Recognition quality enhancement

Combination of multiple classifier decisions (meta-classifiers):

- Simple voting
- Weighted voting
- Combination of classifiers rank lists
- Naïve probability combination rule
- Bayesian combination rule (with confusion matrix)
- Behavior-Knowledge space

OCR postprocessing methods:

- Lexical methods (trigrams, dictionaries, etc.)
- Word context
- Document context

Ludmila I. Kuncheva, *Combining Pattern Classifiers. Methods and Algorithms*, Wiley Interscience, 2014 (2nd edition)

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Metaclassifiers

Application premises:

- Knowledge
  Generally, classifiers have regions in which their results are better then average recognition ration.
- Consensus
  Result of the democracy?
- Diversity
  It is easier to built “locally good” classifiers.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

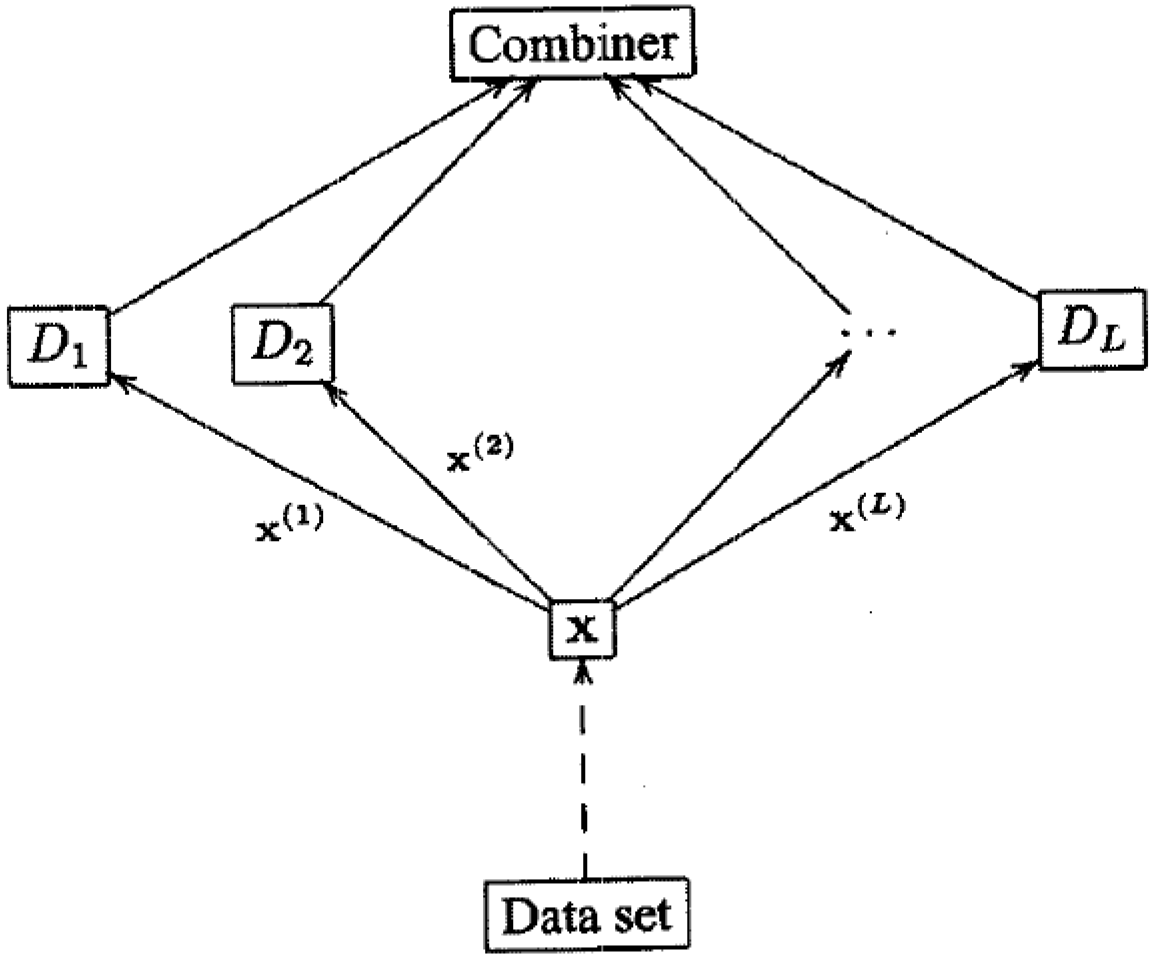
Metaclassifiers

There are four main aspects which the developer of a metaclassifier has to consider:

- Degree of integrity
  Degree of integrity serves as a measure of how strongly the individual classifiers are directed by the metaclassifier.
- Representation of classifier(s) results
  Results can contain different amounts of data: from a single label of one classification winner to a list of different candidate labels with their probability.
- Combination of classifier results
  A method chosen depends greatly on the representation of classifiers results, but also on time and means developer is able to spend on metaclassifier design.
- Methodological and representational restrictions.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Metaclassifiers



A. *Combination level:* Design different combiners.

B. *Classifier level:* Use different base classifiers.

C. *Feature level:* Use different feature subsets.

D. *Data level:* Use different data subsets.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Result representation

Generally classifier can output its result in one of the three forms:

1. **Abstract level**: Each classifier *e* provides a single result/label *j* without any additional information, i.e. $e(x)=j$.
2. **Rank level**: Each classifier provides an ordered list of ranked results, where the first element is the most likely one, i.e. $e(x)=L$, where *L* is **ordered** list of labels. A reject is denoted by $L=\emptyset$.
3. **Measurement level**: Each classifier produces alternatives along with a real value indicating the recognition confidence. The values need not be probability, but can also be distance measures to given reference patterns.

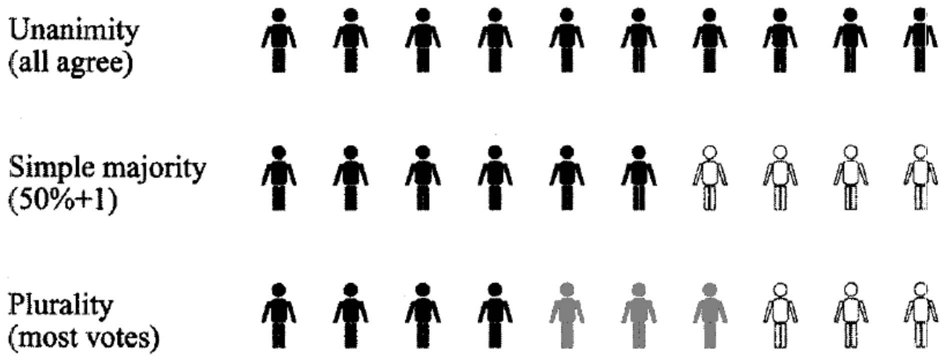
Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Combination of classifiers results

Abstraction level

- Simple voting

- Weighted voting



Rank level

- Class set reduction
- Class set reordering

Measurement level

- Bayes methods
- Other probabilistic frameworks

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Class set reduction

The goal of the reduction method is to extract subset of classes which hopefully captures the correct class. Thus, for class set reduction the success criterion is two-fold:

- The probability of inclusion of the true class in the result set.
- The size of the result set.

Two approaches are proposed for class set reduction:

- The intersection of the individual sets.
- The union of the individual sets.

Common to both approaches is the fact that they consider a neighborhood for each classifier (classes ranked from the top down to a certain specified rank position). According to the different approaches, the thresholds determining the neighborhood size for each classifier are estimated differently.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Class set reordering

In class set reordering, the objective is to derive a combined ranking of the given classes, such that the true class is ranked as close to the top as possible. The only success criterion for that approach is the rank position of the true class in the combined ranking.
The two common approaches for class set reordering are:

Highest Rank: The method simply assigns each class the best rank position of this class in any of the individual rankings. The main disadvantage of this method is that for a large number of classifiers many classes are ranked equally.

Borda Count: The method assigns a Borda count (for classifier *k*) $B_k(C_i)$ to every class C_i, which denotes the number of classes in P which are ranked below class C_i by classifier *k*.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Class set reordering

Borda Count (ctnd)

The individual classifiers' Borda counts are merged for every class C_i with: $B(C_i) = \sum_{k=1}^{K} B_k(C_i)$, where *K* is the number of classifiers. Subsequently all classes are ranked according to their Borda count values in decreasing order.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Very naïve *(non)probabilistic* version

The measurement level provides class decisions along with confidence values for these classes. The confidence values can be similarity as well as distance measures. The big problem is the transformation of different confidence measures utilized by different classifiers.
Let’s assume that all individual classifiers are Bayes classifiers, i.e. they approximate *a posteriori* probability $P(C_i \mid x)$ of object being class C_i given feature vector x. One straightforward way to combine the outputs of classifiers is to approximate the posteriors by averaging the measurements, i.e.

$$P(C_i \mid x) = \frac{1}{K} \sum_{k=1}^{K} P_k(C_i \mid x)$$

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Combining classification results

Let’s assume that we have a set of abstract level classifiers. How should we combine results of the individual classifiers to enhance recognition quality?
Note, that **better** recognition results, mean **lower** recognition ratio and at the same time **lower** error ratio with **greater** reject ratio.

Combination methods discussed further consist of:

- Majority voting,
- Weighted majority voting with weights derived from a genetic search algorithm,
- Bayesian formulation,
- Behaviour-Knowledge Space method.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Combining classification results

In real systems cost of the recognition error k_e and the cost of the reject decision k_r is generally different: in most cases $k_e >> k_r$.
Of course, we would like to decrease as much as possible both error ratio p_e and reject ratio p_r. In practice decreasing p_e results in increasing p_r.
To measure classifier performance – taking into account trade-off between recognition/rejection and error rates – we can define function with parameter β selected depending on the application area:

$$F = p_c - \beta * p_e$$

The aim of combining classifiers is maximizing this function.
Equivalent formulation uses function $F = p_r - (1+\beta) * p_e$; this function should be minimized.
Of course the problem is with selecting β value, which holds the proper relationship between p_r and p_e.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Genetic algorithm

A genetic algorithm begins with an initial *population* of randomly generated potential solutions to an optimization problem. The value of an objective function (*fitness function*) of each solution is evaluated, and the “best” solutions are selected for survival. Then the genetic algorithm manipulates these selected solutions in its search for better solutions.
Each solution is encoded into a binary string (*chromosome*), so that new encoded solutions can be generated through the exchange of information among surviving solutions (*crossovers*) as well as sporadic alterations in the bit string encodings of the solutions (*mutations*).

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Genetic algorithm

Population: 50
Gen size: 10 bits (weight from the range <0, 1023>)
Crossover probability: 0,9
Mutation probability: 0,05
Fitness function scaling:

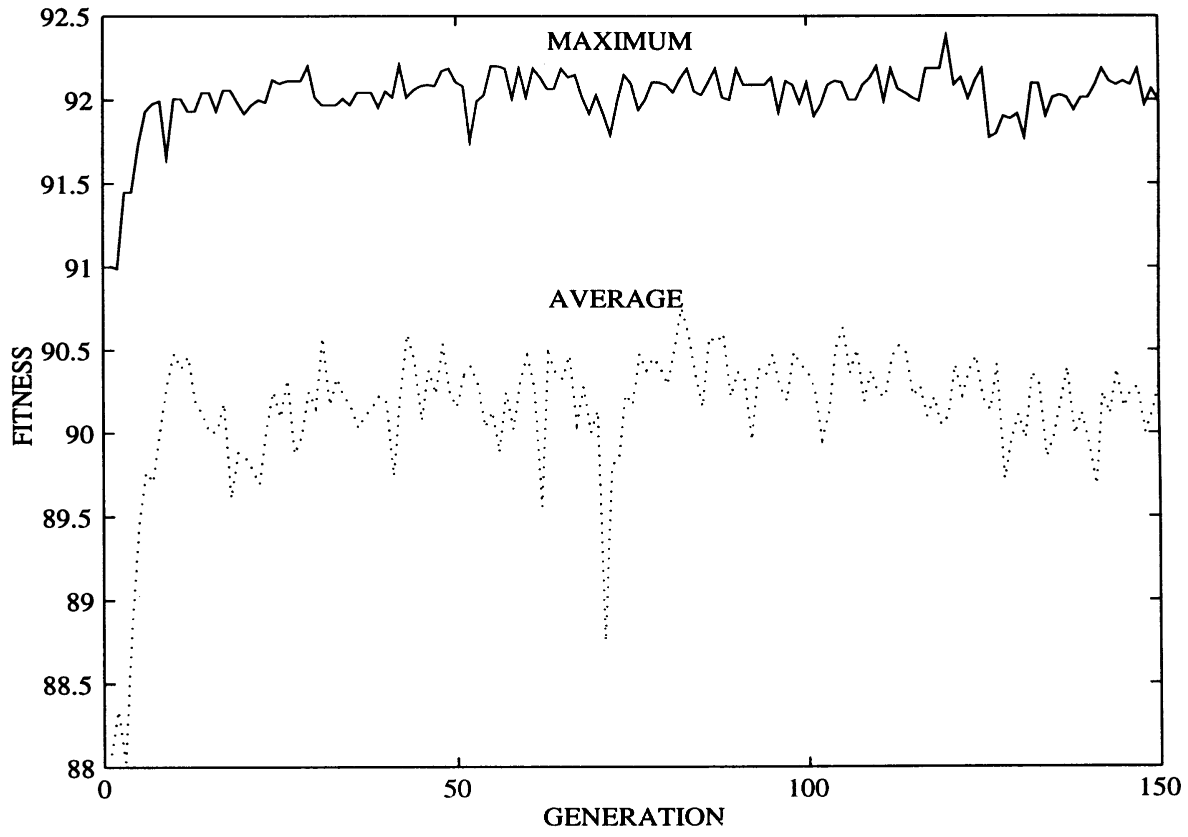
1. Scale is selected for the best fitness value f_{max}, such that $f_{max}=1,5*f_{avg}$ (f_{avg} is the average fitness in population)
2. Individuals which fitness function is below some threshold f_t are eliminated from evolution: $f_t=f_{avg}-2,5*\sigma$ (σ is standard deviation of fitness value in population).
3. Finally, scaled fitness function is given by:

$$f' = \frac{f - f_t}{f_{\max} - f_t}\left(1.5 f_{avg} - f_t\right) + f_t$$

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Genetic algorithm

After each generation weights were normalized to sum up to 1.



Maximum and average fitness values of genetic algorithm.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Bayesian combination rule

The genetic algorithm implemented assigns a weight to the vote of each classifier, and this weight would be applied to all patterns regardless of the decision mage by the classifier. Using Bayesian rule we can take into consideration the performance of each classifier on the training samples of each class. To represent the performance of a classifier the confusion matrix C is used. For a problem with M possible classes plus the reject option, C is an Mx(M+1) matrix in which the entry C_{ij} denotes the number of patterns with actual class i that is assigned class j by the classifier ($j \leq M$), and when $j=M+1$, it represents the number of patterns that are rejected.
Where there are K classifiers, there would be K confusion matrices C_k, $1 \leq k \leq K$.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Bayesian combination rule

Conditional probability, that a pattern *x* actually belongs to class *i*, given that classifier *k* assigns it to class *j*, can be estimated as:

$$P\left(x \in C_i \mid e_k(x) = j\right) = \frac{C_{ij}^k}{\sum_{i=1}^{M} C_{ij}^k}$$

Combining this for *K* classifiers, we define belief function:

$$bel(i) = P\left(x \in C_i \mid e_1(x) = j_1, \ldots, e_K(x) = j_K\right)$$

By applying the Bayes formula and assuming independence of the classifier decisions we can approximate belief function as:

$$bel(i) \doteq \frac{\prod_{k=1}^{K} P\left(x \in C_i \mid e_k(x) = j_k\right)}{\sum_{i=1}^{M} \prod_{k=1}^{K} P\left(x \in C_i \mid e_k(x) = j_k\right)}$$

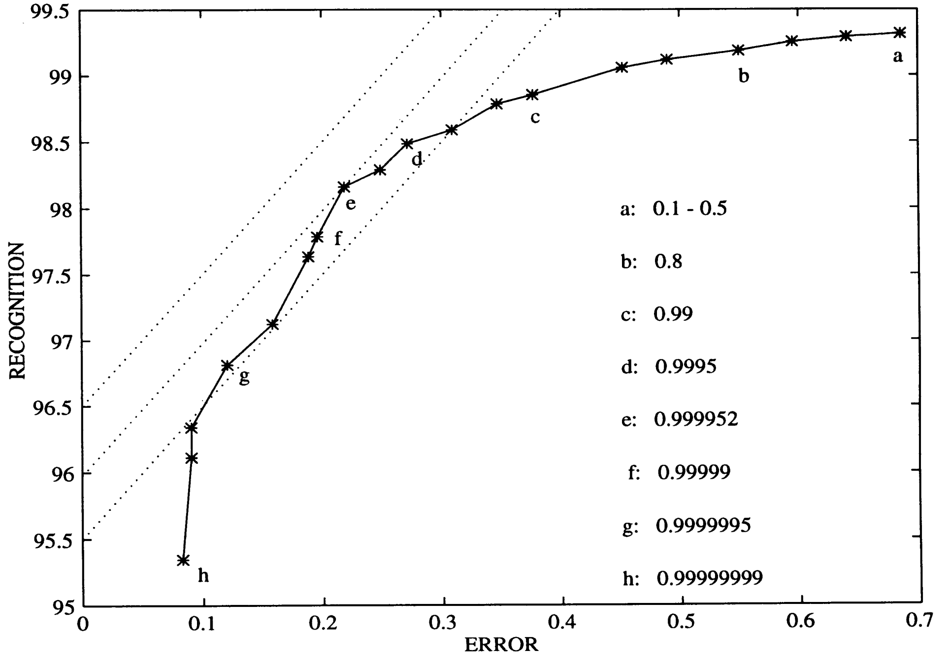
Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Bayesian combination rule

For any input pattern *x*, we can assign *x* to class *j* if:

$$\underset{i \neq j}{\forall}\, bel(j) \geq bel(i) \quad \text{and} \quad bel(j) > \alpha$$

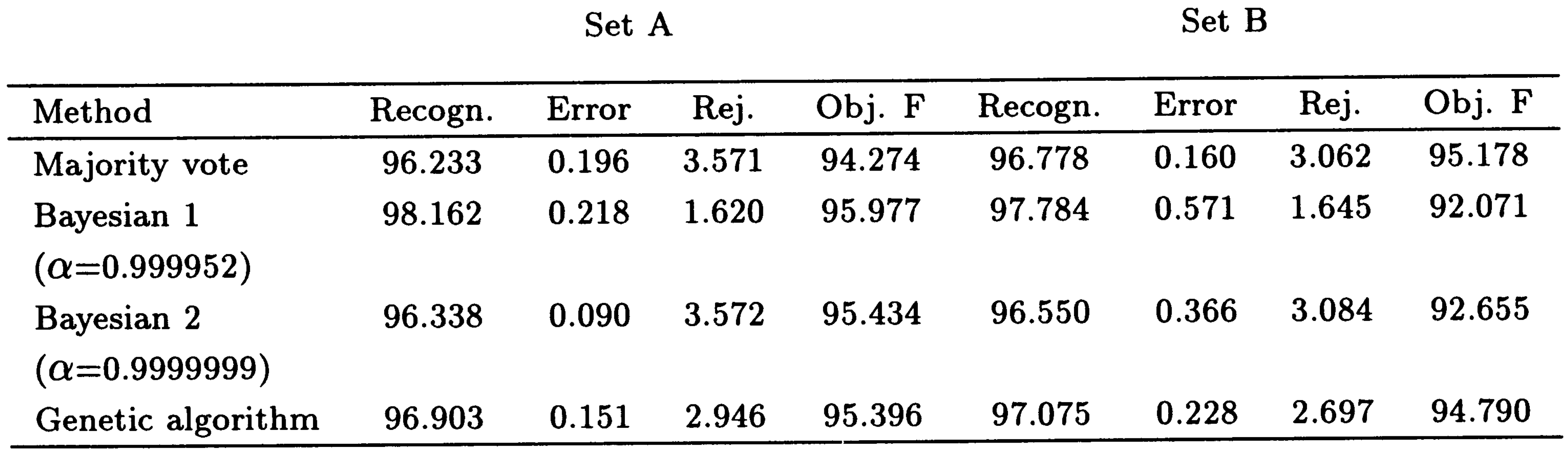
Otherwise *x* is rejected, and it also rejected if for all *k* $e_k(x) = M + 1$ (i.e. if *x* is rejected bye all classifiers).



Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Experiment results

Experiment was performed on 7 classifiers recognizing handwritten digits. The set consists of 46451 digits, divided into training (for classifiers) and testing subsets. Testing set was again divided into the training (A) and testing (B) subsets. Subset A was used to compute weights.



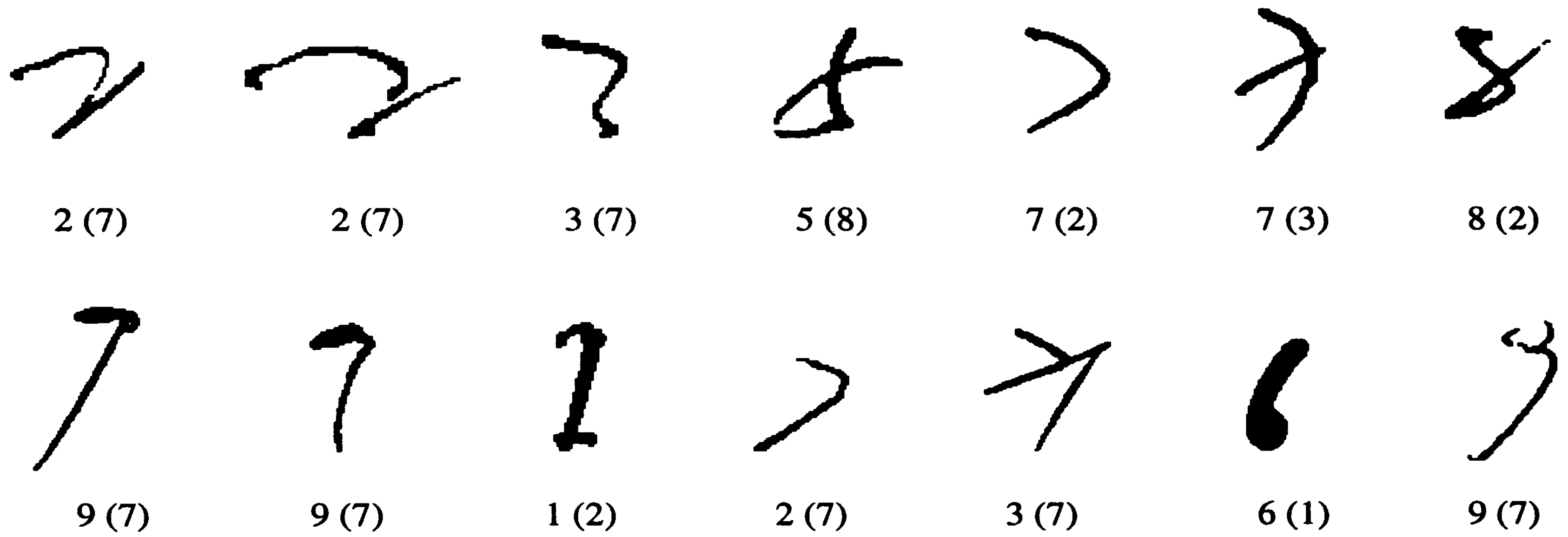
	Set A				Set B			
Method	Recogn.	Error	Rej.	Obj. F	Recogn.	Error	Rej.	Obj. F
Majority vote	96.233	0.196	3.571	94.274	96.778	0.160	3.062	95.178
Bayesian 1 (α=0.999952)	98.162	0.218	1.620	95.977	97.784	0.571	1.645	92.071
Bayesian 2 (α=0.9999999)	96.338	0.090	3.572	95.434	96.550	0.366	3.084	92.655
Genetic algorithm	96.903	0.151	2.946	95.396	97.075	0.228	2.697	94.790

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Conclusions

1. Simple majority voting is best on set B. This suggests that A set is too small and in weighted voting methods we can see overfitting results.
2. Size of set A is definitely too small for Bayesian combination rule. The source of the problem is very small number of samples lying off the main diagonal of the confusion matrix. What's more, the better the classifier the smaller part of the training samples will miss the main diagonal.
3. Since the genetic algorithm computes fewer parameters (7 weights) it is less sensitive to differences in data characteristics in sets *A* and *B*. At the same time genetic algorithm does not assume classifier independence.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds



Patterns misclassified by all methods.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Behavior-Knowledge space

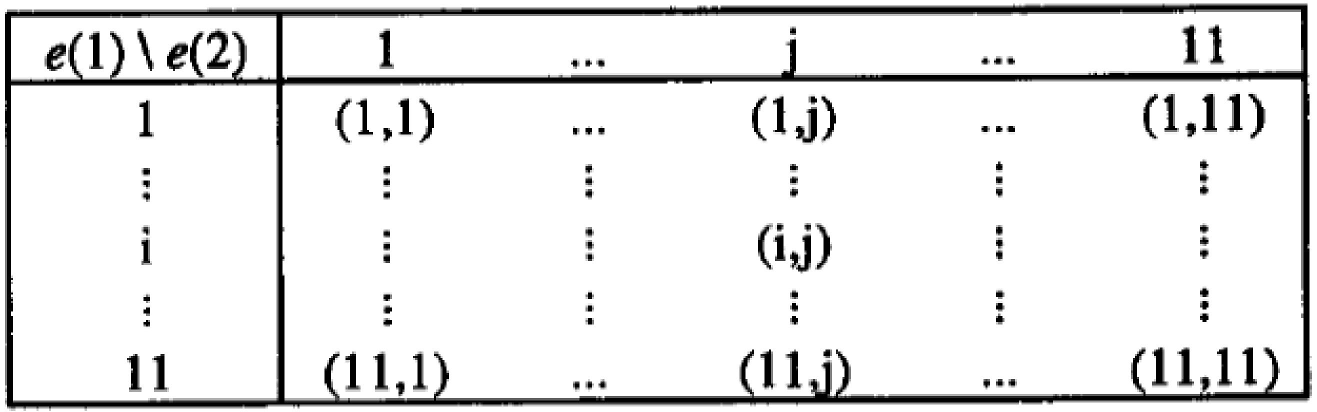
The main problem with most abstract level combination methods (Bayesian, but also majority vote!) is that they require classifier independency. It is because they treat each classifier equally, or they derive probabilities from the confusion matrix of a single classifier.

To avoid independence assumption, the information should be derived from a knowledge space which can *concurrently* record the decisions of all classifiers on each learned sample. Since this knowledge space records the behavior of all classifiers it is called “Behavior-Knowledge Space”. Classifier results create K-dimensional space, in which each dimension corresponds to the descision of one classifier. Each of the dimensions has size M+1. The cell which is the intersection of the classifiers’ decisions of the current input is the *focal* cell.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Behavior-Knowledge space

2-D BEHAVIOR-KNOWLEDGE SPACE



$e(1) \setminus e(2)$	1	...	j	...	11
1	(1,1)	...	(1,j)	...	(1,11)
⋮	⋮	⋮	⋮	⋮	⋮
i	⋮	⋮	(i,j)	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
11	(11,1)	...	(11,j)	...	(11,11)

Each of the cells stores:

1. The total number of incoming samples: $T_{e(1)..e(K)}$
2. The total number of incoming samples belonging to class m: $n_{e(1)..e(K)}(m)$
3. The best representative class of a cell: $R_{e(1)..e(K)}$

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Operation of BKS

The knowledge-modeling stage uses the learning set of samples with both genuine and recognized class labels to construct BKS (compute $n_{e(1)..e(K)}(m)$ for all the samples); then the values of $T_{e(1)..e(K)}$ and $R_{e(1)..e(K)}$ are computed for each cell.

The decision-making stage, according to the constructed BKS and the decisions offered from the individual classifiers, enters the focal cell and makes the final decision by the following rule:

$$E(x) = \begin{cases} R_{e(1)\ldots e(K)}, & dla\ T_{e(1)\ldots e(K)} > 0\ i\ \dfrac{n_{e(1)\ldots e(K)}\left(R_{e(1)\ldots e(K)}\right)}{T_{e(1)\ldots e(K)}} \geq \lambda \\ M+1, & wpp \end{cases}$$

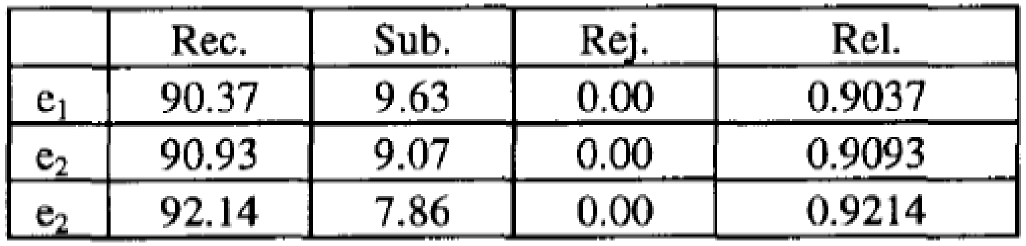
Threshold λ (in the range 0-1) controls the reliability of the final decision.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Results

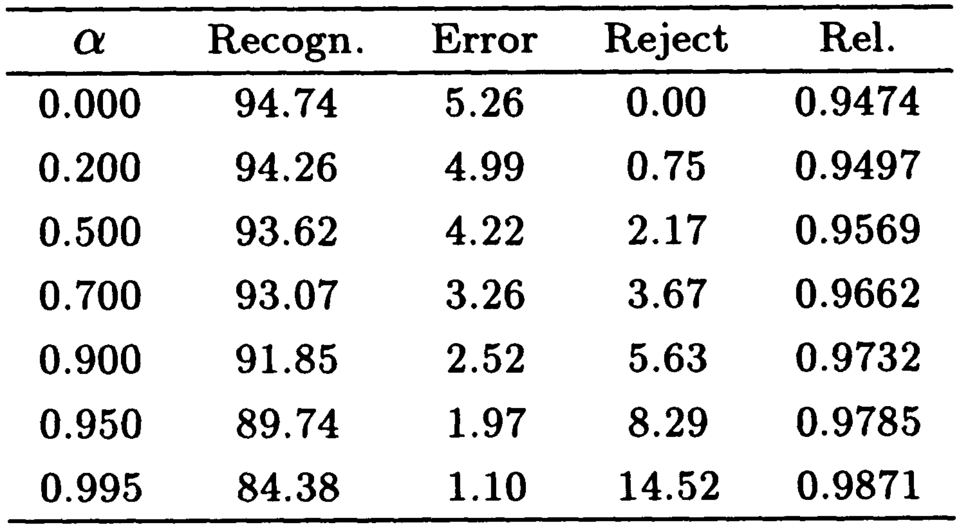
There were 3 digit classifiers used. ~5k digits were used to train classifiers, ~40k digits form the test set.

Classifiers' coefficients:



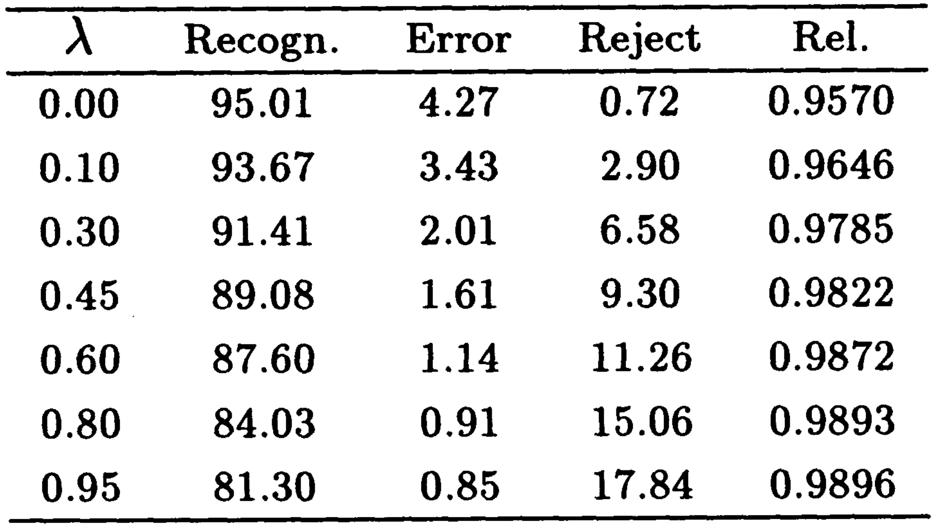
	Rec.	Sub.	Rej.	Rel.
e_1	90.37	9.63	0.00	0.9037
e_2	90.93	9.07	0.00	0.9093
e_2	92.14	7.86	0.00	0.9214

Bayesian rule results



α	Recogn.	Error	Reject	Rel.
0.000	94.74	5.26	0.00	0.9474
0.200	94.26	4.99	0.75	0.9497
0.500	93.62	4.22	2.17	0.9569
0.700	93.07	3.26	3.67	0.9662
0.900	91.85	2.52	5.63	0.9732
0.950	89.74	1.97	8.29	0.9785
0.995	84.38	1.10	14.52	0.9871

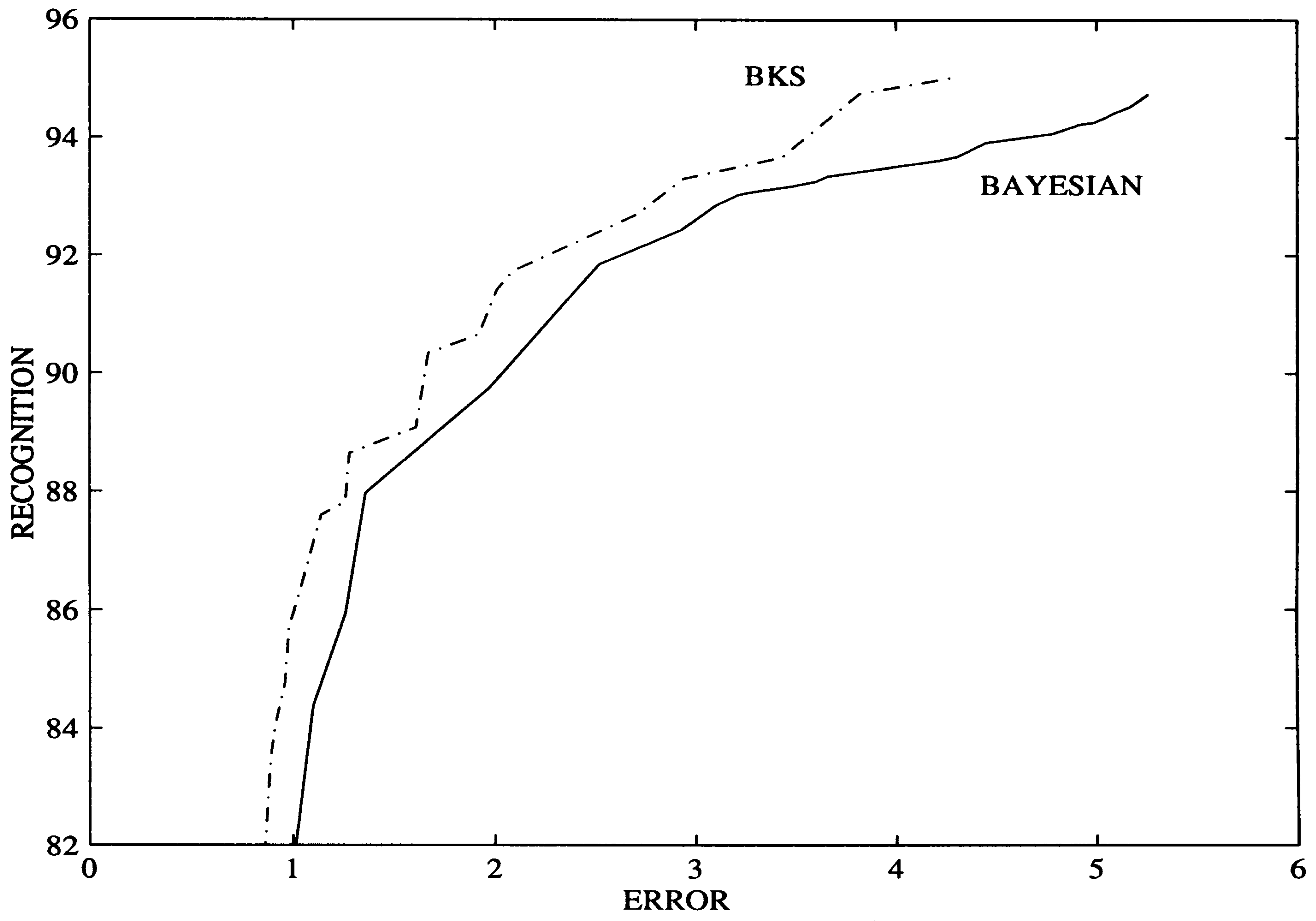
BKS results



λ	Recogn.	Error	Reject	Rel.
0.00	95.01	4.27	0.72	0.9570
0.10	93.67	3.43	2.90	0.9646
0.30	91.41	2.01	6.58	0.9785
0.45	89.08	1.61	9.30	0.9822
0.60	87.60	1.14	11.26	0.9872
0.80	84.03	0.91	15.06	0.9893
0.95	81.30	0.85	17.84	0.9896

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Results



Performance of Bayesian and BKS combination methods.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Lexical post-processing

In general, lexical post-processing has the task of verifying OCR results using lexical knowledge and of generating a ranked list of possible word candidates when the input is noisy. This ranking process includes some distance or similarity measure between the misspelled input word and the respective word candidates.

Lexical information can be represented by:

- N-grams
- N-grams with probability
- Dictionaries of allowable values (complete or not)

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds