Comparison Between Product and Mean Classifier Combination Rules

David M.J. Tax, Robert P.W. Duin and Martijn van Breukelen
Pattern Recognition Group
Faculty of Applied Physics, Delft University of Technology
Lorentzweg 1, 2628 CJ Delft, The Netherlands
e-mail: {davidt,bob,martijnb}@ph.tn.tudelft.nl

Abstract

To obtain better classification results, the outputs of an ensemble of classifiers can be combined instead of just choosing the best classifier. This combining is often done by using a simple linear combination of the outputs of the classifiers or by using order statistics (using the order in the outputs for different classes). In this paper we will show that using the normalized product of the outputs of the classifiers can be more powerful for classification performance. We will show in which cases a product combination is to be preferred and where a combination by averaging can be more useful. This will be supported by theoretical and experimental observations.

1 Introduction

Certainly a very important property for a classifier is to respond meaningfully to novel patterns, i.e. the classifier *generalizes* [Wol94]. To obtain a network which generalizes well, one often constructs several different classifiers. Each of these classifiers have different decision boundaries and generalizes differently. The classifiers which generalizes best on a test set is then chosen to perform the classification task. It is observed though that it may be a waste to use only one of the pool of classifiers and ignore the information contained in the other classifiers [LT93], [BC94].To use all information in the classifiers, the outputs of all networks can be combined for the final decision. This combined classifier often outperforms the single classifiers and is more robust[SS95].

Combining may not only improve the generalization of the classifications, but may also suffer less from time and space constraints. When a large feature space can be split in several smaller spaces, on each feature space a classifier can be constructed (or learned). Constructing and combining these small classifiers to one larger network may be less time and space demanding than constructing one large classifier on the total feature space[JM97].

Several ways to combine classifiers exist. The combining of classifiers is often done using linear combinations of classifiers outputs (this can be just averaging or more advanced weighted linear combinations[Has94], [Jac95]), rank-based combining (in which each output class is ranked by the classifiers and then combined[HHS94]), voting-based combination ([BC94]) or combination based on Dempster-Shafer theory of evidence([Rog94]).

In this paper we will focus on the combination of estimations of class *probability densities.* Two very simple combination rules will be considered: the mean rule and the product rule (see for example [KHD96]). These combination rules combine the probability estimations by simple summation and multiplying respectively. We will show in what situations the product rule or the mean rule is to be preferred as combination rule.

In section 2 the mean and product combination rules will be introduced and some theoretical background will be given. In section 3 experiments will be shown, both artificial and real world problems. Section 4 will summarize the conclusions.

2 Combining rules

2.1 Derivation of the rules

The ultimate goal for a classifier is to correctly estimate the probability that an object belongs to a certain class ω_j. This object is represented by a measurement vector x, in which each component is a measurement of a feature. When R measurement vectors $x^1, ..., x^R$ from different feature spaces are available, this probability $P(\omega_j|x^1, ..., x^R)$ has to be approximated (see also [KHD96]). In each of the R feature spaces a classifier can be constructed which approximates the true class probability $P(\omega_j|x^k)$ in that feature space:

$$f_j^k(x^k) = P(\omega_j|x^k) + \epsilon_j^k(x^k) \tag{1}$$

A good combination rule uses these $f_j^k(x^k)$ to approximate $P(\omega_j|x^1, ..., x^R)$ as optimal as possible.

Two combination rules will be considered, the mean rule and the product rule:

$$\begin{aligned} f_j(x^1, ..., x^R) &= \frac{1}{R}\sum_{k=1}^{R} f_j^k(x^k) &(2)\\ f_j(x^1, ..., x^R) &= \frac{\prod_{k=1}^R f_j^k(x^k)}{\sum_{j'}\prod_{k=1}^R f_{j'}^k(x^k)} &(3)\end{aligned}$$

Two extreme cases can be distinguished, the first in which the feature spaces (and therefore the measurements of the objects) are all the same, the second in which all feature spaces and measurements are different and independent. In the case of identical feature spaces, the classifiers all use the same data x and approximate the same probability-distribution when they are designed to do so:

$$P(x^1, ..., x^R|\omega_j) = P(x^1|\omega_j) \cdot \delta(x^1 - x^2) \cdot ...\delta(x^{R-1} - x^R) \tag{4}$$

Using Bayes we can derive:

$$P(\omega_j|x^1, ..., x^R) = \frac{P(x^1, ..., x^R|\omega_j)P(\omega_j)}{P(x^1, ..., x^R)} = P(\omega_j|x^k) \quad \text{for any } k,\ 1 \le k \le R \tag{5}$$

This $P(\omega_j|x^k)$ has to be estimated by $f_j^k(x^k)$. To obtain a less error-sensitive estimation, all $f_j^k(x^k)$'s can be averaged and thus eq.(2) is obtained.

In the second case all feature spaces are different and independent and the probabilities can be written as:

$$P(x^1, ..., x^R|\omega_j) = P(x^1|\omega_j) \cdot P(x^2|\omega_j) \cdot ...P(x^R|\omega_j) \tag{6}$$

Using Bayes again, we derive:

$$P(\omega_j|x^1,...,x^R) = \frac{\prod_k \Big(P(x^k|\omega_j)P(\omega_j)\Big)}{\sum_{j'}\left\{\left(\frac{P(\omega_j)}{P(\omega_{j'})}\right)^{R-1}\prod_{k'}P(x^{k'}|\omega_{j'})P(\omega_{j'})\right\}} \tag{7}$$

In case of equal apriori class probabilities ($P(\omega_j) = 1/(\text{number of classes})$), this formula reduces to a product rule (eq.(3)) with $\epsilon_j^k(x^k) = 0$.

2.2 Choice between product and mean rule

To make a choice between mean and product rule, one has to compare the number of misclassifications made by the different combination rules. Misclassification in two class problems means that although an object x belongs with probability larger than a half ($P(\omega_A|x) > 0.5$) to class ω_A, the combination rule obtains a contrary result ($P_{est}(\omega_A|x) < 0.5$).

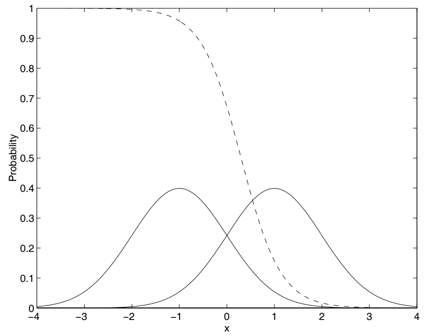

Figure 1: Solid line: True probability distribution, left class is called ω_A, the right is class ω_B, dashed line: conditional distribution estimated for class ω_A by a classifier.

In figure 1 the probability densities of two gaussian distributed classes is shown ($\mathcal{N}(-1.0, 1.0)$ and $\mathcal{N}(+1.0, 1.0)$). The two classes have an overlap of 15.9%. R classifiers are constructed, which estimate conditional probability estimation that x belongs to the left class ω_A using their own estimated class probability distributions ($\mathcal{N}(-1.0+\epsilon_A, 1.0)$ and $\mathcal{N}(+1.0+\epsilon_B, 1.0)$). Because the classifiers make imperfect estimations (the estimated mean differs ϵ_j from the true mean) some classifications will go wrong.

In figure 2 the probabilities obtained by the combination rules of 250 sets of three classifiers ($R = 3$) are shown. The classifiers have errors $\epsilon_j = \mathcal{N}(0.05, 0.0)$ in their estimation of the means of the classes. Each dot in the figure represents the result of the mean rule and product rule. In the left figure ('mean model') classifiers are given the same measurement vector x, which models an identical feature space for all classifiers. For this x the $P(\omega_A|x) = 0.65$. In the right figure ('product model') independent measurements from a class A are given, which models independent feature spaces. x's are drawn from class ω_A with probability 0.65 and probability of 0.35 from class ω_B.

To prefer one combination rule above another, the upper-left and lower-right parts of the figures have to be compared. When objects appear in the upper-left part, it means that the mean combination rule made a good combination of the R classifiers for this object, while the product combination rule did not (object x is classified to the wrong class). The reverse holds for the lower-right part. The right figure in figure 2 shows a better performance for the product rule. In the left figure both mean and product rule are equally well.

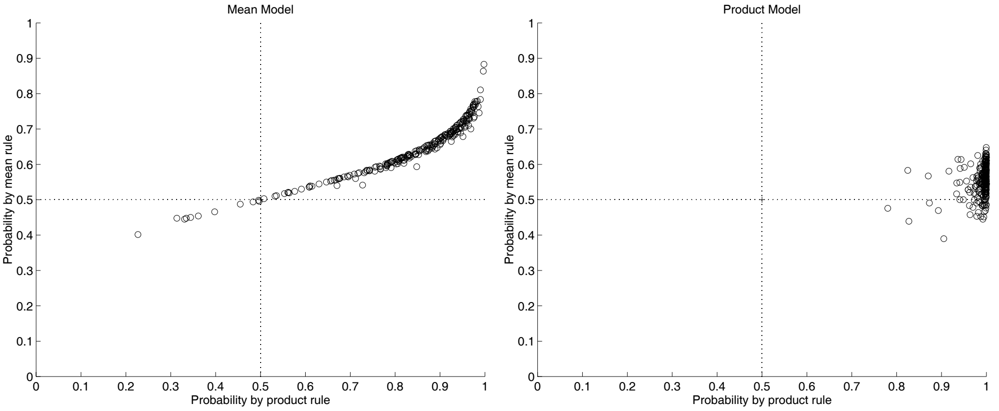

Figure 2: Probabilities obtained by product rule and mean rule for an object x with $P(\omega_A|x) = 0.65$. The errors of the estimations of the probability distributions are small: $\epsilon_j = \mathcal{N}(0.05, 0.0)$.

For (relative) small errors made by the classifiers, we see that both in the mean model as in the product model the product rule performs well for $P(\omega_A|x) = 0.65$. For other probabilities this holds also. In situations where $P(\omega_A|x) = 0$ or 1 both combination rules perform very good, while when $P(\omega_A|x) = 0.5$ performances are equally poor. Thus for the case of small estimation errors the product can be preferred for all combination tasks.

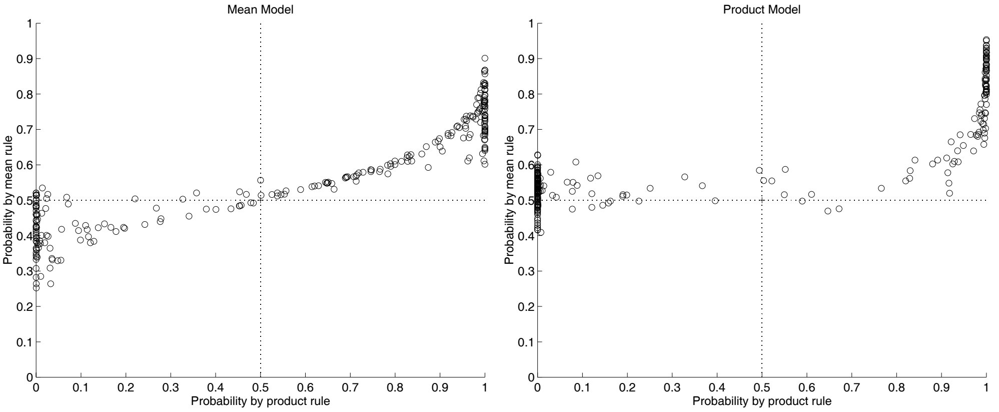

Figure 3: Probabilities obtained by product rule and mean rule for an object x with $P(\omega_A|x) = 0.65$. The errors of the estimations of the probability distributions are small, except for one classifier: $\epsilon_j^{k'} = \mathcal{N}(5.0, 0.0)$. '+' indicates where $P_{prod} = 0.5$ and $P_{mean} = 0.5$.

When the errors are large however, especially when one classifier makes a very bad estimation, the picture changes drasticly (see figure 3). In the mean rule the probabilities stay between 0.4 and 1.0 (with a $P(\omega_A|x) = 0.65$) but in the product rule they cover the entire range from 0.0 to 1.0. Especially in the region where the mean rule estimates probabilities of around 0.5 and 0.6, the product rule is very uncertain and outputs values between 0.0 and 1.0. This sensitivity can be confirmed by a theoretical analysis (see [KHD96]). Even more serious is the region where the mean rule gives probabilities around 0.4 or 0.5. Here the product rule outputs values of 0.0. This occurs when one of the classifiers has overruled all other estimations by its output of 0.0 which acts like a veto. In cases where the probability densities are very badly estimated and contain a lot of zeros, this veto becomes very dominant.

3 Experiments

To show that these two extremes can be really observed, two experiments will be done, using a pattern recognition Matlab toolbox ([Dui95]). In the first artificial problem 3 classifiers are trained on three independent feature spaces. In these 3 feature spaces two classes are present, which are distributed by 3 different distributions: two distributions with overlapping normal densities, one with equal and one with different covariance matrices and a banana shaped data set. The three classifiers are normal density based quadratic classifiers. By using more training patterns the accuracy of the estimations of the classifiers improves and the product rule can be used for combining (see table 1).

Classifier	Error ($P = 16$)	Error ($P = 160$)
classifier feature set 1	0.230 ± 0.112	0.200 ± 0.039
classifier feature set 2	0.170 ± 0.095	0.078 ± 0.021
classifier feature set 3	0.130 ± 0.111	0.116 ± 0.036
classifier all feat.sets	0.125 ± 0.130	0.013 ± 0.009
product combination	0.160 ± 0.142	0.021 ± 0.012
mean combination	0.035 ± 0.077	0.049 ± 0.018

Table 1: Classification performance on a test set of 160 patterns of the combination rules of 3 classifiers in independent feature spaces. Classifiers are normal density based quadratic classifiers, P is the total number of train patterns, averaged on 25 runs.

The second experiment is a hand written digit problem. From a set of 2000 hand written digits (200 per class) four different feature sets are measured. These features are the Fourier transformed, Karhunen-Loève transformed, ordinary pixel values and Zernike moments (see [BDT97]). In these four feature spaces four simple linear classifiers are trained. These linear classifiers assume normal densities with equal covariance matrices. To obtain a probability estimation over the whole feature space a sigmoid function is fitted using the Maximum Likelihood criterium. On the decision boundary the probability is 0.5, far away the probabilities are 0.0 or 1.0. In table 2 the result of the individual classifiers and the combination rules are shown (tested on an independent test set of 50 patterns per class). We see that in both cases, large and small number of train samples, the product rule outperforms the mean rule. Although the probability density can not be estimated very well by just a linear classifier, it seems that in this classification task no classifiers disturb the product rule by giving a probability estimation of 0.0. Combined the classifiers give a quite reasonable classification result.

Classifier	Error ($P = 50$)	Error ($P = 150$)
classifier Fourier feature set	0.214 ± 0.008	0.198 ± 0.017
classifier Karhunen-Loève feature set	0.054 ± 0.010	0.054 ± 0.013
classifier Pixel feature set	0.105 ± 0.010	0.050 ± 0.013
classifier Zernike feature set	0.153 ± 0.005	0.152 ± 0.015
product combination	0.024 ± 0.009	0.014 ± 0.004
mean combination	0.032 ± 0.006	0.028 ± 0.005

Table 2: Performance of the combination rules of 4 classifiers in independent feature spaces. Classifiers are normal density based linear classifiers, P is the total number of train patterns. Results are averaged on 5 runs.

4 Conclusion

We investigated the differences between two combination rules for combining probability estimations of several classifiers. These rules are the mean rule and the product rule. It is shown that when the estimation of the classifiers have all small errors and the classifiers operate in several independent feature spaces, it becomes theoretically preferable to combine the output estimations of the classifiers by multiplying them. Also when the classifiers are used in the same feature space, the product combination rule can be used, for the differences between the mean and product combination rule are very small. This is confirmed by experiments.

When estimations of the classifiers contain large errors, the estimated probabilities can best be combined by the mean rule. Not only is the product rule very sensitive to errors in the probability estimations, it also contains a veto mechanism: when one classifier outputs a zero for an estimation, the complete combination rule outputs zero. The mean rule is not as sensitive as the product rule and is thus to be preferred in the case of larger estimation errors.

The next challenge will be to estimate by forehand which of the two combination rules have to be used in a certain classification problem. It may be possible to apply the mean rule to subsets of classifiers and next combine all classifiers with yet combined classifiers with the product rule. Therefore more research is needed.

Acknowledgments

This work was partly supported by the Foundation for Applied Sciences (STW), the Foundation for Computer Science in the Netherlands (SION) and the Dutch Organization for Scientific Research (NWO).

References

[BC94] Battiti R. and Colla A.M. Democracy in neural nets: Voting schemes for classification. *Neural Networks*, 7(4):691–707, 1994.

[BDT97] Breukelen van M., Duin R.P.W., and Tax D.M.J. Combining classifiers for the recognition of handwritten digits. *Paper for Workshop on Statistical Techniques in Pattern Recognition*, 1997.

[Dui95] Duin R.P.W. Prtools, a matlab toolbox for pattern recognition, October 1995.

[Has94] Hashem S. Optimal linear combinations of neural networks. *Neural Networks*, October 1994.

[HHS94] Ho T.K., Hull J.J., and Srihari S.N. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, January 1994.

[Jac95] Jacobs R. Method for combining experts' probability assessments. *Neural Computation*, 7(5), 1995. 867-888.

[JM97] Ji C. and Ma S. Combinations of weak classifiers. *IEEE Transactions on Neural Networks*, 8(1):32–42, january 1997.

[KHD96] Kittler J., Hatef M., and Duin R.P.W. Combining classifiers. *Proc. of ICPR'96*, pages 897–901, 1996.

[LT93] LeBlanc M. and Tibshirani R. Combining estimates in regression and classification. Technical Report 9318, Department of Statistics, University of Toronto, november 1993.

[Rog94] Rogova G. Combining the results of several neural network classifiers. *Neural Networks*, 7(5):777–781, 1994.

[SS95] Sharkey A. and Sharkey N. How to improve the reliability of artificial neural networks. Technical Report CS-95-11, Department of Computer Science, University of Sheffield, 1995.

[Wol94] Wolpert D.H. *The Mathematics of Generalization.* Goehring D., Santa Fe Institute, 1994.