Hierarchical Behavior Knowledge Space

Hubert Cecotti and Abdel Belaïd

READ Group , LORIA/CNRS
Campus Scientifique BP 239,
54506 Vandoeuvre-les-Nancy cedex France
{cecotti;abelaid}@loria.fr

Abstract. In this paper we present a new method for fusing classifiers output for problems with a number of classes $M > 2$. We extend the well-known Behavior Knowledge Space method with a hierarchical approach of the different cells. We propose to add the ranking information of the classifiers output for the combination. Each cell can be divided into new sub-spaces in order to solve ambiguities. We show that this method allows a better control of the rejection, without using new classifiers for the empty cells. This method has been applied on a set of classifiers created by bagging. It has been successfully tested on handwritten character recognition allowing better-detailed results. The technique has been compared with other classical combination methods.

1 Introduction

In any recognition system, an optimal reliability is one of the main requirements. In order to obtain such high reliability, the system must be able to consider the rejection. We distinguish three main ways to build a system able to reject:

- A single classifier, which also considered a rejection class (trained with junk patterns for example).
- A single classifier, which does not consider a rejection class, but uses some rejection rules for rejecting or not the results.
- A multi-classifiers system (MCS), where the rejection is processed by the module that fuses each classifier output.

Among these solutions, we consider in this paper the third solution for several reasons. We consider that for a high reliability several classifiers must take part into the recognition, to have several points of view of the problem. In this work, we will consider a specific MCS type: MCS with a parallel topology. In this case, the outputs of each classifier are combined thanks to a fusing module [**?**,**?**]. Usually, classifiers are combined by voting methods, belief functions, statistical techniques or Dempster-Shafer evidence theory. We distinguish several types of fusing rules:

- Fixed rules: majority voting, Borda count method... These rules are usually simple, fast and they are well-suite for classifiers ensembles that have similar performances and low correlated errors.

- Trained rules: Bayesian, behavior knowledge space, neural network,... These rules are potentially better than the fixed rules as they use knowledge about how to combine. These rules allow taking more into account the complementarities between classifiers.

This paper will focus on the behavior knowledge space method and its potential issues [**?**,**?**]. We will show that it is possible to improve this method by adding information about the ranking output of each classifier. Accordingly, in the first part, the initial fusing rule using the behavior knowledge space will be defined. Then the hierarchical behavior knowledge space is described in the second section. The third section will present the different classifiers. Finally, we exhibit the improvement given by the method with several experiments.

2 Behavior Knowledge Space

We consider a problem with M classes: C_i, $1 \leq i \leq M$ and D classifiers: e_i, $1 \leq i \leq D$.

For applying the Bayes rule, classifiers must be independent. Each classifier must act separately in a total independent way. This condition cannot be always verified. The method using the history of the classifiers behavior allows getting free from this condition. The BKS (Behavior-Knowledge Space) method allows to determine a belief degree of a proposition $x \in C_i$ based on the combination of the first best answer of each classifier $e_k = j_k, k \in \{1, \ldots, D\}$:

$$bel(C_i) = \frac{P(e_1(x) = j_1, \ldots, e_D(x) = j_D, x \in C_i)}{P(e_1(x) = j_1, \ldots, e_D(x) = j_D)}$$

This equation corresponds to the degree of belief definition by a Bayesian approach. It can be represented in a behavior knowledge space (BKS) [**?**]. This space represents the behavior for all the possible combinations in the training database. The BKS is a D dimensional space where each dimension represents the decision of a classifier. Each classifier has $M+1$ possible outputs (M classes and 1 rejection class). The intersection of the decision of each classifier corresponds to a cell of the BKS. This method will estimate M^D posterior probabilities.

Each cell of the space is noted by $BKS(j_1, j_2, \ldots, j_D)$ with $j_i \in \{1, \ldots, M+1\}$ $\forall i \in \{1, \ldots, D\}$. Each cell of the BKS is defined by 3 features:

- $n(j_1, \ldots, j_D)(i)$: the total number of samples x such that $e_1(x) = j_1, \ldots, e_D(x) = j_D$ and $x \in C_i$ $i \in \{1, \ldots, M\}$.
- $S(j_1, \ldots, j_D)$: the total number of samples x such that $e_1(x) = j_1, \ldots, e_D(x) = j_D$ and

  $$S(j_1, \ldots, j_D) = \sum_{i=1}^{M} (n(j1, \ldots, jD)(i))$$

  $S(j_1, \ldots, j_D)$ corresponds to the total sum of the samples that have as combination result the configuration $j1, \ldots, jD$.

- $B_{j1,\ldots,jD}$ is the best representative class of the cell $j_1,\ldots,j_D$ of the BKS then:
$$n(j_1,\ldots,j_D)(B(j_1,\ldots,j_D)) = max_i(n(j_1,\ldots,j_D)(i))$$
$$B(j_1,\ldots,j_D) = Argmax_i(n(j_1,\ldots,j_D)(i))$$
with $i \in \{1,\ldots,M\}$.

In an implementation view, the BKS space can be represented by a space BKS' of dimension $D+1$, where D dimensions correspond to the classifiers outputs and the last dimension represents the final optimal result of the combination. The BKS' is a space where all the extracted results are represented. In this case, each cell of $BKS'(j_1,\ldots,j_{D+1})$ is a natural positive number such that:
$$BKS'(j_1,\ldots,j_{D+1}) = n(j_1,\ldots,j_D)(j_{D+1})$$

The degree of belief that a sample x belongs to the class C_i denoted by $bel(C_i)$ $i \in \{1,\ldots,M\}$ is defined by:

$$\begin{aligned} bel(C_i) &= \frac{P(e_1(x)=j_1,\ldots,e_D(x)=j_D, x\in C_i)}{P(e_1(x)=j_1,\ldots,e_D(x)=j_D)} \\ &= \frac{BKS'(j_1,\ldots,j_D,i)}{\sum_{k=1}^{M} BKS'(j_1,\ldots,j_D,k)} \\ &= \frac{n(j_1,\ldots,j_D)(i)}{S(j_1,\ldots,j_D)} \end{aligned}$$

Finally, the combination E of the classifiers will give to the input x the following class:

$$E(x) = \begin{cases} R(j_1,\ldots,j_D) & if\ (S(j_1,\ldots,j_D) > 0) \\ & and\ (bel(C_{R_{j1,\ldots,jD}}) \geq \alpha) \\ M+1 & else \end{cases}$$

where α is a rejection threshold; $0 \leq \alpha \leq 1$.

During the test phase, it is possible to access to an empty cell. In this case, it means that the classifiers output combination has been never seen during the creation of the space. The input x is rejected.

In a statistical point of view, the BKS method tries to estimate the probability distribution of the classifiers outputs thanks to the frequencies of its occurrences. Although the BKS does not require a special dependency between each classifier, several observations can be made for this method.

3 Possible improvements

The BKS method suffers of several defaults [**?**]:

- The size of the database is one issue. In order to estimate the distribution of each classifier output, a large database is needed for representing all the possible combinations. However, this observation is mostly valid for weak classifiers that cannot offer a good recognition. For strong classifiers, all the non-empty cells are expected to stay close to the diagonal of the behavior space. Weak classifiers may lead to a better generalization thanks to their cover of the input space, but they will require much more samples to fill the space. Because of the statistical nature of the BKS, the quality of the database is very important for obtaining a good generalization.
- The confusion of BKS cells where the representative class R has a very low probability. If such cell exists, the result remains ambiguous. Although this cell will propose the best solution, many patterns will obtain a bad class. For the training database, in each cell, $S(j_1, \ldots, j_D) - max_i(n(j_1, \ldots, j_D)(i))$ samples, with $i \in \{1, \ldots, M\}$, will not be recognized correctly. In order to solve this problem, it is possible to add a new classifier specialized for dealing with the confusion problem involved by the ambiguous cell. Instead of using such process, we propose to extract more knowledge contained in the classifiers outputs: the ranking.

In the BKS method, only the first result of each classifier is considered during the combination. The confidence value and the ranking of the different classes are unfortunately not considered. We propose to use more information in order to improve the description of the ideal combination.

4 Hierarchical Behavior Knowledge Space

The Hierarchical Behavior Knowledge Space is based on the hypothesis that the ranking of the different classes for each classifier may bring relevant information for improving the quality of the combination. During the creation of the behavior space, the only information is the first best answer. The addition to new information to the space will lead to the creation of new cells. The $HBKS$ is strongly equivalent to the BKS space for 2 classes. Indeed, for a two classes problem, with C_0 and C_1, we have $P(C_0) = 1 - P(C_1)$. Thus there is no information in the second best answer as it is dependent of the first. For taking advantage of the cell splitting, we must have $M > 2$. The new space becomes a tree of sub-space. The root of space is defined by the initial BKS. For each cell $j_1, \ldots, j_D$ if $bel(C_{R_{j1,\ldots,jD}}) \geq \alpha$ then the cell is split into $(M - i + 1)^D$ cells where i is the actual rank of the cell.

Each cell of the space is noted by $HBKS((j_{1,1}, \ldots, j_{1,k}), \ldots, (j_{D,1}, \ldots, j_{D,k}))$ with $j_{i,k} \in \{1, \ldots, M+1\}$ $\forall (i, k) \in (\{1, \ldots, D\} \times \{1, \ldots, M-1\})$. k is the rank of the output.

For the following definition, we note by J the cell
$(j_{1,1}, \ldots, j_{1,k}), \ldots, (j_{D,1}, \ldots, j_{D,k})$

Each cell of the $HBKS$ is defined by 3 features:

- $n'(J)(i)$: the total number of samples x such that the best answer of $e_d(x)$ is $j_{d,1}$, the k^{th} best answer of $e_d(x)$ is $j_{d,1}$ with $d \in \{1, \dots, D\}$ and $x \in C_i$ $i \in \{1, \dots, M\}$. We note $e_{d,k}(x)$ the k^{th} best answer of $e_d(x)$.
- $S'(J)$: the total number of samples x such that $e_{d,1}(x)$ is $j_{d,1}$ and $e_{d,k}(x)$ is $j_{d,k}$.
$$S'(J) = \sum_{i=1}^{M} (n'(J)(i))$$
$S'(J)$ corresponds to the total sum of the samples that have as outputs the configuration J.
- $B'(J)$ is the best representative class of the cell J of the $HBKS$ then:
$$n'(J)(B'(J)) = max_{(i \in \{1,\dots,M\})}(n'(J)(i))$$
$$B'(J) = Argmax_{(i \in \{1,\dots,M\})}(n'(J)(i))$$

The creation of such sub-spaces can denoise the initial cells and discover evidence of confusion between classes. For example, the rejection of a cell can be due to the noise of the different outputs. It is the case where always the same 2 classes are confused and the combination can solve the ambiguity. If a couple of classes is confused, which have never happened, the creation of sub-spaces in the BKS may solve this problem.

$$E(x) = \begin{cases} R'(j_1, \dots, j_D) \; if \; (S'(j_1, \dots, j_D) > 0) \\ \qquad\qquad\quad\;\; and \; (bel(C_{R'_{j_1,\dots,j_D}}) \geq \alpha) \\ Split \; the \; cell \quad else \end{cases}$$

$$\begin{aligned} bel(C_i) &= \frac{P(e_1(x) = j_{1,1}, \dots, e_D(x) = j_D, x \in C_i)}{P(e_1(x) = j_1, e_2(x) = j_2, \dots, e_D(x) = j_D)} \\ &= \frac{BKS'(j_1, \dots, j_D, i)}{\sum_{k=1}^{M} BKS'(j_1, \dots, j_D, k)} \\ &= \frac{n(j_1, \dots, j_D)(i)}{S(j_1, \dots, j_D)} \end{aligned}$$

The number of cells of the $HBKS$ is defined by:

$$M^D + \sum_{k=1}^{M-1} l_k((M-k)^D)$$

where l_k is the number of new sub-spaces at the step k or the number of ambiguous cells at the step $k-1$.

The maximum number of cells of the $HBKS$ is defined by:

$$M^D + \sum_{k=1}^{M-1} (M-k+1)^D * (M-k)^D - (M-k+1)^D$$

When a sub-space is created for a cell, the sub-space replaces its corresponding cell.

The creation of the $HBKS$ can be built this way:

```
$k \leftarrow 1$
Fill the HBKS with the training database
While $(\exists J|((S'(J) > 0)\ and\ (bel(C_{R'(J)}) \leq \alpha))$
{
   For all $J|((S'(J) > 0)\ and\ (bel(C_{R'(J)}) \leq \alpha))$
      Create a sub-space for the cell $J$
   Fill the new HBKS sub-spaces with the training database
   $k \leftarrow k + 1$
}
```

The table **??** presents an example for a problem with 2 classifiers and 3 classes (A,B and C). Each cell of the table represents $n(j_1, \ldots, j_D)(i)$. In the BKS cell AB, there is an ambiguity between the answers A and B. In the classical BKS, this cell could have been considered as being too ambiguous. The table **??** shows the subspace involved by the split of the cell AB. In these new cells, some cells remain ambiguous (c1,c4) but for some others the problem may be solved (c2,c3).

Table 1. Example of some cells in a BKS.

top 1	AA	AB	AC	BA	BB	BC	CA	CB	CC
A	90	50	80	40	9	30	20	0	0
B	5	51	12	60	80	30	20	30	0
C	5	0	8	0	11	40	60	70	100

Table 2. Example for a subspace in a HBKS.

cell	c1	c2	c3	c4
top 1	AB	AB	AB	AB
top 2	BA	BC	CA	CA
A	21	20	5	1
B	20	2	20	2
C	0	4	3	2

5 Classifiers

In this section, the different classifiers used for the combination are described. They are based on the same architecture: a convolutional neural network. This type of classifier has been already successfully used on handwritten digits recognition and word recognition [?].

5.1 Convolutional Neural Network

The neural network used is composed of 5 layers, it is based on the topology given in [?]:

- The first one corresponds to the input image. The image is normalized by its center and reduced to a size of 29*29 [?].
- The next two layers corresponds to the information extraction, performed by convolutions. The second layer is composed of 10 maps, each one corresponds to a specific image transformation by convolution and sub-sampling reducing its size. The third layer is composed of 50 maps. For these 2 layers, the activation function is $f(\sigma) = 1.7159 * tanh((2.0/3.0) * \sigma)$ [?]
- The last two layers are fully connected. For these 2 layers, the activation function is $f(\sigma) = 1/(1 + exp(-\sigma))$.
- The last one corresponds to the output: 10 neurons, for the number of classes.

For a neuron n, weights are initialized with values w such that $|w| \leq 1/\sqrt{N_{input}}$ where N_{input} represents the number of inputs to the neuron n. During the back propagation, shared weights are corrected by the factor $2/\sqrt{N_{share}}$ where N_{share} represents the number of neurons that share the same set of weights.

5.2 Creation of the classifiers

Each classifier is built on the same architecture, described previously. D classifiers are created, each classifier being trained on a different versions of the initial database. For creating the ensemble of classifiers, we did use the bagging technique [?]. This method is based on obtaining different training sets of equal size as the original one, by using the statistical bootstrap method.

6 Experiments

6.1 Database description

The system has been tested on the MNIST handwritten digits database [?]. This database contains separated handwritten digit images of $28 * 28$ in gray level. The learning set contains 60000 images and the test set contains 10000 images. In the learning set, 50000 images are used for real learning; 10000 images are used to find the best parameters. For the experiments, 3 classifiers have been created. Each one is trained with 33151 images.

6.2 Results

The results obtained on each classifier are presented in the table **??**. We present the best result on the test database and the results obtained with the same network on the training database. For a single classifier or MCS, the results are defined by a triplet $\tau_r/\tau_s/\tau_q$ where τ_r, τ_s and τ_q are the recognition rate, the error rate and the rejection rate respectively.

The results obtained on the whole training database and test database are presented in the table **??**. For each classifier, the results correspond to the network that gives the best result on the validation database. Although these classifiers do not offer the best results on this database function to the state-of-the-art [**?**,**?**], they still provide all a high accuracy. The table **??** illustrates the results on the test database for the different tops. The good result is almost always in the first three best answers, which justifies the choice of our approach for the problem.

Table 3. Results.

	Training	Test
C1	99.28 / 0.72 / 0	98.51 / 1.49 / 0
C2	99.32 / 0.68 / 0	98.56 / 1.44 / 0
C3	99.27 / 0.73 / 0	98.61 / 1.39 / 0

Table 4. Recognition rate.

	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6
C1	98.51	99.63	99.83	99.92	99.99	100
C2	98.56	99.63	99.90	99.94	99.97	99.98
C3	98.61	99.58	99.89	99.96	99.97	99.99

The 3 classifiers have been combined with classical fixed rules:

- The Majority Voting; 2 classifiers must agree to accept the answer.
- The Oracle illustrates the result of an optimal output selection.
- The Maximum rule.
- The combination by outputs sums.
- The combination by outputs products.
- The Borda Count method, which takes into account the outputs ranking [**?**,**?**].

The results of these combinations on the test database are shown in the table **??**. The Borda Count method, which uses rank-level information, gives one of the best results. It is again a proof for considering the ranking during the combination for our problem.

Table 5. Combination results.

	MNIST Test
Majority voting	98.64 / 1.29 / 0.07
Oracle	99.13 / 0.87 / 0
Max	98.69 / 1.31 / 0
Sum	98.68 / 1.32 / 0
Product	98.64 / 1.36 / 0
Borda Count	98.66 / 1.34 / 0
BKS	98.45 / 1.41 / 0.14

For the learning database, we did observe that the $HBKS$ is an optimal combination: each sub-spaces lead to the good answer. It may lead to the creation of a sub-space with only one pattern in the space. Such sub-spaces have however no generalization power. The table **??** presents for different thresholds the results with the BKS and $HBKS$ methods. For the $HBKS$, we give the number of sub-spaces and non-empty cells. The sub-spaces results describe the special effect of the $HBKS$. These results are defined by a triplet (ξ_r, ξ_s, ξ_q) where ξ_r, ξ_s and ξ_q are the number of well recognized patterns, errors and rejected patterns respectively. For a low threshold (0.4), the $HBKS$ has no effect compared to the BKS. When the threshold is higher, the number of processed patterns by the $HBKS$ is higher. Although the addition of information may be risky, we show that information can be added by the sub-spaces while keeping a good reliability.

Table 6. Results.

Rejection threshold	Number of sub-spaces	Number of cells	BKS	HBKS	Sub-spaces results
0.4	2	223	98.45 / 1.41 / 0.14	98.45 / 1.41 / 0.14	(0,0,0)
0.5	5	230	98.43 / 1.40 / 0.17	98.44 / 1.41 / 0.15	(1,1,1)
0.6	58	362	98.31 / 1.29 / 0.40	98.35 / 1.31 / 0.34	(4,2,20)
0.7	107	485	98.23 / 1.20 / 0.57	98.31 / 1.25 / 0.44	(8,5,30)
0.8	137	590	98.11 / 1.14 / 0.75	98.21 / 1.21 / 0.44	(10,7,44)
0.9	166	702	98.10 / 1.02 / 0.88	98.20 / 1.08 / 0.72	(10,6,58)

7 Conclusion

In this paper, a new fusing method has been presented for multi-classifiers systems with a parallel topology for problems of M classes ($M > 2$). It corresponds to an improvement of the existing BKS method by adding knowledge about the rank of the results. Function to a fixed confidence threshold value, each cell is divided into sub-spaces in order to solve ambiguities. We have shown that the proposed method can allow an optimal rejection control for the training database. It also provides new information for some ambiguous cells, without using new classifiers for the empty cells. For an optimal use of this method classifiers must provide ranking results, which have a real sense. Further works would deal with the optimal use of the $HBKS$ method and the threshold selection for getting the best generalization.

References

1. Alpaydin, E.: Improved classification accuracy by training multiple models and taking a vote. In: 6th Italian Workshop. Neural Nets Wirn Vietri-93. (1994) 180–185
2. Borda, J-C.: Mmoire sur les lections au scrutin, Histoire de l'acadmie royale des sciences, Paris, (1781)
3. Breiman, L.: Bagging Predictors, Machine Learning 24 (2), (1996) 123–140
4. Van Erp M., and Schomaker L.: Variants of the Borda count method for combining ranked classifier hypotheses. In Proc. of the Seventh International Workshop on Frontiers in Handwriting Recognition (2000) 443–452
5. Gunes, V., Ménard, M., Loonis, P., Petit-Renaud, S.: Systems of classifiers: state of the art and trends. International Journal of Pattern Recognition and Artificial Intelligence 17 (8) World-Scientific, (2004)
6. Huang, Y.S., Suen, C.Y.: A method of combining multiple experts for the recognition of unconstrained handwritten numerals. IEEE Trans Pattern Anal Mach Intell 17 (1), (1995) 90–94
7. Lam, L., Suen, C.Y.: Application of majority voting to pattern recognition: an analysis of its behavior and performance. IEEE Trans Pattern Anal Mach Intell 27 (5), (1997) 553–568
8. LeCun Y., Bottou L., Bengio Y., Haffner P.: Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE 86 (11), (1998) 2278–2324
9. LeCun, Y., Bottou, L., Orr, G., and Muller, K.: Efficient BackProp, in Neural Networks: Tricks of the trade (G. Orr and Muller K., eds.) (1998)
10. Liu, C.-L., Nakashima K., Sako H., Fujisawa H.: Handwritten digit recognition: investigation of normalization and feature extraction techniques, Pattern Recognition 37 (2004) 265–279
11. Rahman, A.F.R., Fairhurst, M.C: Multiple classifier decision combination strategies for character recognition: A review. International Journal on Document Analysis and Recognition 5, (2003) 166–194
12. Raudys, S., Roli, F.: The Behavior Knowledge Space Fusion Method: Analysis of Generalization Error and Strategies for Performance Improvement. Multiple Classifier Systems 4 (2003) 55–64
13. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. 7th International Conference on Document Analysis and Recognition. (2003) 958–962