Multiple Classifier Systems:

Saroj K. Meher

Systems Science and Informatics Unit,

Indian Statistical Institute, Bangalore

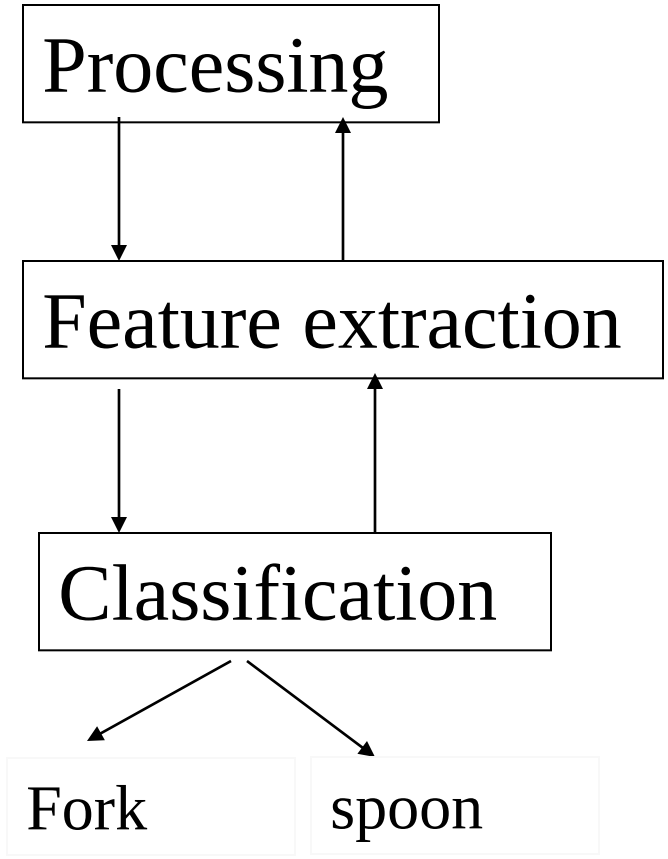
Objective: MCS

- An introduction on multiple classifier combination
- Motivation and basic concepts
- Why could we integrate classifiers?
- When do multiple classifier work?
- Main methods for creating multiple classifiers
- Main methods for fusing multiple classifiers
- Applications, achievement and open issues

Pattern Classification System

Measurement	$\rightarrow$	*Feature*	$\rightarrow$	*Decision*
Space		*Space*		*Space*

Pattern Classifier System



Pattern Classifier System (cont'd)

- A "classifier" is any mapping from the space of features(measurements) to a space of class labels (names, tags, distances, probabilities)
- A classifier is a hypothesis about the real relation between features and class labels
- A "learning algorithm" is a method to construct hypotheses
- A learning algorithm applied to a set of samples (training set) outputs a classifier

Pattern Classifier System: Issues

- Unfortunately, no dominant classifier exists for all the data distributions, and the data distribution of the task at hand is usually unknown
- Not one classifier can discriminative well enough if the number of classes are huge
- For applications where the objects/classes of content are numerous, unlimited, unpredictable, one specific classifier/detector cannot solve the problem.
- Although one of the designs would yield the best performance, the sets of patterns misclassified by the different classifiers would not necessarily overlap,
- Different classifier designs potentially offered complementary information about the patterns to be classified, which could be harnessed to improve the performance of the selected classifier,
- The idea is not to rely on a single decision making scheme. Instead, all the designs, or their subset, are used for decision making by combining their individual opinions to derive a consensus decision.
- No classifier is known to be the best for all cases and its selection for a given practical task is very difficult.

Why could we integrate classifiers ?

- Independent classifiers for the same goal.
  - Person identification by voice, face and handwriting.
- Sometimes more than a single training set is available, each collected at different time or in a different environment. These training sets may even use different features.
- Different classifiers trained on the same data may not only differ in their global performance, but they also may show strong local differences. Each classifier may have its own region in the feature space where it performs the best.
- Some classifiers such as neural networks show different results with different initializations due to the randomness inherent in the training procedure. Instead of selecting the best network and discarding the others, one can combine various networks, thereby taking advantage of all the attempts to learn from data.

Why could we integrate classifiers ? (cont'd)

- Beside avoiding the selection of the worse classifier, under particular hypothesis, 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
- For linear combiners, Turner and Ghosh (1996) showed that averaging outputs of individual classifiers with unbiased and uncorrelated errors can improve the performance of the best individual classifier and, for infinite number of classifiers, provide the optimal Bayes classifier

Multiple classifier systems (Definition)

- A multiple classifier system (MCS) is a structured way to combine (exploit) the outputs of individual classifiers
- MCS can be thought as:
  - Multiple expert systems
  - Committees of experts
  - Mixtures of experts
  - Classifier ensembles
  - Composite classifier systems

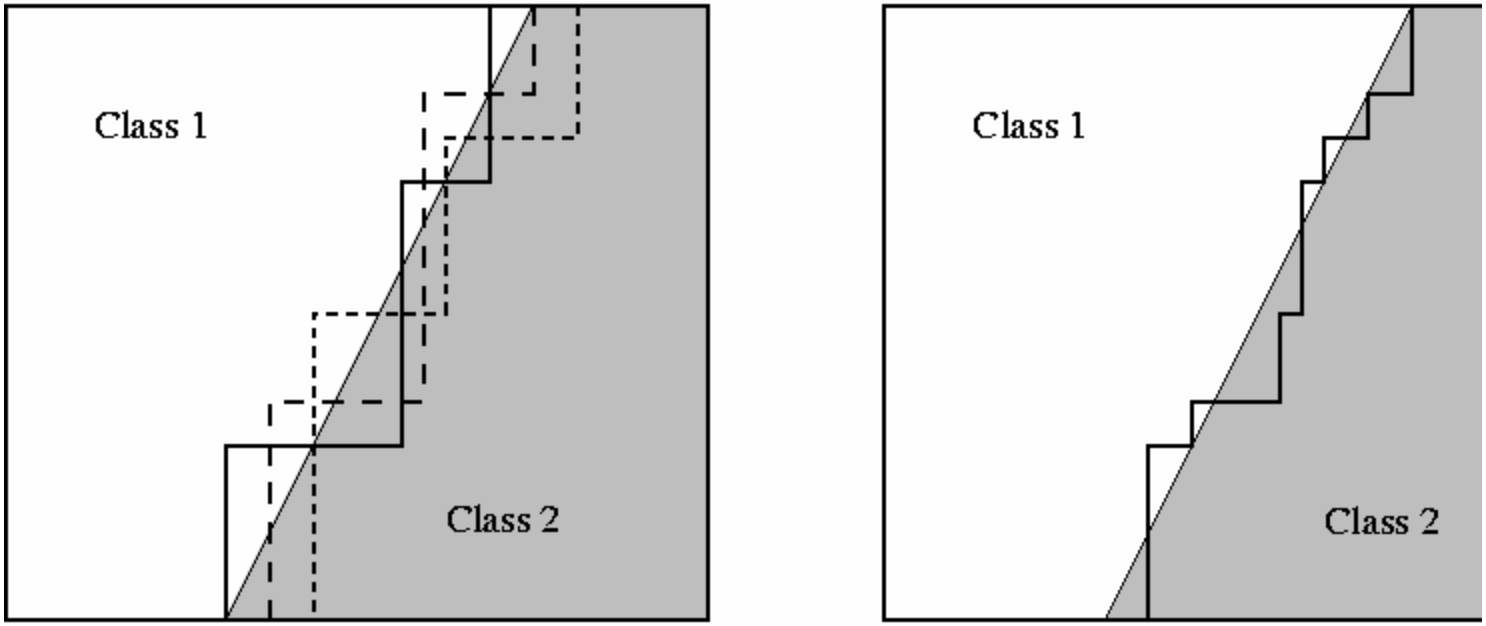
Why do multiple classifiers work ?

Dietterich(2002) showed that ensembles overcome three problems:

- The Statistical Problem arises when the hypothesis space is too large for the amount of available data. Hence, there are many hypotheses with the same accuracy on the data and the learning algorithm chooses only one of them! There is a risk that the accuracy of the chosen hypothesis is low on unseen data!
- The Computational Problem arises when the learning algorithm cannot guarantee finding the best hypothesis.
- The Representational Problem arises when the hypothesis space does not contain any good approximation of the target class(es).

Why do multiple classifiers work ?

- The diagonal decision boundary may be difficult for individual classifiers, but may be approximated by ensemble averaging.
- Decision boundaries constricted by decision trees → hyperplanes parallel to the coordinate axis – “staircases”.
- By averaging a large number of „staircases” the diagonal boundary can be approximated with some accuracy.



Multiple Classifier Systems (MCS)

Sensor Fusion:

- Sensor Fusion: use of data from multiple sensors in an intelligent system to form one representation in order to improve accuracy.
- Sensor Integration: use of multiple sensors to provide information about a sub-task during the different modes of operation.

Creating Classifier Ensembles

- Different feature spaces: face, voice, fingerprint;
- Different training sets: Sampling;
- Different classifiers: K_NN, Neural Net, SVM;
- Different architectures: Neural net: layers, Units, transfer function;
- Different parameter values: K in K_NN, Kernel in SVM;
- Different initializations: Neural net

Creating Classifier Ensembles

- Varying the set of initializations: A number of distinct classifiers can be built with different learning parameters, such as the initial weights in an MLP, etc
- Varying the topology: Using different topologies, or architectures, for classification can lead to different generalization models
- Varying the algorithm employed: Applying different classification algorithms for the same topology may produce diverse classifiers
- Varying the data: The mostly used approach to produce classifiers with different generalizations.

Varying the Data (cont'd)

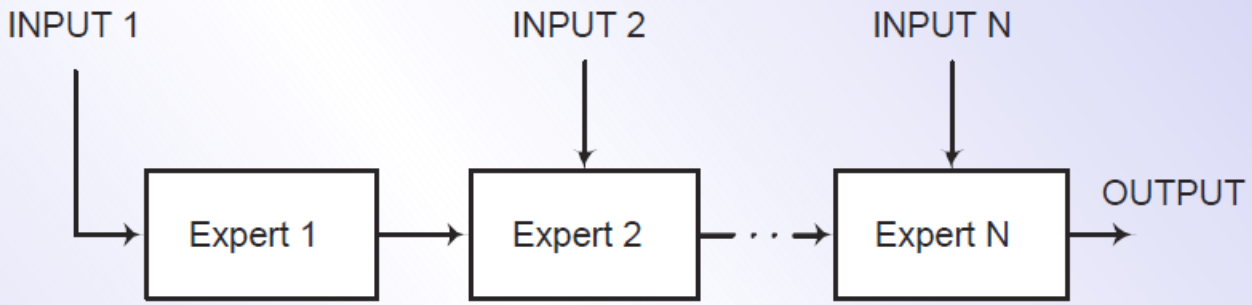
- Sampling Data: A common approach is to use some sort of sampling technique, such that different classifiers are trained on different subsets of the data.
- Disjoint Training Sets: Similar to sampling, however, uses mutually exclusive, or disjoint, training sets. That is we use sampling without replacement to avoid overlap between the training sets.
- Boosting and Adaptive Re-sampling: A series of weak learner can be converted to a strong learner using boosting.
- Different Data Sources: Under the circumstances that data from different input sources (e.g. sensors) are available. It is especially useful when these sources provide different sources of information.
- Preprocessing: Data maybe varied by applying different pre-processing methods to each set. Alternatively, data sets maybe distorted differently.

Multiple Classifier Systems (MCS): Basic concept

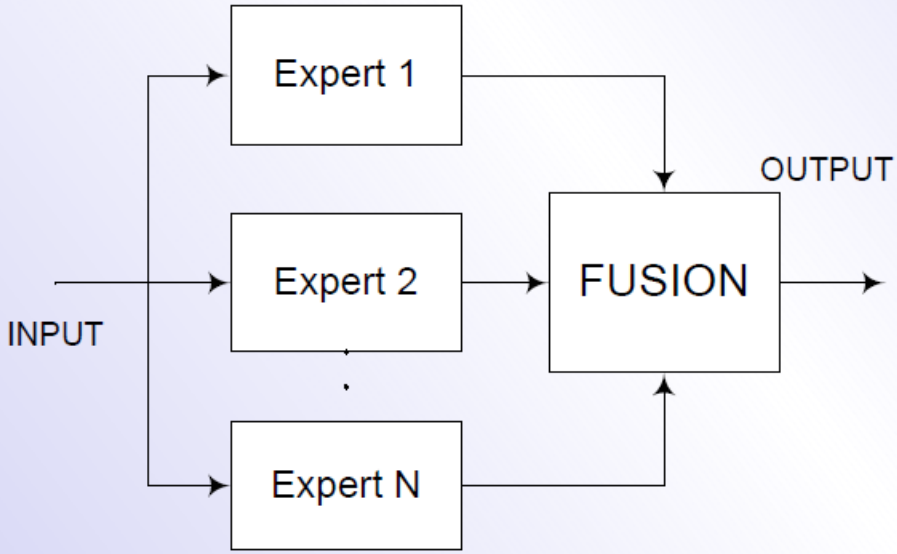
- Multiple Classifier Systems (MCS) can be characterized by:
  - The Architecture
  - Fixed/Trained Combination strategy
  - Others

Multiple Classifier Systems (MCS) (cont'd)

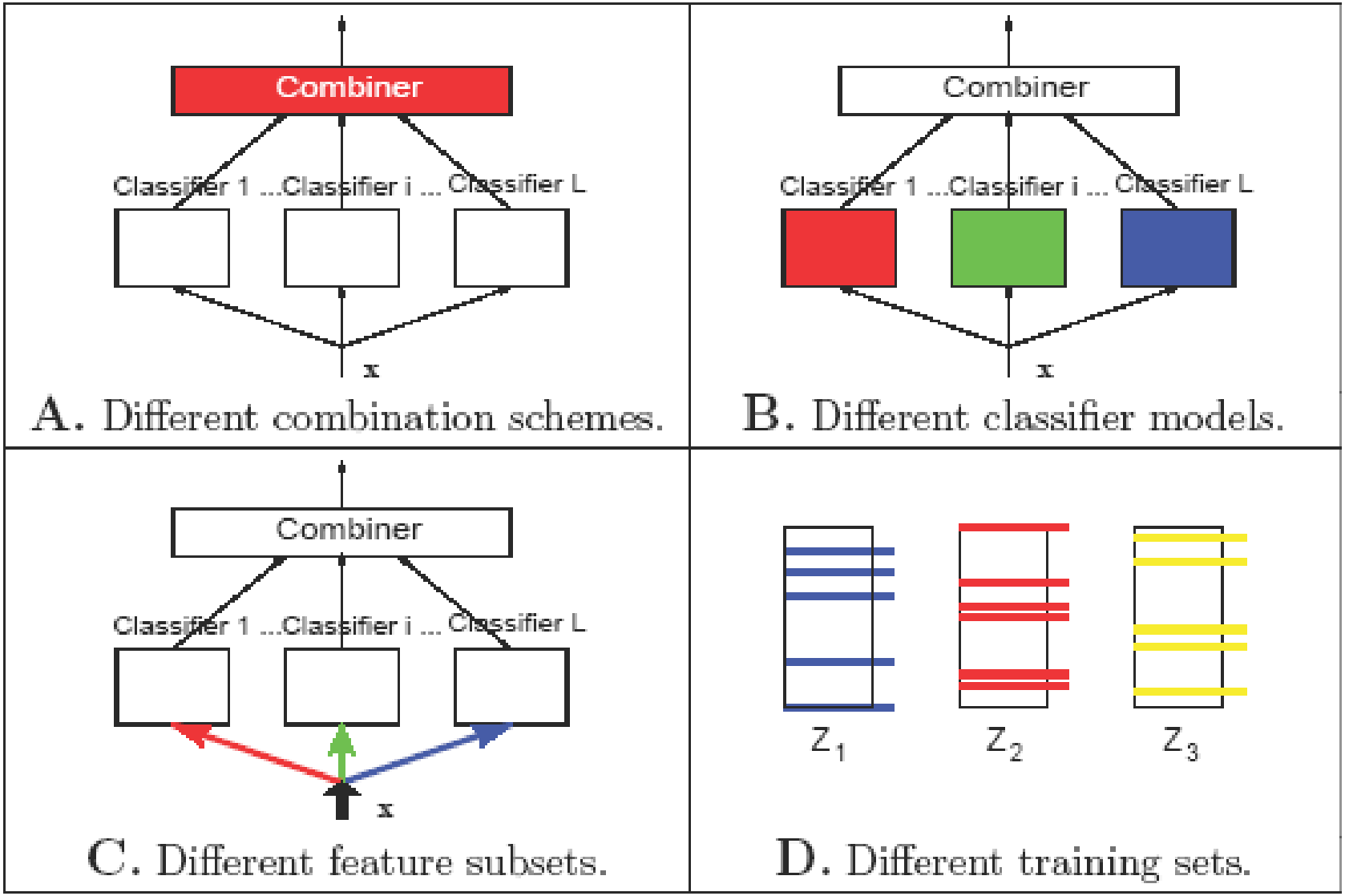
★ Cascade



★ Parallel



Multiple Classifier Systems (MCS) (cont'd)



Four approaches of designing a classifier combination system

Fixed Combination Rules

- Product, Minimum
  - Independent feature spaces;
  - Different areas of expertise;
  - Error free posterior probability estimates
- 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 Rule
  - Trust the most confident classifier/expert;
  - Bad if some classifiers(experts) are badly trained.

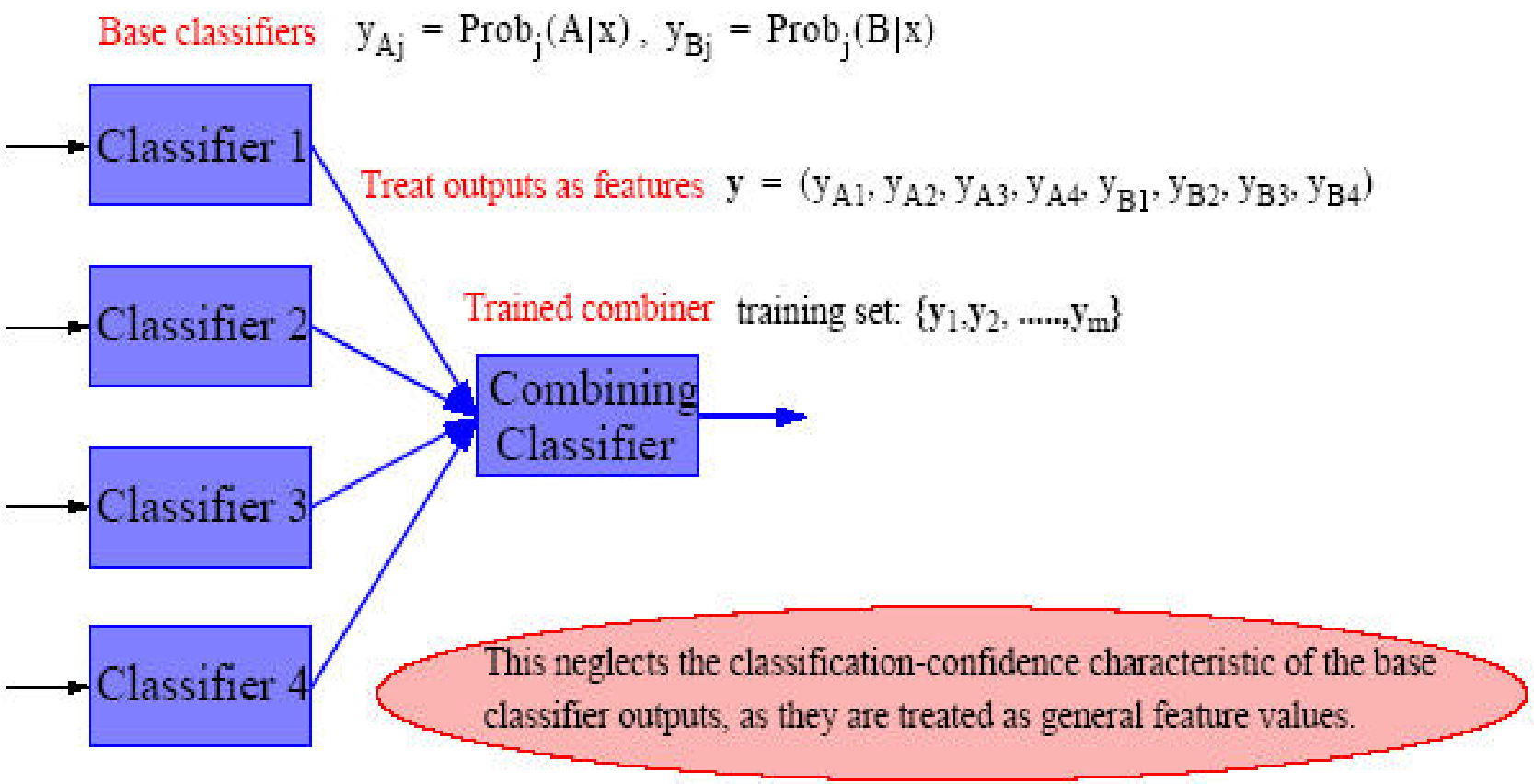
Ever optimal?

Fixed Combination 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)

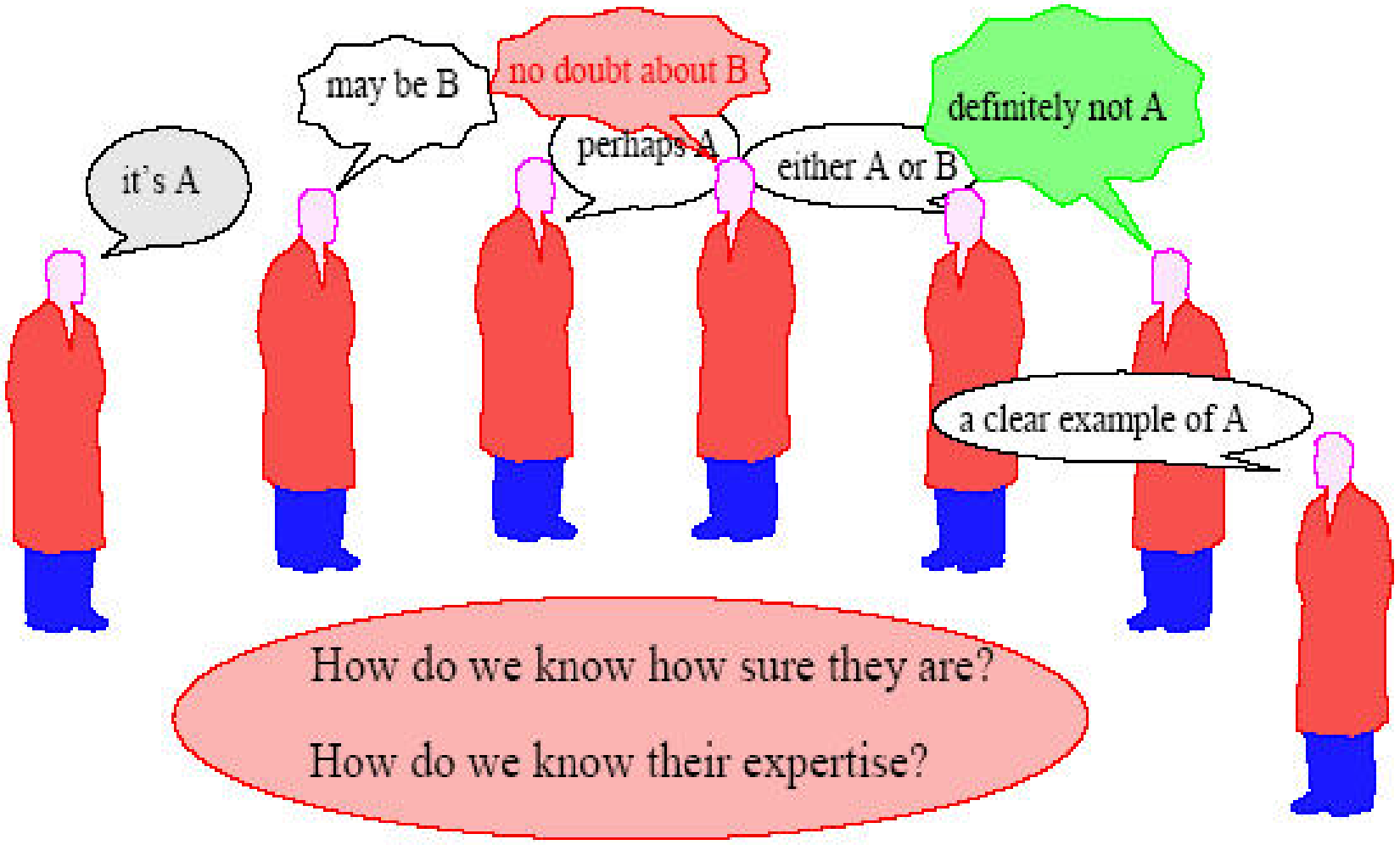
Fixed combining rules are never optimal

Trained combiner



Remarks on fixed and trained combination strategies

- Fixed rules
  - Simplicity
  - Low memory and time requirements
  - Well-suited for ensembles of classifiers with independent/low correlated errors and similar performances
- Trained rules
  - Flexibility: potentially better performances than fixed rules
  - Trained rules are claimed to be more suitable than fixed ones for classifiers correlated or exhibiting different performances
  - High memory and time requirements


it's A
may be B
no doubt about B
perhaps A
either A or B
definitely not A
a clear example of A
How do we know how sure they are?
How do we know their expertise?

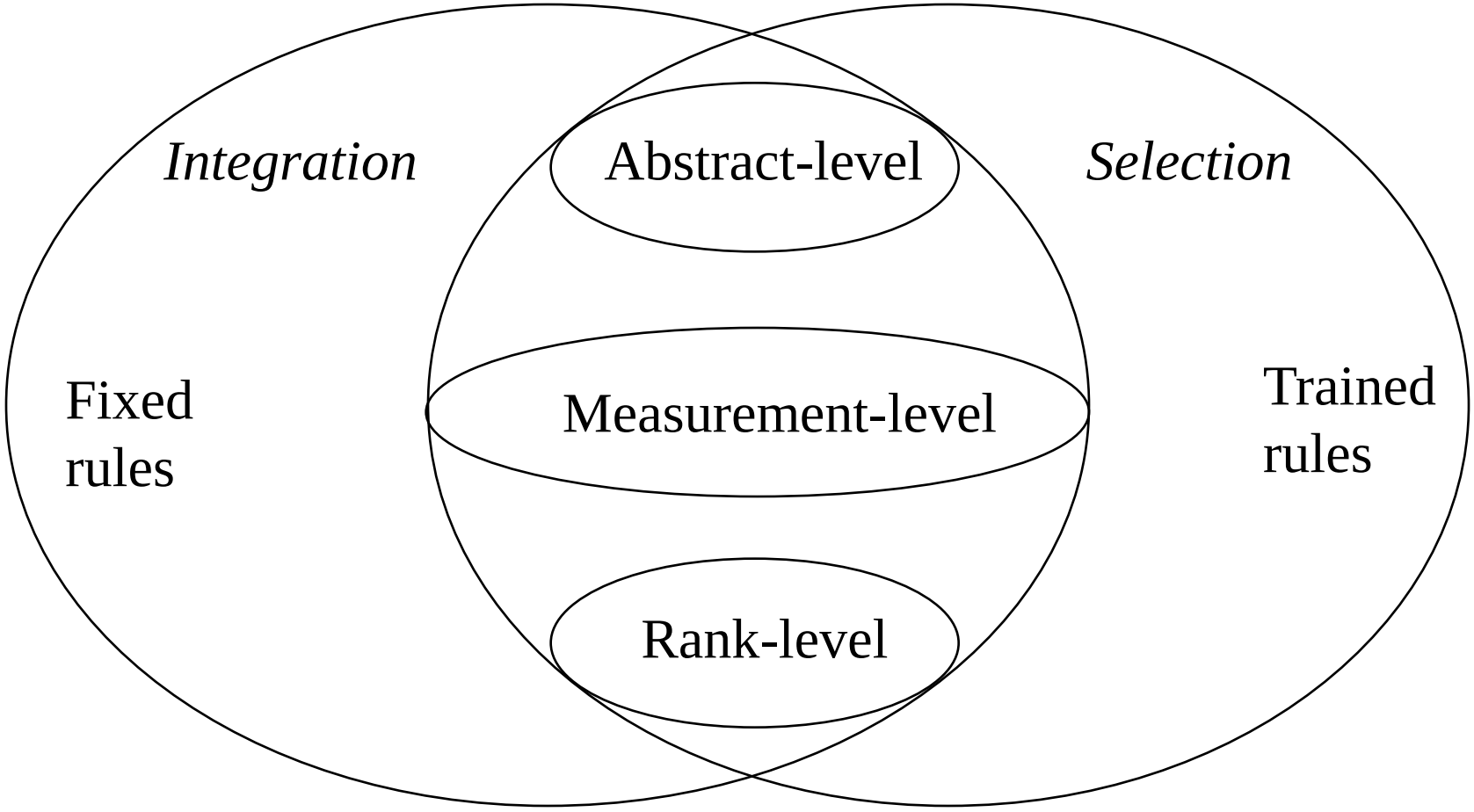
Multiple Classifier Systems (MCS) (cont'd)

Classifier Output Level of Representation

- **Abstract Level output:**
  - Each classifier outputs a unique class label for each input pattern
- **Rank Level output:**
  - Each classifier outputs a list of possible classes, with ranking, for each input pattern
- **Measurement Level output:**
  - Each classifier outputs class "confidence" levels for each input pattern

For each of the above categories, methods can be further subdivided into: Integration vs. Selection rules and Fixed rules vs. Trained Rules

Multiple Classifier Systems (MCS) (cont'd)



Design of Classifier Ensembles

- How do we create the individual classifiers?
- How do we perform the combination of these classifiers?

Combining Classifier Ensembles

- Averaging and Weighted Averaging
- Non-linear Combining Methods
  - Voting Methods
  - Rank Based Methods
  - Probabilistic methods
- Fuzzy Integral Methods

Average Vote

$$Q(x) = \arg\max_{j=1}^{N} \left(\frac{1}{K} \sum_{i=1}^{K} y_{ij}(x) \right)$$

- *N* is the number of classes
- *x* is the input pattern
- *K* represents the number of classifiers
- $y_{ik}(x)$ represents the output of the i_{th} classifier for the j_{th} class for the input *x*

Assign a number between zero and one for each candidate.

Compare the Summation of the votes value. The higher is the winner.

Drawbacks: Sensitive towards skewed classifier values of voting.

Weighted Average

$$Q(x) = \arg\max_{j=1}^{N} \left(\frac{1}{K} \sum_{i=1}^{K} w_i y_{ij}(x) \right)$$

- *N* is the number of classes
- *x* is the input pattern
- *K* represents the number of classifiers
- $y_{ik}(x)$ represents the output of the i_{th} classifier for the j_{th} class for the input *x*

The weights w_i, $i = 1, \cdots, K$ can be derived by minimizing the error of the different classifiers on the training set.

Non Linear Combining Methods

- Voting Methods
  - Majority, Maximum, Minimum, Prod, etc...
- Rank Based Methods
  - Borda Count
- Probablistic methods
  - Bayesian Methods

Majority Vote Rule

- Majority Vote
  - Bad if some classifiers (experts) are very good or very bad
- Maximum Vote
  - Trust the most confident classifier/expert
  - Bad if some classifiers (experts) are badly trained
  - Sensitivity to over-confident base classifiers
- Product Rule
  - Base Classifiers are never really independent

Majority Vote Rule (cont'd)

Usually *N* is odd.

The frequency of the winner class must be at least N/2.

If the *N* classifiers make independent errors and they have the same error probability e<0.5, then it can be shown that the error E of the majority voting rule is monotonically decreasing in N (Hansen and Salamon, IEEE-T on PAMI, 1990):

$$\lim_{N\to\infty} \sum_{k>\frac{N}{2}}^{N} \binom{N}{k} e^k (1-e)^{N-k} = 0$$

Clearly, performances of majority vote quickly decreases for dependent classifiers

Rank Based Methods

Borda Count

$$Q(x) = arg\max_{j=1}^{N} \left(B(j) = \sum_{i=1}^{K} B_i(j) \right)$$

$B_{i,j}(x)$ rank assigned by classifier i for class j given input x

Drawbacks: Does not consider information in the strengths of the preferences

Probabilistic Methods

Bayesian Combination

- c^i is the confusion matrix estimated on a training set for the ith classifier . Elements c^i_{jk} denotes the number of data points that are classified to be class k, whereas they are actually class j.
- The conditional probability that a sample x actually belongs to class j, given that classifier i assigns it to class k, can be estimated as

$$P\left(x \in q_j \mid \lambda_i(x) = j_i\right) = c^i_{jk} / \sum_{j=1}^{N} c_{jk} - i$$

- Assuming that the different classifiers are independent, a belief value that the input x belongs to class j can be approximated by

$$Bel(j) = \frac{\prod_{i=1}^{K} P(x \in q_j | \lambda_i(x) = j_i)}{\sum_{j=1}^{N} \prod_{i=1}^{K} P(x \in q_j | \lambda_i(x) = j_i)}$$

Probabilistic Methods (cont'd)

Dempster-Shafer Approach[Xu et. al. 92]

- Θ is a set of outcomes of an experiment
- $n(\Theta)$ is the number of elements in Θ.
- P is the set of propositions, or possible subsets, of Θ

A basic probability assignment (BPA) is assigned to each proposition or subset of Θ.
If $A \in P$ is a subset of Θ, then $BPA(A)$ represents the impact of the evidence (the output of the classifier) on A. From the BPA, a numeric value in the range [0,1] that indicates the belief in proposition A, denoted by $bel(A)$, is computed. The belief in A, $bel(A)$, indicates the degree to which the evidence or classifier output supports A and is given by

$$bel(A) = \sum_{B \subseteq A} BPA(A).$$

Dempster-Shafer Approach

If $A \in P$ is a subset of Θ which is not the null set, $BPA_1(A)$ is the BPA for one classifier, and $BPA_2(A)$ is the BPA for the other classifier, the combining rule is given by

$$BPA(A) = \frac{\sum_{C \cap D = A} BPA_1(C) BPA_2(D)}{1-k}$$

where $BPA(A)$ is the overall BPA after fusion $C \in P$, $D \in P$ and k is given by

$$k = \sum_{C \cap D = null} BPA_1(C) BPA_2(D)$$

Note that the classifier outputs are assumed to be independent. The BPAs for all $A \in P$ are found, and the beliefs $bel(A)$ are computed before proceeding to invoke the decision rule based on the beliefs. Note that if $k = 1$, the two evidences are in complete conflict, and $BPA(A)$ does not exist.

Fuzzy Integral Methods

Definition 1. A set function $g : 2^Z \rightarrow [0, 1]$ is a fuzzy measure if

- $g(0) = 0; g(Z) = 1$,
- if $A; B \subset 2^Z$ and $A \subset B$, then; $g(A) \leq g(B)$,
- if $A_n \subset 2^Z$ for $1 \leq n \leq \infty$ and the sequence $\{A_n\}$ is a monotone in the sense of inclusion, then $\lim_{n\rightarrow\infty} g(A_n) = g\left(\lim_{n\rightarrow\infty} A_n\right)$

In general, the fuzzy measure of a union of two disjoint subsets cannot be directly computed from the fuzzy measures of the subsets. Sugeno [Sugeno 77] has proposed the decomposable so called λ-fuzzy measure satisfying the following additional property:

$$g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B)$$

for all $A, B \subset Z$ and $A \cap B = 0$, and for some $\lambda > -1$.

Fuzzy Integral Methods (cont'd)

Let $Z = \{z_1, z_2, \cdots, z_K\}$ be a finite set (a set of committee members in our case), and let $g^i = g(\{z_i\})$. The values g^i are called the densities of the measure. The value of λ is found from the equation $g(Z) = 1$, which is equivalent to solving the following equation:

$$\lambda + 1 = \prod_{i=1}^{K}(1 + \lambda g^i).$$

When g is the λ-fuzzy measure, the values of $g(A_i)$ can be computed recursively as follows:

$$g(A_1) = g(\{z_1\}) = g^1$$

and

$$g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}), for 1 < i \leq K$$

Fuzzy Integral Methods (cont'd)

Definition 2. Let g be a fuzzy measure on Z. The *discrete Choquet integral* of a function $h : Z \rightarrow R^+$ with respect to g is defined as

$$C_g\{h(z_1), \cdots, h(z_K)\} = \sum_{i=1}^{K} \{h(z_i) - h(z_{i-1})\} g(A_i)$$

where indices i are permuted so that $0 \leq h(z_1) \leq \cdots \leq h(z_K) \leq 1$, $A_i = \{z_i, \cdots, z_K\}$, and $h(z_0) = 0$

Combining via Choquet Integral

Adopting Sugeno's λ-fuzzy measure and assigned the fuzzy densities g^i, that is, the degree of importance of each classifier, based on the performance of the classifier on validation data. The densities can be computed as follows:

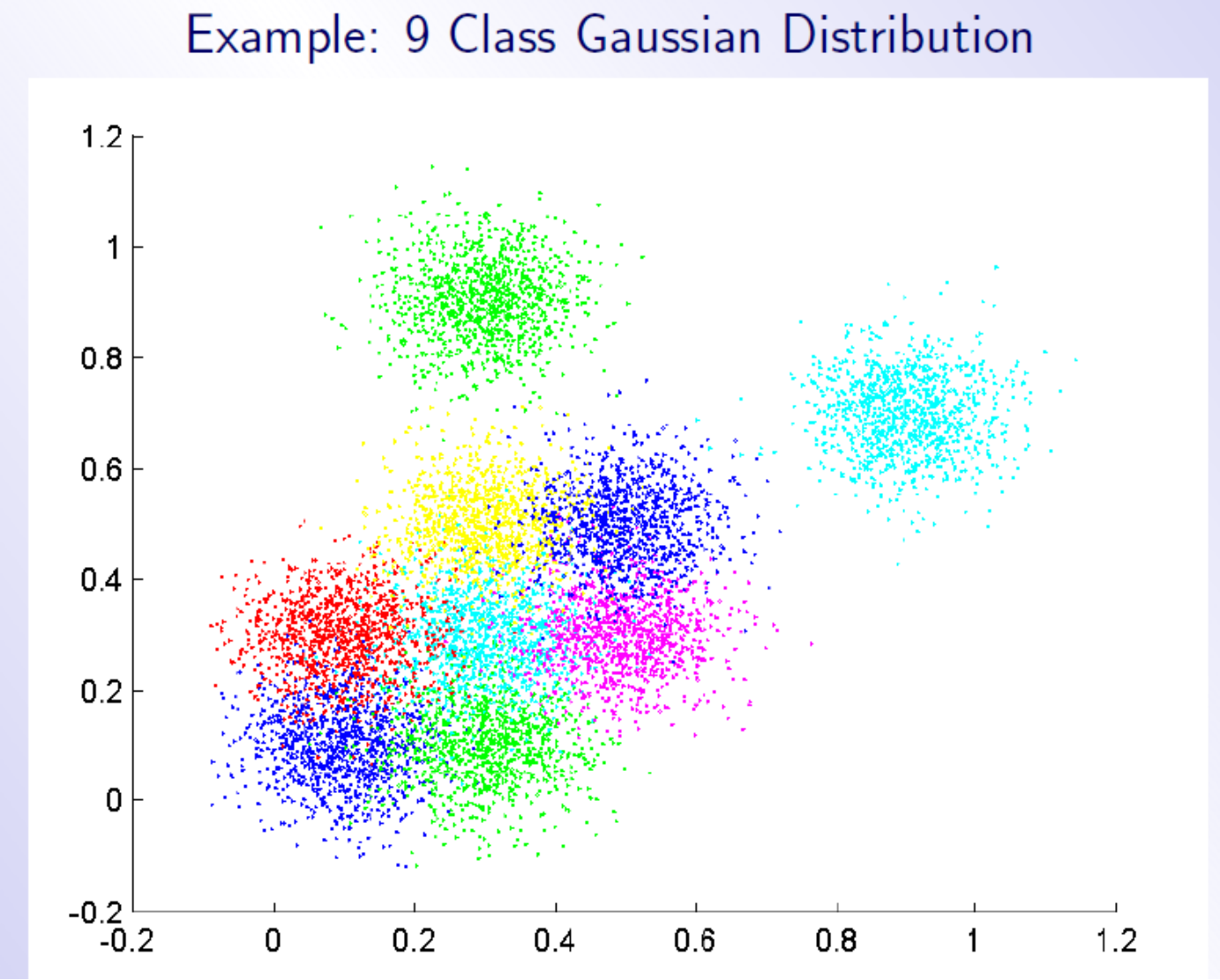
$$g^i = \frac{p_i}{\sum_{j=1}^{K} p_j} d_S$$

where p_i is the performance of the ith classifier, and d_S is the desired sum of the fuzzy densities. The committee members were assumed to have N outputs representing N classes, and data point x needs to be assigned to one of the classes. The class label Q for the data point x is then determined as

$$Q(x) = \arg \max_{j=1,\cdots,N} C_g(j)$$

where $C_g(q)$ is the Choquet integral for the class q. The values of function $h(z)$ that appear in the Choquet integral are given by the output values of the members of the committee (the evidence provided by the members).

Combining Classifiers (cont'd)



Combining Classifiers (cont'd)

Network	Training	Testing
Net1	79.09	79.31
Net2	82.04	81.85
Net3	76.49	76.73
Net4	76.31	75.51
Net5	80.33	80.47
Net6	71.40	70.71
Net7	80.69	80.56
Net8	82.87	82.67
Net9	78.58	78.42
Net10	77.22	76.78

Example

- Training 10 Neural Networks
  - 2 Hidden nodes
  - Learning Rate $= 0.1$
  - Momentum $= 0.5$
  - Epochs $= 1000$

Combining Classifiers (cont'd)

Example

Method	Training	Testing
Best Classifier	82.87	82.67
Average Vote	84.80	84.78
Weighted Average	85.29	85.09
Majority	84.33	83.87
Maximum	83.40	83.11
Product	11.11	11.11
Bayesian	85.67	85.76
Fuzzy	85.29	85.09

Combining Strategies

Static Combining All the methods present are static combining approaches, in the sense that the combiner decision rule is independent of the feature vector. Static approaches can be broadly divided into non-trainable and trainable

- **Non-trainable:** The voting is performed independently of the performance of each individual classifier Various combiners may be used, depending on the type of output produced by the classifier, including
  - **Voting:** used when each classifier produces a single class label. In this case, each classifier votes for a particular class, voting used to find a the winner.
  - **Averaging:** used when each classifier produces a confidence estimate. In this case, the winner is the class with the highest average posterior.
  - **Borda counts:** used when each classifier produces a rank. The Borda count of a class is the number of classes ranked below it.

Combining Strategies (cont'd)

- **Trainable:** The combiner undergoes a separate training phase to improve the performance of the ensemble machine. Trainable approaches include
  - **Weighted averaging:** the output of each classifier is weighted by a measure of its own performance.
  - **Fuzzy integral:** the output of each classifier is assigned a fuzzy density based on its own performance.

Combining Strategies (cont'd)

Adaptive Combining The combiner is a function that depends on the input feature vector. Thus, the ensemble implements a function that is local to each region in feature space

- This divide-and-conquer approach leads to modular ensembles where relatively simple classifiers specialize in different parts of the input-output space.
    - Note that, in contrast with static-combiner ensembles, the individual experts here do not need to perform well for all inputs, only in their region of expertise.
- Representative examples of this approach are Mixture of Experts (ME) and Hierarchical ME

Open issues

- General combination strategies are only sub-optimal solutions to most applications;

Multiple classifiers System: Challenges

- MCS is possible if individual classifiers make “different” errors
  - Combining identical classifiers is useless!

- How to create such systems and when they may perform better than their components used independently?

- Conclusions from some studies (e.g. Hansen&Salamon90, Ali&Pazzani96):Member classifiers should make uncorrelated errors with respect to one another; each classifier should perform better than a random guess.

Fuzzy Logic

Why Fuzzy Logic (FL) ?

- Conventional methods cannot deal with the imprecise representation of information
- FL deals with graded representation of classes
- FL allows an element to be a member of more than one category or class with graded membership values
- It works even for problems having insufficient information.

Existing Fuzzy Supervised Classification Methods

- Fuzzy *k*-nearest neighborhood (*k*-NN)
- Fuzzy maximum likelihood (FML)
- Fuzzy if-then rule (Fi-tR)
- Fuzzy explicit (FE)

RESULTS

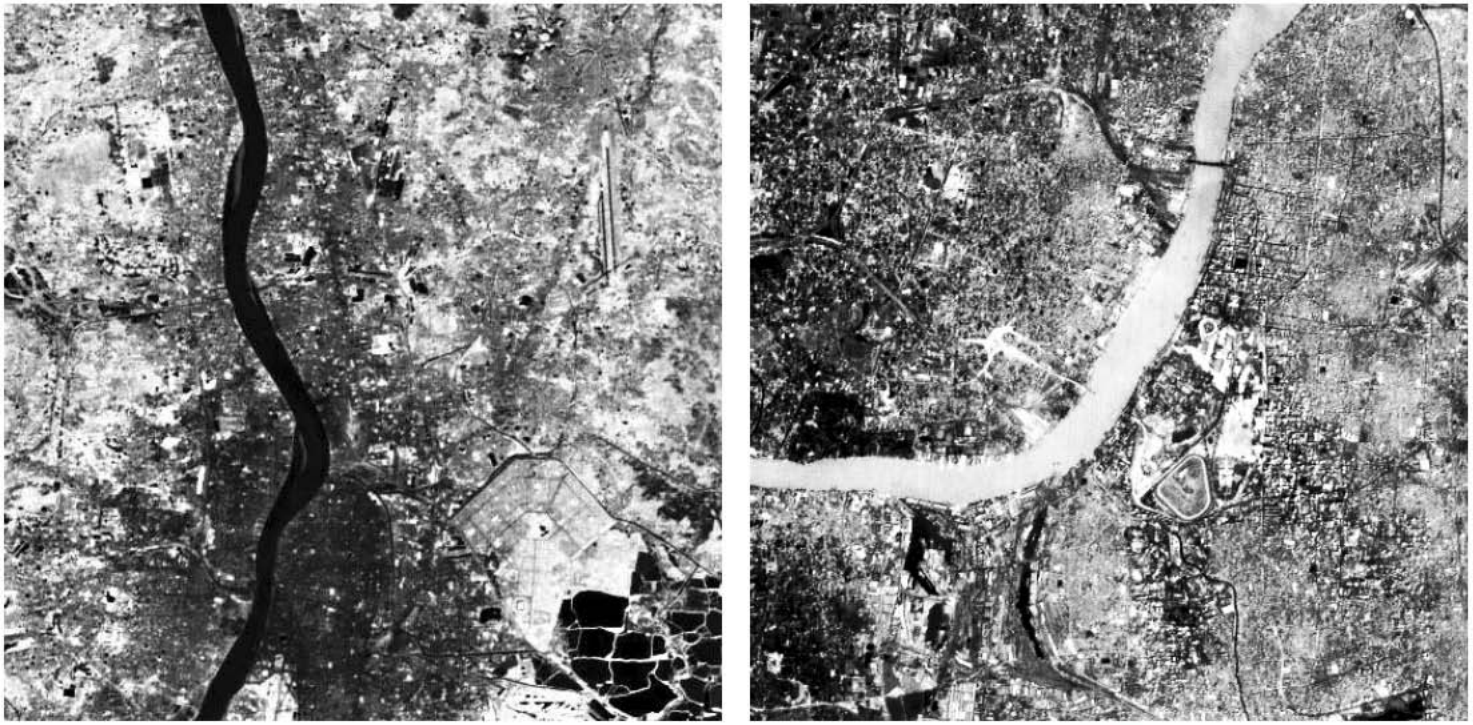
Classification method	*Image*			
	IRS-1A		*SPOT*	
	β	*XB*	β	*XB*
F*k*-NN	7.0121	0.9594	6.9212	2.5004
FMLC	7.0523	0.9356	6.9896	2.4231
FE	7.1312	0.9112	7.0137	2.3031
FPARR	*8.1717*	*0.8310*	*8.1078*	*2.1021*

RESULTS

Combination technique		*β index*		*XB index*	
		IRS-1A image	*SPOT image*	*IRS-1A image*	*SPOT image*
Voting		8.3134	8.2314	0.8211	2.1005
Fuzzy aggregation reasoning rule	*Maximum*	8.2787	8.3651	0.7903	2.1000
	Minimum	8.3213	8.5134	0.7879	1.9733
	Product	8.6217	*8.6321*	0.8003	2.0178
	Sum	8.4312	8.3781	0.8202	2.0013
	Mean	8.2013	8.2011	0.8201	1.9010
Probabilistic product		8.5011	8.6005	0.7983	1.9334
Fuzzy integral		8.5078	8.5017	*0.7710*	1.9768
Decision template		8.4032	8.5712	0.7801	*1.9001*
Dempster-Shafer		*8.6421*	8.5312	0.7781	1.9783
NFC		*8.8012*	*8.7763*	*0.7697*	*1.8738*

RESULTS

Figure 2 Original (a) IRS-1A (band-4) and (b) SPOT (band-3) image



(a) (b)

RESULTS

Figure 3 Classified (a) IRS-1A image and (b) SPOT image using proposed NF combiner