ELSEVIER

Pattern Recognition Letters 22 (2001) 1283–1289

Pattern Recognition Letters

www.elsevier.com/locate/patrec

On combining classifiers using sum and product rules

Luís A. Alexandre [a,b,*], Aurélio C. Campilho [a,c], Mohamed Kamel [d]

[a] *INEB – Instituto de Engenharia Biomédica, Rua Dr. Roberto Frias, sn, 4200-465 Porto, Portugal*
[b] *Departamento de Informática, Universidade da Beira Interior, 6200-001 Covilhã, Portugal*
[c] *Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, sn, 4200-465 Porto, Portugal*
[d] *Department of Systems Design Engineering, University of Waterloo, Waterloo, Ont., Canada N2L 3G1*

Abstract

This paper presents a comparative study of the performance of arithmetic and geometric means as rules to combine multiple classifiers. For problems with two classes, we prove that these combination rules are equivalent when using two classifiers and the sum of the estimates of the a posteriori probabilities is equal to one. We also prove that the case of a two class problem and a combination of two classifiers is the only one where such equivalence occurs. We present experiments illustrating the equivalence of the rules under the above mentioned assumptions.

Keywords: Classification; Combining classifiers; Classifier fusion; k nearest-neighbours; Neural networks

1. Introduction

It is well known that in many situations combining the output of several classifiers leads to an improved classification result (Hansen and Salamon, 1990; Rogova, 1994; Tax et al., 1997; Opitz and Maclin, 1999). This happens because each classifier makes errors on a different region of the input space. In other words, the subset of the input space that each classifier will attribute a correct label will differ from one classifier to another. This implies that by using information from more than one classifier it is probable that a better overall accuracy can be obtained for a given problem.

When combining the outputs of different classifiers two cases emerge: all classifiers use the same features or they work in different feature spaces (Kittler et al., 1998). The results in this paper are valid in both cases.

There has been some interest on the comparative performance of the sum and product rules (or the arithmetic and geometric means) (Kittler et al., 1996; Tax et al., 1997; Kittler et al., 1998). The arithmetic mean is one of the most frequently used combination rules since it is easy to implement and normally produces good results.

In (Kittler et al., 1998), the authors show that for combination rules based on the sum, such as the arithmetic mean, and for the case of classifiers working in different feature spaces, the arithmetic mean is less sensitive to errors than geometric mean.

Tax et al. (1997) found experimentally that the combining rules based on the product give better

* Corresponding author.
E-mail addresses: lfbaa@noe.ubi.pt (L.A. Alexandre), campilho@fe.up.pt (A.C. Campilho), mkamel@watfast.uwaterloo.ca (M. Kamel).

0167-8655/01/$ - see front matter
PII: S0167-8655(01)00073-3

results when all classifiers produce small errors. If at least one of the classifiers makes large errors then the arithmetic mean rule gives better results.

We show that when working in a classification problem with two classes, and using two classifiers that give estimates of the a posteriori probabilities that sum to one, such as k nearest-neighbour (k-NN) classifiers, the combination rules arithmetic mean (or the sum) and the geometric mean (or the product) are equivalent. That is, they have exactly the same error rates. We also show that this is the only case when these rules are equivalent when using this type of classifiers.

In Section 2 we define the problem and introduce the notation. In Section 3 we study the performance of the two combination rules under the above mentioned assumptions. In Section 4 we investigate the consequences of violating these assumptions. In Section 5 several experiments are presented that illustrate the different aspects of the problem under consideration. Section 6 presents a discussion of the results and in the last section conclusions are posted.

2. Basic Concepts

This section introduces the formalism, presents the problem definition and describes the combination process.

2.1. Problem definition

A pattern is, in general, a p-dimensional, real valued, vector $\boldsymbol{x}$. It is associated with a class label which can be represented by $y \in \{c_1, \ldots, c_L\}$. We call a set of patterns with their classes a test set: $\mathrm{TS} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, T\}$.

The goal of classification is, given a TS, to correctly attribute a class label to a pattern not in the TS. This action is made by a classifier.

Consider that the problem has L classes. Consider also a classifier that can approximate the a posteriori probability functions $p(c_j|\boldsymbol{x})$, which gives the probability of the pattern $\boldsymbol{x}$ belonging to a given class c_j, given that $\boldsymbol{x}$ was observed. It is then natural to classify the pattern by choosing the class with the largest a posteriori probability:

$$\boldsymbol{x} \in c_k \quad \text{if } p(c_k|\boldsymbol{x}) = \max_j p(c_j|\boldsymbol{x}) \tag{1}$$

2.2. Single classifier

Consider a single classifier whose outputs are expected to approximate a posteriori probabilities $p(c_i|\boldsymbol{x})$, where c_i stands for class i and $\boldsymbol{x}$ is the input to the classifier.

The approximation to the a posteriori probability $p(c_i|\boldsymbol{x})$ provided by a single classifier j, is

$$f_i^j(\boldsymbol{x}) = p(c_i|\boldsymbol{x}) + \epsilon_i^j(\boldsymbol{x}),$$

where $\epsilon_i^j(\boldsymbol{x})$ represents the error that the classifier j introduces, when approximating the a posteriori probability $p(c_i|\boldsymbol{x})$.

2.3. Combining different classifiers

As mentioned before, strong evidence exists that better classification can be obtained if instead of using the predictions of a single classifier, the information from several classifiers is used. This information is then combined to produce a final decision.

Consider N classifiers that produce approximations to the a posteriori probabilities. For a given input pattern $\boldsymbol{x}$, each classifier j will produce L approximations to the a posteriori probabilities, $f_i^j(\boldsymbol{x}),\ i = 1, \ldots, L$.

The combining of information from the different classifiers is done by building new predictions for the a posteriori probabilities from the individual classifiers' predictions. The combined prediction for class c_i is then

$$f_i^{\mathrm{comb}}(\boldsymbol{u}_i) = G(\boldsymbol{u}_i), \tag{2}$$

where $\boldsymbol{u}_i = (f_i^1(\boldsymbol{x}), \ldots, f_i^N(\boldsymbol{x}))$, 'comb' represents a given combination rule and $G()$ is some function.

This process is illustrated in Fig. 1, where the final class label j is such that

$$f_j^{\mathrm{comb}} > f_i^{\mathrm{comb}} \quad \forall i \neq j;\ \ i, j \in \{1, \ldots, L\}$$

and the K_i are the classifiers.

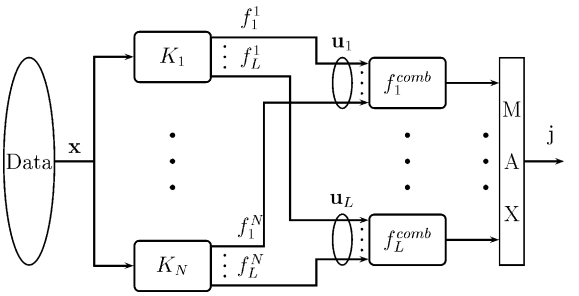

Fig. 1. The combination process.

2.4. Averaging combination rules

We are interested in comparing the performance of combining using the following possibilities for the function $G(\boldsymbol{u}_i)$: an arithmetic mean, giving the following form for f_i^{comb}

$$f_i^{\text{am}}(\boldsymbol{u}_i) = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{u}_i(j) \tag{3}$$

and a geometric mean

$$f_i^{\text{gm}}(\boldsymbol{u}_i) = \left(\prod_{j=1}^{N} \boldsymbol{u}_i(j)\right)^{1/N}, \tag{4}$$

where $\boldsymbol{u}_i(j)$ denotes the jth component of vector $\boldsymbol{u}_i$.

3. Performance of the combination rules

We start by proving that under some conditions the previous combination rules are equivalent and then show that those are the only conditions for the equivalence.

3.1. Equivalence of sum and product with $L = N = 2$

Since $L = 2$, there are two vectors of a posteriori probabilities: $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. And given that $N = 2$, these vectors have two coordinates: $\boldsymbol{u}_i(j)$, $i = 1, 2$; $j = 1, 2$. We are considering classifiers whose estimates of the a posteriori probabilities sum to one, that is

$$\boldsymbol{u}_1(i) + \boldsymbol{u}_2(i) = 1, \quad i = 1, 2. \tag{5}$$

There are two cases when the sum and the product disagree in their predicted class.

Case 1:

$$\begin{aligned}\boldsymbol{u}_1(1)\boldsymbol{u}_1(2) > \boldsymbol{u}_2(1)\boldsymbol{u}_2(2) \wedge \boldsymbol{u}_1(1) + \boldsymbol{u}_1(2)\\ < \boldsymbol{u}_2(1) + \boldsymbol{u}_2(2)\end{aligned} \tag{6}$$

Given (5) we can rewrite the first part of expression (6) as

$$(1 - \boldsymbol{u}_2(1))(1 - \boldsymbol{u}_2(2)) > \boldsymbol{u}_2(1)\boldsymbol{u}_2(2)$$

or

$$\boldsymbol{u}_2(1) + \boldsymbol{u}_2(2) < 1. \tag{7}$$

The second part of expression (6) can also be rewritten using (5)

$$1 - \boldsymbol{u}_2(1) + 1 - \boldsymbol{u}_2(2) < \boldsymbol{u}_2(1) + \boldsymbol{u}_2(2)$$

or

$$\boldsymbol{u}_2(1) + \boldsymbol{u}_2(2) > 1. \tag{8}$$

It is impossible that a point $\boldsymbol{u}_2$ satisfies both (7) and (8) simultaneously.

The second case when the sum and the product disagree in their predicted class is

Case 2:

$$\begin{aligned}\boldsymbol{u}_1(1)\boldsymbol{u}_1(2) < \boldsymbol{u}_2(1)\boldsymbol{u}_2(2) \wedge \boldsymbol{u}_1(1) + \boldsymbol{u}_1(2)\\ > \boldsymbol{u}_2(1) + \boldsymbol{u}_2(2)\end{aligned} \tag{9}$$

As in Case 1, a point now would have to satisfy conditions (7) and (8), which is impossible. This proves that it is not possible to find two points that obey (5) and make the sum and product disagree, for a two class problem.

This makes the arithmetic and the geometric mean equivalent when combining two classifiers that obey (5) in any two class problem.

3.2. General case

In this section we prove that this equivalence of the combination rules does not generalise for other values of $(L,\ N)$ other than $(2, 2)$.

In general, Eq. (5) becomes

$$\sum_{i=1}^{L} \boldsymbol{u}_i(j) = 1, \quad j = 1, \ldots, N. \tag{10}$$

We start by noting that for the case $(L = 2,\ N = 3)$ the following two points make the product and the sum rule disagree

$$\{\boldsymbol{u}_1(0;0.8;0.8),\ \boldsymbol{u}_2(1;0.2;0.2)\}. \tag{11}$$

(The arithmetic mean of $\boldsymbol{u}_1$ is greater than the arithmetic mean for $\boldsymbol{u}_2$ making class 1 the chosen for the arithmetic mean combining rule. The geometric mean of $\boldsymbol{u}_1$ is smaller than for $\boldsymbol{u}_2$ making class 2 the chosen one for the geometric mean.)

For the case $(L = 3,\ N = 2)$ the following three points make the combination rules disagree.

$$\{\boldsymbol{u}_1(0.1;0.65),\ \boldsymbol{u}_2(0.4;0.3),\ \boldsymbol{u}_3(0.5;0.05)\}. \tag{12}$$

We now describe a way to create from these points other points that make those rules disagree for values of $(L,\ N)$ other than $(2,\ 2)$.

To create a new set of points that make the combination rules disagree for $(L,\ N)$ from a set of points that make the combination rules disagree for $(L-1,\ N)$ it is enough to consider a new point $\boldsymbol{u}_L$ with all coordinates equal to zero. Example: from the points in (11), create a new set of points that make the product and the sum rule disagree for $(L = 3,\ N = 3)$. The new set of points is then

$$\{\boldsymbol{u}_1(0;0.8;0.8),\ \boldsymbol{u}_2(1;0.2;0.2),\ \boldsymbol{u}_3(0;0;0)\}. \tag{13}$$

This process guarantees that the new points obey to (10) and they do not change the initial decisions of the classifiers since what happens is that the new point will have the smallest sum and the smallest product (both zero) and since the decision is made using the maximum it will not interfere with the initial conflicting decision.

To create a new set of points that make the combination rules disagree for $(L,\ N)$ from a set of points that make the combination rules disagree for $(L,\ N-1)$ it is enough to add to each point a new coordinate with value equal to $1/L$.

Example: from the points in (11), create a new set of points that make the product and the sum rule disagree for $(L = 2,\ N = 4)$. The new set of points is then

$$\{\boldsymbol{u}_1(0;0.8;0.8;0.5),\ \boldsymbol{u}_2(1;0.2;0.2;0.5)\}. \tag{14}$$

This process guarantees that the new points obey to (10) and they do not change the initial decisions of the classifiers since what happens is that both points will have their sum increased with the same amount $(1/L)$ and their product multiplied by the same amount (also $1/L$), thus not changing the initial conflicting decision.

By using these two procedures and the sets of points in (11) and (12), one can find points that make the combination rules disagree for all values of $(L,\ N)$ other than $(2,\ 2)$. (Note that the cases with $N = 1$ are not combinations of classifiers and the cases with $L = 1$ are not classification problems since all points belong to one single class.)

This way we have shown that the only case when the arithmetic and the geometric means give the same result, when used for combining the outputs of classifiers that give estimates for the a posteriori probabilities that sum to one, is when $L = N = 2$.

4. Estimates of the a posteriori probabilities

Depending on the type of classifier used to produce the estimation we face two scenarios: either the outputs of the classifier sum to one thus obeying Eq. (10) (scenario A) or the outputs do not sum to one (scenario B).

4.1. Scenario A

Examples of cases that fall into scenario A are those produced by a k-NN classifier. It is possible to produce estimates of the a posteriori probabilities by dividing the number of the k-NNs belonging to a given class, by k, as in

$$f_i(\boldsymbol{x}) = \frac{nc_i(\boldsymbol{x})}{k} \tag{15}$$

with nc_i being the number of the k-NNs of $\boldsymbol{x}$ belonging to class c_i (Bishop, 1995). This way,

$$\sum_{i=1}^{L} f_i(\boldsymbol{x}) = \sum_{i=1}^{L} \frac{nc_i(\boldsymbol{x})}{k} = 1$$

which makes k-NN classifiers' predictions of the a posteriori probabilities fall into scenario A.

4.2. Scenario B

Cases that fall into scenario B are those produced, for instance, by a neural network. In this

case there are no guarantees that the outputs of each NN will sum to one.

Another interesting characteristic of these classifiers is the fact that the error may push the estimate of the a posteriori probabilities out of the [0, 1] interval. This can have a major effect especially on the geometric mean, since the product of an odd number of negative estimates produces a negative estimate which will not be chosen even against the worst positive estimation. This may have a severe impact on the accuracy of the geometric mean combination estimates.

We will conduct experiments illustrating these two scenarios and complete the discussion afterwards.

5. Experiments

In this section we present several experiments that confirm the previous results.

5.1. Data sets

We used three data sets from the UCI repository (Blake et al., 1998). Details are presented in Table 1. The first column lists the reference we use for a given data set, the second lists their names, the third the number of patterns, the fourth the number of features and the last the number of classes in the problem.

5.2. Scenario A

In these experiments, k-NN classifiers using euclidean distance are used to produce estimates of the a posteriori probabilities using Eq. (15). These estimates are combined using arithmetic and geometric means. Experiments were made with two, three and four k-NN classifiers. Table 2 presents the errors for each data set, and for the combinations: 'AMi' means 'arithmetic mean combination of the first i classifiers' and 'GMi' means 'geometric mean combination of the first i classifiers'.

Table 1
Data sets

Reference	Name	# Points	# Features	# Classes
DS1	Diabetes	768	8	2
DS2	Iono-sphere	351	34	2
DS3	Sonar	208	60	2

Table 2
Classification errors (in percentage)

Classifiers	DS1	DS2	DS3
1NN	32.03	13.43	17.31
3NN	30.60	15.14	18.27
5NN	28.52	15.43	17.31
7NN	27.21	16.29	23.08
AM2	32.03	13.43	17.31
GM2	32.03	13.43	17.31
AM3	32.03	13.43	17.31
GM3	32.03	13.43	17.31
AM4	30.08	14.00	19.23
GM4	32.03	13.43	17.31

5.3. Scenario B

We made experiments combining two, three and four feed-forward neural networks only differing in their topology and in the weight initialisation (which is done randomly). The classifiers should approximate a posteriori probabilities, and these neural networks do so (Richard and Lippmann, 1991). The classifiers are multi-layer perceptrons (MLPs), trained for 300 epochs using resilient backpropagation (Demuth and Beale, 1998). Their topology is presented in Table 3, where the first number indicates the number of neurons on the input layer, the second represents the number of neurons in the hidden layer and the last represents the number of neurons on the output layer. Tables 4–6 present the average errors for each data set along with the respective S.D.

Table 3
Topology of the MLPs

Classifiers	DS1	DS2	DS3
MLP1	[4 6 2]	[10 5 2]	[5 40 2]
MLP2	[6 6 2]	[15 10 2]	[5 50 2]
MLP3	[8 6 2]	[20 10 2]	[5 60 2]
MLP4	[8 8 2]	[25 10 2]	[10 40 2]

Table 4
Average classification errors and S.D. (in percentage) for $N = 2$

Classifiers	DS1	DS2	DS3
MLP1	31.43 (4.63)	11.43 (3.10)	33.22 (5.06)
MLP2	27.80 (4.10)	11.46 (3.02)	30.43 (3.99)
AM	27.98 (3.54)	10.86 (2.77)	29.86 (3.98)
GM	27.86 (3.55)	10.80 (2.73)	30.53 (4.66)

Table 5
Average classification errors and S.D. (in percentage) for $N = 3$

Classifiers	DS1	DS2	DS3
MLP1	32.51 (4.42)	12.40 (3.31)	32.98 (5.86)
MLP2	29.35 (4.21)	11.37 (2.29)	31.97 (4.57)
MLP3	28.98 (5.09)	11.77 (3.55)	32.84 (5.44)
AM	28.10 (4.38)	9.94 (2.31)	28.75 (4.78)
GM	29.41 (4.30)	10.60 (2.23)	30.19 (4.93)

Table 6
Average classification errors and S.D. (in percentage) for $N = 4$

Classifiers	DS1	DS2	DS3
MLP1	30.18 (5.99)	10.86 (2.78)	34.62 (4.48)
MLP2	30.18 (4.51)	11.09 (2.50)	33.08 (5.92)
MLP3	28.71 (3.51)	12.06 (3.72)	34.66 (5.56)
MLP4	29.40 (4.25)	11.60 (2.33)	34.38 (3.09)
AM	27.04 (3.57)	9.74 (2.60)	30.63 (3.31)
GM	28.46 (4.42)	10.34 (2.23)	31.01 (4.81)

The columns show the average error in percentage for the isolated classifiers and for the combinations, for each data set. The experiments were repeated 20 times.

There was not a special care in tuning the individual classifiers, since the important issue is comparative performance.

6. Discussion

As expected, in scenario A, the combination of two classifiers provide equal results for all data sets, since they are all two class problems. Notice that in the case $N = 3$, the combination rules also perform equally well. But as seen in Section 3.2 this is not always true. In the combination of four classifiers the geometric mean gives better results than the arithmetic mean for data sets DS2 and DS3.

In scenario B, we see that the combination rules do not have the same performance, not even for the case of $N = 2$. This is due to the fact that these classifiers do not obey expression (10) as mentioned before.

The arithmetic mean outperforms the geometric mean. The performance gain relative to the geometric mean is in agreement with results from Kittler et al. (1998), where it was noted that the product rule is more sensitive to error than the sum rule. Note that an eventual negative output for an estimate of the a posteriori probabilities can be easily compensated in the arithmetic mean by the other positive values, since we are talking about a sum. In the geometric mean, an odd number of negative estimates makes the result negative. The bad performance of the geometric mean is connected with the fact that a possible negative value for the a posteriori probabilities estimates by only one classifier can make the result for a given class smaller (negative) than all the other possible not so good classes but that had no negative output for any of their a posteriori probabilities estimates. Another justification for the bad performance of the geometric mean is that a given classifier can make the combination result zero or very small by giving a zero or very small estimate for the a posteriori probabilities even if all other classifiers give high values for the a posteriori probabilities.

7. Conclusions

In this paper we study the relative performance of two types of averaging combination rules: arithmetic and geometric means. We show that, for a problem with two classes, and when using two classifiers that give a posteriori probabilities values that sum to one, such as k-NN classifiers, these rules have exactly the same performance. That is, in these cases more than two classifiers should be combined in order to have different performances from these combination rules. We also show that this is the only case when this equivalence holds. When more than two classifiers are combined the geometric mean showed better

performance than the arithmetic mean in our experiments.

We also study the behaviour of these rules when using classifiers that do not give estimates of the a posteriori probabilities that sum to one, such as MLPs. We conclude experimentally, that in this case the arithmetic mean outperforms the geometric mean.

Note that the studied combination rules all give the same importance to all the classifiers. This leaves open the combinations using weighted average. See on this subject the following references: Hashem (1997), Alexandre et al. (2000) and Ueda (2000).

Acknowledgements

We acknowledge the support of project number POSI/35590/SRI/2000 approved by the portuguese FCT and POSI partially financed by FEDER.

References

Alexandre, L., Campilho, A., Kamel, M., 2000. Combining independent and unbiased classifiers using weighted average. In: Proc. 15th Internat. Conf. on Pattern Recognition, Vol. 2. IEEE Press, Barcelona, Spain.

Bishop, C., 1995. Neural Networks for Pattern Recognition. Oxford University Press, Oxford.

Blake, C., Keogh, E., Merz, C., 1998. UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Demuth, H., Beale, M., 1998. Neural Network Toolbox User's Guide. The MathWorks, Inc.

Hansen, L., Salamon, P., 1990. Neural network ensembles. IEEE Trans. PAMI 12 (10), 993–1001.

Hashem, S., 1997. Optimal linear combinations of neural networks. Neural Networks 10 (4), 599–614.

Kittler, J., Hatef, M., Duin, R., 1996. Combining classifiers. In: Proc. Internat. Conf. on Pattern Recognition'96.

Kittler, J., Hatef, M., Duin, R., Matas, J., 1998. On combining classifiers. IEEE Trans. PAMI 20 (3), 226–239.

Opitz, D., Maclin, R., 1999. Popular ensemble methods: An empirical study. J. Artif. Intell. Res. (11), 169–198.

Richard, M., Lippmann, R., 1991. Neural network classifiers estimate Bayesian a posteriori probabilities. Neural Comput. (3), 461–483.

Rogova, G., 1994. Combining the results of several neural network classifiers. Neural Networks 7 (5), 777–781.

Tax, D., Duin, R., Breukelen, M., 1997. Comparison between product and mean classifier combination rules. In: Workshop on Statistical Techniques in Pattern Recognition. Prague, Czech Republic.

Ueda, N., 2000. Optimal linear combination of neural networks for improving classification performance. IEEE Trans. PAMI 22 (2), 207–215.