An Improved Moderation Technique for Fusion of K-Nearest Neighbor classifiers

Fuad M. Alkoot
Telecommunication and Navigation Institute, PAAET
P.O.Box 4575, Alsalmia, 22046, Kuwait
alkoot@paaet.edu.kw

Abstract

Moderation is a strategy that modifies the expert probability estimates in order to overcome problems caused by multiplication of zero estimates with non zero ones, i.e. the veto effect. We experiment with different bagging methods containing moderated experts. We propose a modification to the original moderation method which leads to improved performance. Empirical tests on 4 real data sets prove the advantage of moderating kNN experts in a bagging scenario. Although the original moderation method does not outperform MProduct, the proposed moderation method of this paper does.

keywords: Bagging, Moderation, Fusion, nearest neighbor, MProduct

1 Introduction

The use of classifier fusion to improve classification accuracy has become increasingly popular and it is attracting a growing number of researchers from the pattern recognition community. The simplest fusion methods that do not require training are Sum, Product, Vote and order statistics combiners [7, 6]. While Product is directly derived from the bayesian based class a posteriori probability estimate, Sum which is derived from Product under restricting assumptions, sometimes outperforms Product [1]. However Product has also been shown to outperform Sum [8]. After finding the reason behind Product's mixed performance, we were able to modify it and improve its performance drastically[4]. The resulting Modified Product (MProduct) method was able to outperform all the simple fusion methods [2, 4]. The idea basically is that when the classes are not well represented in the training set the expert decision based on this training set may contain zero decisions. These zero probability estimates veto the decisions of the other experts in a combiner system, especially when a fusion strategy similar to the product method is used. Modifying the output of some of these vetoing experts may yield better fusion results, as is the case when MProduct is used.

Another strategy that implements a modification of the experts probability estimates is *Moderation* [3]. It is faster than MProduct since the modification is done at the classification stage, by adding a constant to all class estimates, during the decision making process and not at the expert output. MProduct, however, requires additional calculations at the fusion stage in order to decide whether a modification of expert outputs is to be performed or not.

We investigate the performance of Moderation applied to bagging k-Nearest Neighbor (kNN) experts. We find that, in consistence with theoretical expectations, it does improve Product while it leaves the single expert unaffected. Sum does not seem to be drastically affected although its performance fluctuates un-significantly. This was also the result when Moderation was used with MProduct. However there were few instances at which MProduct degraded significantly due to Moderation. This inspired us to modify the Moderation formula.

In the next section, 2, we describe Moderation. In section 3 we explain the different bagging methods [2] used in our experiments. In section 4 we setup the experimental methodology followed by results in section 5. In section 6 we discuss the results and draw the paper to conclusion in section 7.

2 Moderation

Due to space restriction we omit the derivation of the moderation technique. The reader may refere to [3] for a complete derivation. Moderation equation is simply:

$$P_j(\omega/x_i) = \frac{\kappa + 1}{K + 2} \tag{1}$$

Where it is the Moderation equivalent of $\frac{\kappa}{K}$. It will guarantee that $P_j(\omega/x_i)$ will never reach zero unless $K \to \infty$. However at $K \to \infty$ the kNN classifiers reach an optimum design with a bayesian error rate.

An essential criteria of a Moderation equation is that primarily it should be able to remove the cause of the veto, that is remove zero probability estimates , and secondly, concurrently, do not alter the non zero probability estimates. The closer the probability estimates are to the decision boundary the more important this second criteria becomes. For large values of K equation 1 works well because the constant in the numerator has a minute effect. However when K is small the constant term will have a considerable altering effect on the probability estimates, as shown in figure 1. At the same time that we want to overcome the zero outputs, we do not want the nonzero probability estimate largely altered, especially when decisions are close to the boundary of $\kappa = \frac{1}{2}K$. Although altering the estimate using the constant term, i.e. adding bias, does not toggle the probability estimates to a different class, any large bias that moves the probability estimates towards the boundary is at least not useful, if not harmful.

Due to this fact, which is also proven experimentally in the results section, equation 1 leads to sub-optimum performance. Considering the reasons stated above we propose a new Moderation equation as follows. Since we want to reduce the value of the constant term in the numerator for small values of K, we can replace the constant 1 with another constant j that depends on K. Hence the second Moderation equation will be:

$$P_j(\omega/x_i) = \frac{\kappa + j}{K + 2j} \tag{2}$$

$$j = \frac{1}{K^\alpha} \tag{3}$$

The power term α can be any constant, our experimental use of $\alpha = 10$ yielded good results. We refer to it as Moderation 2, or Mod2 for short.

Even though the moderation method of equation 2 solves the shortcomings of equation 1 it has a weakness too. It leads to near zero values of j for larger values of K. However as K gets larger the training set is less biased and we most likely will not need to Moderate. This is why our experimental results did not show any degradation in comparison to the first moderation method, at large values of K. Nevertheless, it would be safer if we could use a value j that would be relatively small when K is small and gradually increase as K increases, reaching a cutoff constant value when K becomes large. This can be achieved using the natural logarithm. Hence

$$j = \frac{\ln(K)}{c} \tag{4}$$

where c is a constant that minimizes the bias caused by adding j to the posterior probability estimate. This is referred to as moderation method 3, (Mod3).

It is also possible to use a value for j that increases linearly with K. We tested $j = \frac{K}{c}$, where c is a scaling constant which we set equal to 20000. This is referred to as moderation method 4, (Mod4).

3 Bagging and Modified Bagging Procedures

When a data set is small, the proportions of training patterns from the different classes may be unrepresentative. The probability of drawing a training set with samples from some class completely missing becomes non negligible. When this occurs, bagging may even become counterproductive. Three modifications of the standard bagging method were considered. We name the standard procedure as method 1 and its modified versions as methods 2-4. The methods which exploit increasing amounts of prior knowledge are explained in [2].

4 Experimental Methodology

A single training set is randomly taken from the original sample space, i.e. the full data set. The K-NN classifier built using this original learning set is referred to as the single expert. The remaining samples are used as a test set. Using the learning set, 25 boot sets are generated, by bootstrapping. The decision of the 25 boot sets are aggregated to classify the test set. These results are referred to as the bagged expert results. We compare these results to those obtained from the single expert, and to those obtained from other bagging methods. The above is repeated for four training set sizes. The sizes used were 10, 20, 40, and 80 samples.

We measure the performance of the four methods of creating bootstrap sets for two types of learning sets. In the first case the learning set is created by randomly taking samples from the full data set. The second type of learning set is constructed using Method 3. The tables show only the results for the regular learning set because both learning sets yielded approximately similar results.

All experiments are repeated 100 times and we average the error rates by dividing by the number of repetitions.The whole experiment is repeated using four different moderation methods. We aim to compare these results with the results obtained using un-moderated kNN experts.

4.1 Calculation of errors

To find the misclassification error rate, a test sample is presented to the K-NN classifier. In the un-moderated kNN case the class posterior probabilities $P(\omega_i/x)$ for each test sample are estimated as:

$P(\omega_i/x) = \frac{\kappa_i}{K}$

where κ_i is the number of neighbors from the i-th class among K. While for the moderated case $P(\omega_i/x) = \frac{\kappa_i+j}{K+2j}$, where j is either 1 as in moderation method 1 (Mod1), or is found using moderation methods 2, 3 or 4.

The test sample is assigned a class label that corresponds to the largest posterior probability. If the original label of the sample is found to be different from the assigned label the error counter is incremented.

4.2 Modified Product

MProduct [4] is a method of combining classifier outputs, that reduces the veto effect. The veto effect is caused by small classifier measurement output values being multiplied by each other to give a near zero result. The MProduct fusion is defined as follows:

1- For each class ω_i, we find the number z of experts the output of which is below a threshold t.

2- If z is less than half the number of experts R, then the experts outputs that are below the threshold value are modified by setting their output to the threshold.

3- If z=0 or is larger than half the number of experts, then the experts outputs remain unaltered.

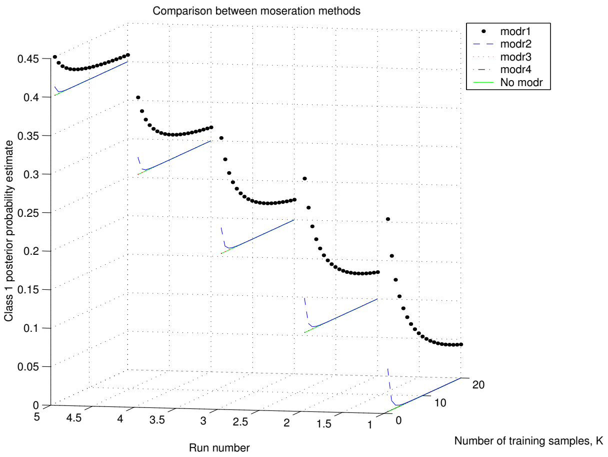

Figure 1: Class 1 posterior probability estimate of different moderation methods for increasing values of K.

The results obtained in this paper are for an MProduct threshold t= 0.0005 which was found to give the best performance on average.

4.3 Comparing Methods

In order to see the effect of Moderation we adopt the simple significance measure of equation 5. It converts the difference between two results, for example results obtained using moderated and un-moderated systems, into a significance of difference measure. This way we will not be misled to ignore small differences when high classification rates exist.

$$significance = \frac{mr - ur}{100 - ur} * 100 \qquad (5)$$

where
mr is the moderated classification rate, or the new result
ur is the unmoderated classification rate, or the baseline result.

If the improvement or degradation exceeds 5 percent we consider it as significant. This value is calculated for all four bagging methods and each of the two learning set types under varying set sizes.

Table 1: Data sets used and the number of samples available in each data set.

Data Name	No. of samples	No. of features	No. of classes
Synthetic	1231	2	2
Seismic	300	25	3
Breast cancer	699	9	2
Wine	178	13	3
Iris	150	4	3

4.4 Data sets

Both synthetic and real data sets were used in our experiments. Synthetic data was chosen to carry out controlled experiments for which the achievable recognition rate is known. The computer generated data is two dimensional involving two classes. The two class densities have an overlap area which was designed to achieve an instability of the class boundary. The features of one dimension were drawn from a uniform distribution. Features of the second dimension have a density that is uniform in a non overlapping area, and a ramp in an overlapping area. The densities of the two classes are symmetric. The ramp was approximated using a uniform random number generator, by cutting the ramp into very small bins. The theoretical Bayes error of this data set is 6.67%. Using the generated samples the empirical Bayes error was found to be 6.82%.

Most of the real data used were the standard sets obtained from the UCI repository [5]. The exception is the seismic data set made available by Shell. Table 1 summarizes the essential information on these data sets.

5 Results

5.1 Results of Moderation I

Moderation is not expected to affect the performance of the single kNN expert. However, Product was the most affected strategy. Moderation caused a major improvement of Prod-

Table 2: Moderated Product compared to unmoderated MProduct, using small bias control threshold. For each data set the four raws are for the four bagging methods from 1 to 4

Data	No. of learning samples			
set	10	20	40	80
Seis.	2.07	1.98	-3.79	-3.40
	-11.45	-0.19	-4.36	-0.49
	-0.11	-2.39	-0.75	-0.54
	-13.21	-1.77	-4.14	-2.50
Wine	-0.22	-0.11	0.59	0.46
	-1.85	1.11	-0.30	0.31
	1.16	0.38	0.42	0.47
	-9.37	-0.87	0.96	0.15
BCW	-1.23	1.04	-0.28	-0.29
	2.75	-0.16	-0.15	-0.36
	-0.56	0.28	-0.41	-0.11
	0.19	**-5.83**	0.06	-0.18
Iris	-0.47	**-4.97**	-0.89	0
	-0.65	-0.40	1.04	0.35
	-0.18	-1.19	2.15	0.00
	-22.41	-1.15	0	0.73
Synth.	-1.86	0.60	0.30	-0.08
	0.80	0.99	0.08	0.22
	0.32	0.15	0.55	-0.14
	0.22	0.55	0.41	-0.13

ucts performance. Results of all datasets improved. As mentioned in the introduction Moderation is based on a similar principle as MProduct but it uses another probability estimate modification strategy. Therefore, it may not yield Product results equivalent to un-moderated MProduct. To answer this question we compare non-moderated MProduct results with moderated Product results obtained using Mod1, table 2. We find that MProduct is mostly better. However the difference is not significant except at 10 instances where MProduct outperformed Product. This indicates that moderation method 1 is less effective than MProduct.

5.2 Results of Moderation II

For the Product fusion experiment a comparison between Mod1 and Mod2 indicates that Mod2 is mostly similar to Mod1 with some exceptions. Mod2 is significantly better at 14 cases in which Mod1 was insignificantly worse than un-moderated MProduct in addition to the 12 cases at which Mod1 was significantly worse than the un-moderated MProduct. This raised expectations that it would be able to outperform the un-moderated MProduct.

Comparing moderated Product using Mod2 to non-moderated MProduct, table 3, we notice that both methods perform similarly and all differences are insignificant. However at the larger BCT mod2 is significantly better when Seismic data is used at size 3 for bagging method 3 when either learning set type was used. Also at size 4 for bagging method 4 using modified learning set. Mod2 was significantly better when Iris data was used at size 1 for bagging method 4 using regular learning set. At larger sizes it is mostly insignificantly better.

Therefore, we conclude that un-moderated MProduct is never significantly better than Product when mod2 is used. Moderation method 2 of Product yields the best results, and the shortcomings of Mod1 are resolved.

Mod 3 was tested but did not yield better results than moderation method 2. The same holds for Mod4, there results were somewhat similar to Mod1. Table **??** show the amount of significant improvement of bagging, when Mod2 is used with Product, over the learning set.

6 Discussion

Mod1 was successful in overcoming the veto effect, however it was not able to outperform MProduct. This was due to the large constant added to experts decisions. This constant has a considerable altering effect on the probability estimates when small K is used. Consequently Mod1 was not able to outperform MProduct at small set sizes, where K is small. However, Mod2 was able to outperform MProduct because it adds a small constant when K is small, and a larger constant as K increases. Moderation did not have any effect on the performance of the single ex-

Table 3: Moderated Product using method 2 compared to un-moderated MProduct

Data	No. of learning samples			
set	10	20	40	80
Seis.	0	1.44	2.23	4.85
	2.58	-0.19	-0.23	0.97
	-0.06	1.34	3.01	1.08
	1.65	0.77	1.74	2.00
Wine	-0.06	-0.11	0.29	0.00
	-0.90	-0.43	-0.59	-0.46
	-0.09	-0.15	0.32	0.16
	2.57	-0.81	-0.29	-0.15
BCW	0.02	-1.39	0.31	0.07
	-0.75	0.04	-0.37	0.07
	-0.02	0.08	-0.28	0.00
	0.63	0.12	-0.45	-0.04
Iris	-0.20	-0.63	0.15	-0.37
	-0.65	-0.40	-0.44	-0.71
	0.30	0.00	-1.16	-1.14
	-0.16	0.71	-1.31	0.36
Synth.	0.01	-0.06	-0.68	0.03
	-0.90	-0.27	-0.74	-0.23
	-0.09	0.04	-0.27	-0.14
	0.04	-0.29	-0.49	-0.30

pert or the sum fusion strategy. MProduct was affected insignificantly except for some cases of significant degradation. This was because MProduct depends on zero outputting experts to boost the performance. Moderation does not leave any expert outputting a zero estimate. Hence, when the expert outputs are moderated, MProduct is nothing but Product.

Just moderating the expert outputs by adding a constant to expert outputs is not enough. It is essential to carefully select the added constant. Among the four types of methods used to add a constant the method used by Mod2 was logically and empirically most successful.

7 Conclusion

The veto effect caused by contradicting experts outputting zero probability estimates leads to fusion strategies performing suboptimaly. This can be resolved using Moderation. We compare different moderation methods and suggest an improved moderation method. Tests on different bagging methods on four real data sets indicate that the proposed moderation method improves the performance of Product over MProduct.

References

[1] F. M. Alkoot and J. Kittler. Experimental evaluation of expert fusion strategies. *Pattern Recognition Letters*, 20(11-13):1361–1369, 1999.

[2] F. M. Alkoot and J. Kittler. Population bias control for bagging knn experts. In *In proceedings of Sensor Fusion: Architectures, Algorithms, and Applications V*, Orlando, Fl, USA, 2001. SPIE.

[3] F. M. Alkoot and J. Kittler. Moderating $k - nn$ classifiers. *Pattern Analysis and Applications*, 5(3):326–332, 2002.

[4] F. M. Alkoot and J. Kittler. Modified product fusion. *Pattern Recognition Letters*, 23(8):957–965, 2002.

[5] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998. University of California, Irvine, Dept. of Information and Computer Sciences.

[6] J. Kittler. Combining classifiers: A theoretical framework. *Pattern Analysis and Applications*, 1:18–27, 1998.

[7] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[8] S. Procter and J. Illingworth. Combining hmm classifiers in a handwritten text recognition system. In *IEEE International Conference on Image Processing*, volume 2, pages 934–938. IEEE Comp Soc, Los Alamitos, CA, USA, 1998.