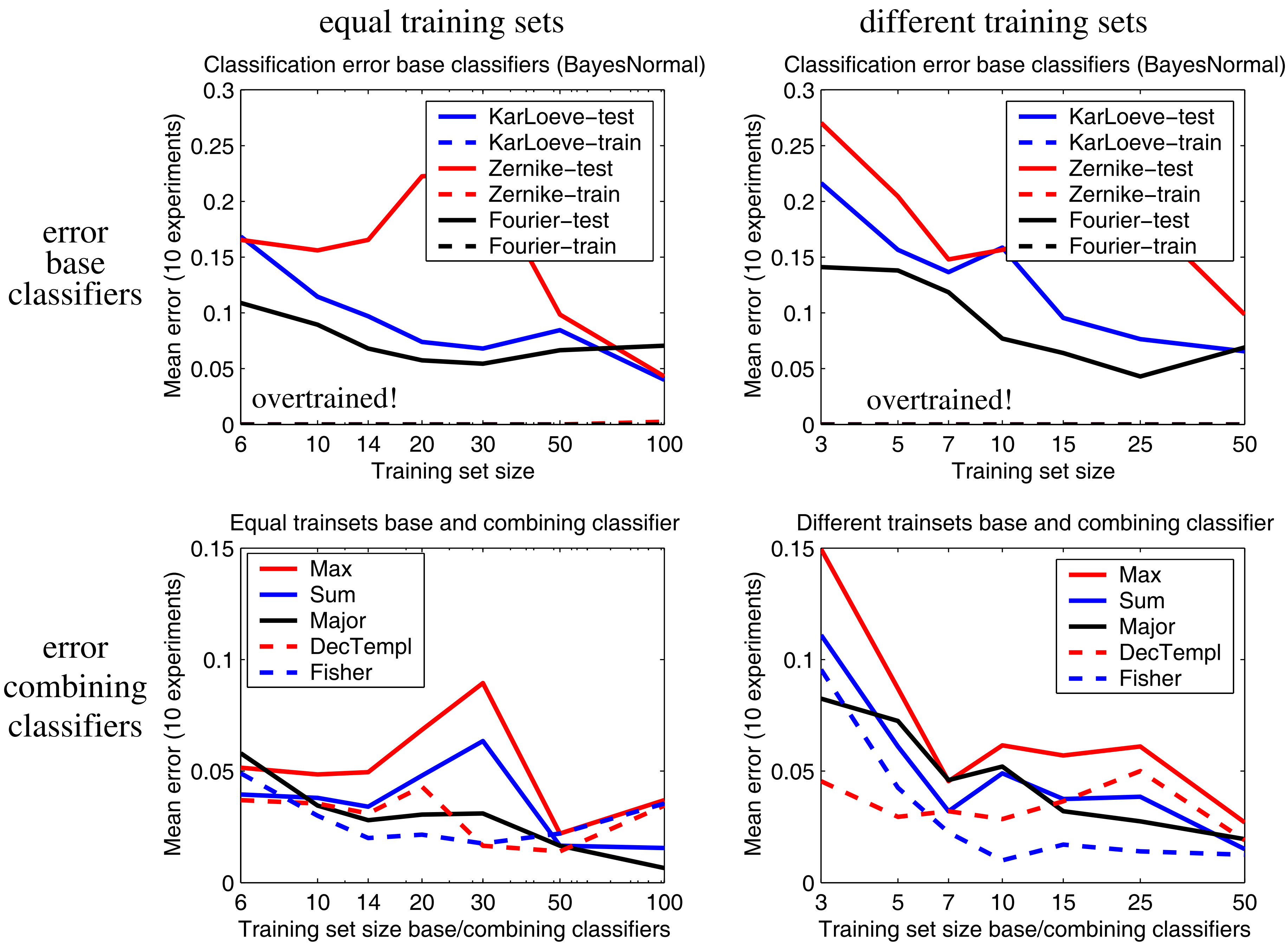


error combining classifiers

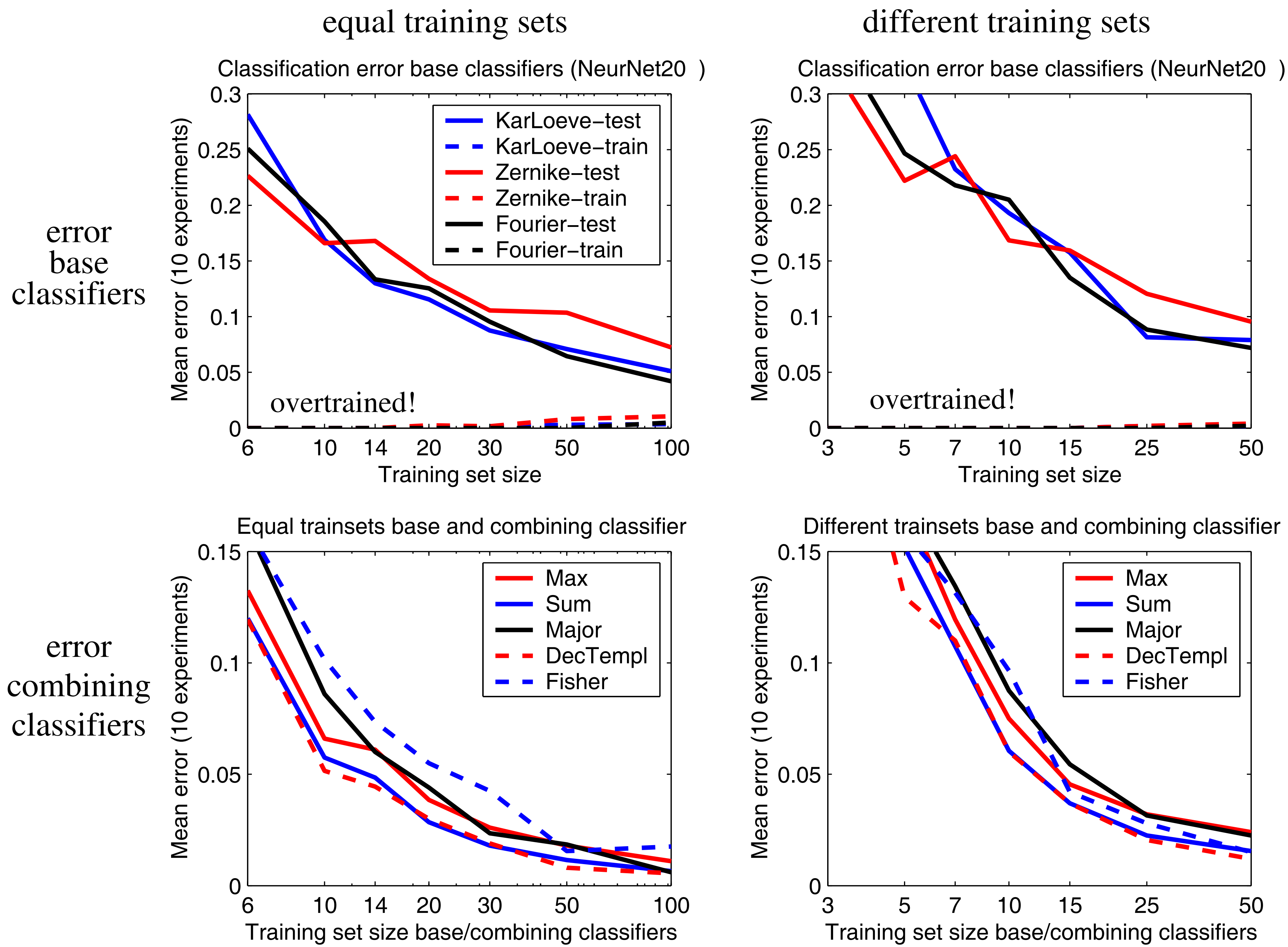




Base Classifiers: Bayes Normal



Base Classifiers: Neural Network (20 hu)



Observations

Undertrained (weak) base classifiers:

Reliable outputs --> fixed combiners may work

no separate training set needed for trained combiner

Overtrained base classifiers:

Unreliable, biased outputs --> majority voting may still work

separate training set needed for trained combiner

--> worse base classifiers

Possible Strategies:

1. Use just a single training set.
   Train the base classifiers carefully, avoiding overtraining.
   Fixed combining rules may work.

2. Use just a single training set.
   Train the base classifiers weakly.
   The same training set may be used for the combining classifier.

3. Separate the available training sets into two parts.
   Use one part for training the base classifiers. Some overtraining is not a problem.
   Use the other part for training the combining classifier.

Possible Strategy: 1

Use just a single training set.

Train the base classifiers carefully, avoiding overtraining.

Fixed combining rules may work.

Possible Strategy: 2

Use just a single training set.

Train the base classifiers weakly.

The same training set may be used for the combining classifier.

Possible Strategy: 3

Separate the available training sets into two parts.

Use one part for the base classifiers. Some overtraining is not a problem.

Use the other part for training the combining classifier.

Note Different Dimensionalities

Design set

Training base classifiers
Dimensionality = # features

Training output combiner
Dimension = #classes × #base classifiers

An equal split may not be the best!

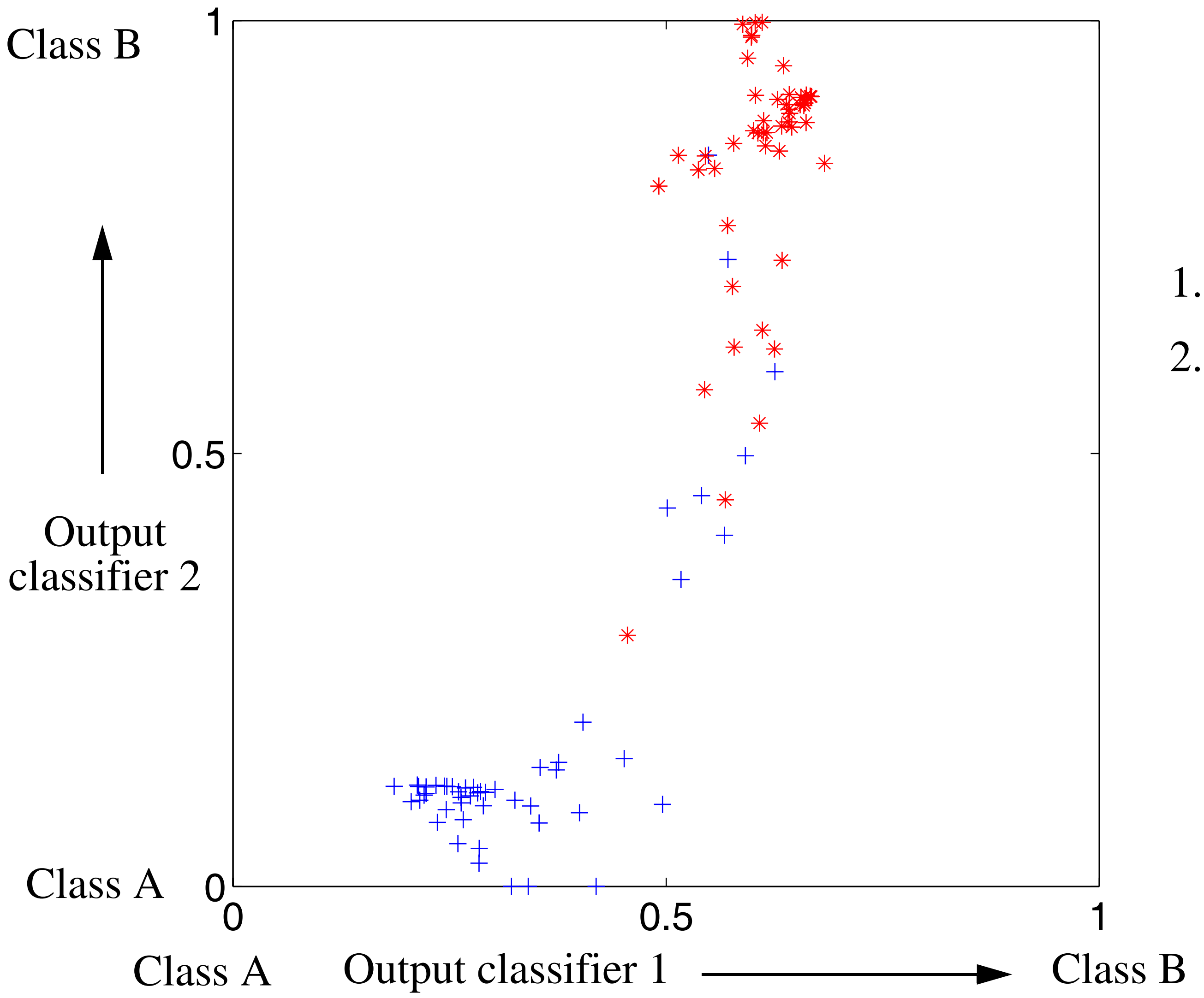
Special Training Rules

Global Selection	Select the best base classifier / fixed combining rule
Calibration:	Scale base classifier outputs in a similar way
Local Selection	Select the best base classifier for the object at hand
Decision Templates	Similar to the more general Nearest Mean Classifier

----

Other possibilities that use the specific character of the base classifier outputs for training?

Special Characteristics of the Combining Classifier Input Space



1. Features are ‘directed’
2. Class distributions far for normal

Relation with Dissimilarities

Dissimilarities in a training set.

$$D_T = \begin{pmatrix} d_{11} d_{12} d_{13} d_{14} d_{15} d_{16} d_{17} \\ d_{21} d_{22} d_{23} d_{24} d_{25} d_{26} d_{27} \\ d_{31} d_{32} d_{33} d_{34} d_{35} d_{36} d_{37} \\ d_{41} d_{42} d_{43} d_{44} d_{45} d_{46} d_{47} \\ d_{51} d_{52} d_{53} d_{54} d_{55} d_{56} d_{57} \\ d_{61} d_{62} d_{63} d_{64} d_{65} d_{66} d_{67} \\ d_{71} d_{72} d_{73} d_{74} d_{75} d_{76} d_{77} \end{pmatrix}$$

$$d_x = (d_1 \ d_2 \ d_3 \ d_4 \ d_5 \ d_6 \ d_7)$$

The traditional Nearest Neighbor rule classifies new objects just by:

Label($\mathrm{argmin}_i(d_i)$) without using D_T for training.

This is similar to a fixed combiner (max-rule).
Trained classifiers for dissimilarities exist and may perform better.
Pekalska et al., PRL-23(8), 2002

Conclusions on Fixed versus Trained Rules for Combiners

Fixed rules are hardly ever theoritically optimal, but perform sometimes surprisingly good.

Trained rules can be optimal for large training sets.

Use of the same training set is might be good for well / undertrained base classifiers.

Different training sets are needed for well / overtrained base classifiers.

How to split the total design set over training sets needs more study.

'Decision templates' is a good training rule, unless we have many base classifiers.

Special purpose combiners are to be developed.