Improving Classifier Fusion via Pool Adjacent Violators Normalization

Gaurav Goswami**, Nalini Ratha†, Richa Singh*, and Mayank Vatsa*
*IIIT-Delhi, India, †IBM Research, USA

***Abstract*—Classifier fusion is a well-studied problem in which decisions from multiple classifiers are combined at the score, rank, or decision level to obtain better results than a single classifier. Subsequently, various techniques for combining classifiers at each of these levels have been proposed in the literature. Many popular methods entail scaling and normalizing the scores obtained by each classifier to a common numerical range before combining the normalized scores using the sum rule or another classifier. In this research, we explore an alternative method to combine classifiers at the score level. The Pool Adjacent Violators (PAV) algorithm has traditionally been utilized to convert classifier match scores to confidence values that model posterior probabilities for test data. The PAV algorithm and other score normalization techniques have studied the same problem without being aware of each other. In this first ever study to combine the two, we propose the PAV algorithm for classifier fusion on publicly available NIST multi-modal biometrics score dataset. We observe that it provides several advantages over existing techniques and find that the interpretation learned by the PAV algorithm is more robust than the scaling learned by other popular normalization algorithms such as min-max. Moreover, the PAV algorithm enables the combined score to be interpreted as confidence and is able to further improve the results obtained by other approaches. We also observe that utilizing traditional normalization techniques first for individual classifiers and then normalizing the fused score using PAV offers a performance boost compared to only using the PAV algorithm.**

I. Introduction

Classification is a popular research area since many functionalities and advantages associated with modern computing systems rely on it. It is a significant component in a multitude of research problems such as object classification, biometrics, speech recognition, weather prediction, financial trend analysis, or natural language processing. A classifier usually takes some form of input from the domain of the problem (e.g., an image, a video, or a text), compares it based on another input or an existing model, and provides some sort of a classification score representing the degree of similarity or dissimilarity between the two inputs to it. Depending on the type of classifier, this score may denote various quantities which may or may not be directly interpretable. While classification by a single classifier can be achieved by comparison of one score to the other, in many cases it is advantageous to utilize multiple classifiers, trained with different parameters, methodologies, or subsets of data. As shown in Fig. 1, in biometrics, classifier fusion helps in cases when the non-ideal images are captured where unimodal classifier alone may not achieve required level of accuracy. Combining multiple classifiers to improve results has been studied extensively in literature [1], [2], [3]. A detailed review of biometrics fusion algorithms can be found in [4]

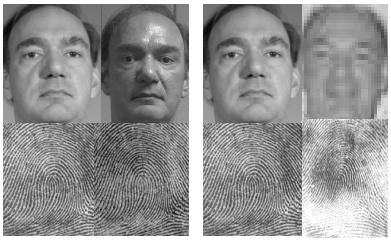
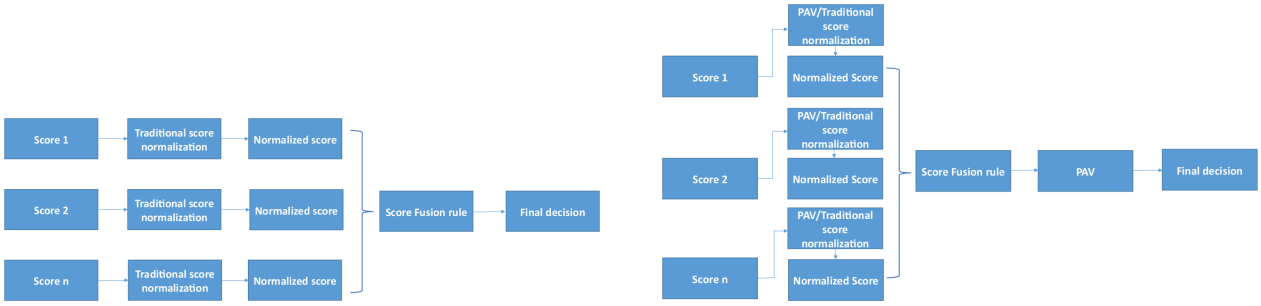


Fig. 1. An illustration to showcase the requirement for fusion in biometrics: (a) fusion not needed when the images are of good *quality*, (b) fusion needed when there are large variations in *quality*.

The traditional classifier fusion framework is illustrated in Fig. 2(a). When multiple classifiers are to be combined, the scores have to be normalized before they are combined such that they lie in a common domain. Therefore, score normalization is an integral part of classification and has received proportionate attention in the literature. A comprehensive review of existing score normalization techniques along with brief descriptions, comparative analysis, and evaluation has been presented by Jain *et al.* [5] where the authors discuss the merits and demerits of popular score normalization techniques such as min-max, z-score, and *tan-h* normalization [6]. There are many other forms of score normalization techniques pertaining to specific applications such as speaker verification [7], metasearch [8], and signature verification [9].

In this paper, we present the Pool Adjacent Violators (PAV) algorithm to improve score normalization techniques for classifier fusion by coupling with traditional score normalization. While the PAV algorithm has been utilized extensively for calibrating classifier scores to probability estimates, its utility in combining classifiers and interaction with traditional score normalization methods has not been explored, to the best of our knowledge. The PAV algorithm [10] has traditionally been utilized for calibrating classifier scores into probability estimates [11]. Fawcett and Niculescu-Mizil [10] demonstrate how the PAV algorithm is functionally equivalent to the Receiver Operating Characteristic Convex Hull (ROCCH) algorithm which tries to find potentially optimal classifiers based on the convex hull of points in ROC space. Aronowitz *et al.* [12]

*This work was completed while Gaurav Goswami was at IBM Research, USA



(a) Traditional (b) Proposed

Fig. 2. An overview of the traditional and the proposed score normalization frameworks for classifier fusion.

have also utilized the PAV algorithm as a method of calibrating classifier scores to probability estimates. In other applications, Brummer and Doddington [13] have used it as an ideal reference to compare against the performance of their proposed prior-weighted proper scoring rules. Mandasari *et al.* [14] have applied the PAV algorithm in the computation of miscalibration cost of their quality measure function for speaker recognition systems. Kim *et al.* [15] have also utilized the PAV algorithm in conjunction with expectation maximization (EM) based thematic clustering to evaluate cluster quality when summarizing topical contents automatically from documents.

As shown in Fig. 2(b), in the proposed framework, two or more scores obtained using different classifiers are first converted to a common numerical range using an appropriate traditional score normalization technique, such as min-max or *tan-h* normalization. Then, the normalized scores are combined using sum rule and the PAV algorithm is used on this combined score to obtain a mapping between the old score values and the new PAV normalized values which are utilized to perform classification. As indicated in the overview, we may also apply PAV as a traditional score normalization to the individual scores. We evaluate both the cases as part of the proposed approach. In this paper, we provide the details of the proposed framework for using PAV as a score normalization technique for combining classifiers, evaluate and analyze its impact on fusion performance, and draw observations about its interactions with the established score normalization techniques which should help the *classification* community.

II. Methodology

First, we present a brief overview of some of the existing score normalization techniques that are utilized for score-level classifier fusion, followed by the details of the PAV algorithm and the proposed framework.

A. Existing Score Normalization Techniques

A number of different methods have been proposed in the literature to address the problem of score normalization, an analytical review of which is presented in [5]. Since not all score normalization techniques offer robustness and the ease of converting data to a common numerical range, in this research, we focus on two popular and effective normalization techniques, namely, min-max and *tan-h* normalization techniques. Min-max normalization is a well known statistical data normalization which scales each data point based on the minimum and maximum values in the score distribution as follows:

$$s' = \frac{s - min(S)}{max(S) - min(S)}, \tag{1}$$

where, S is the score distribution to which the original score s belongs, s' is the new score after normalization, and $min(S)$ and $max(S)$ denote the minimum and maximum value of the score distribution S, respectively. Min-max normalization converts all data to the fixed numeric range of $[0, 1]$ and enables fusion by sum rule for classifiers which originally produce scores in varying data ranges. Tan-h normalization [6] also performs a scaling of the data, but operates using Hampel estimators [6] when determining its statistical scaling parameters to achieve robustness against outliers. It also offers the benefit of converting data to the same fixed numeric range as min-max normalization for every classifier. It is formulated as follows:

$$s' = \frac{1}{2}\left\{\tanh\left(0.01\left(\frac{s-\mu}{\sigma}\right)\right) + 1\right\}, \tag{2}$$

where, s is the original score, s' is the score after normalization, and μ and σ denote the mean and standard deviation estimates, respectively, of the Hampel estimators of the score distribution. The estimator is based on the following influence function:

$$\Psi(m) = \begin{cases} m & 0 \leq |m| < a, \\ a\dot{s}ign(m) & a \leq |m| < b, \\ a\dot{s}ign(m)\left(\frac{c-|m|}{c-b}\right) & b \leq |m| < c, \\ 0 & |m| \geq c. \end{cases} \tag{3}$$

The influence function Ψ processes each score in the distribution and determines its influence on the final estimated distribution. Here, m is a score from the distribution, and a, b, and c are three thresholds that can be adjusted to control the tradeoff between robustness to outliers and efficiency. In this research, we set $a = \sigma$, $b = 2\sigma$, and $c = 3\sigma$, where σ is the standard deviation of the score distribution estimated using median absolute deviation.

B. The Proposed Methodology

While traditional classification consists of predicting a label for given data points during testing, many applications require that the classifier instead provide a score that indicates the likelihood that a data point belongs to a particular class. Usually, the value of the score provides a measure of the classifier's confidence that the data point belongs to that particular label. In the case of binary classification, such scores may be utilized to generate Receiver Operating Characteristic (ROC) curves, which help in evaluating a classifier's performance by varying the acceptance threshold for the true class and recording the true and false accepts. However, not every application scenario can be addressed even with these scores, since each classifier produces them differently and it is difficult to deterministically quantize the difference between two different data points solely based on these scores. For instance, it is not possible to say that one prediction is 36% more accurate than the other if the scores vary by 36%. It is also important to know the actual posterior probability estimate to be able to make decisions efficiently in cost-sensitive scenarios where false accepts might be more penalized than false rejects or vice versa.

Researchers have proposed different methods to address the issue of calibrating the classifier scores to interpretable probability scores. Let C be a binary classifier that maps input data points, X, to real-valued scores S, i.e., $C(X) = S \in \mathbb{R}$. A calibration algorithm creates a function, f, such that, $f(C(X)) = S' \in [0, 1]$, i.e., the new scores S' lie in the range $[0, 1]$ and provide an estimated posterior probability that the each data point in X belongs to the positive class. The PAV algorithm is a non-parametric isotonic regression based method to obtain the desired mapping f, proposed by Zadrozny and Elkan [11]. The core assumption driving the PAV algorithm is that the scores output by the classifier are monotonically increasing with respect to the posterior probability to be estimated. In other words, if a data point has a higher classifier score for the positive class compared to another then it must have a higher probability of belonging to the positive class. It is to be noted that the PAV algorithm simply calibrates the scores to reflect the probabilities rather than an arbitrary value so that quantifiable comparisons such as the one discussed above become possible.

Algorithm 1 PAV algorithm

1: **Input:** Scores and associated labels, (s_i, y_i) with length n
2: **Output:** The probability estimates, S'
3: Sort the scores s_i in increasing order from 0 to n
4: Initialize probability estimates, $s'_i \leftarrow y_i$, and groups $G_i \leftarrow s_i$
5: **while** s'_i is not isotonic **do**
6: **for all** $G_{i-1}, G_i \in G$ **do**
7: **if** $s'_{i-1} > s_i$ **then**
8: Create merged group G_k from G_{i-1}, G_i
9: $\forall s' \in G_k \leftarrow \mu(G_k)$

Given a set of training data consisting of scores and labels, (s_i, y_i), where s_i and y_i are the i^{th} score and associated label respectively, isotonic regression determines the calibration mapping f as follows:

$$f = \arg\min_{z} \sum_{i} (y_i - z(s_i))^2 \tag{4}$$

The learned mapping function, f, can then be utilized to convert classifier scores to probability estimates during testing. The PAV algorithm implements the isotonic regression by a group based methodology. A basic outline of the PAV algorithm is presented in Algorithm 1. Here, $\mu(G_k)$ denotes the mean score value of all instances belonging to the group G_k. First, the algorithm sorts the provided scores and initializes an initial mapping, z_0, with the corresponding labels. Each positive sample is assigned the probability of 1, and each negative sample is assigned a 0 probability. Also, the algorithm creates groups, initially placing each data point in its own group. Then, it iteratively examines the groups in the probability estimates and checks for adjacent violations of the sorted order. By design, the mapping should be isotonic, and hence a violation is essentially when a group appears out of order compared to its adjacent group. To rectify the violation, the algorithm pools together the violating adjacent groups and assigns the average value of the pooled group to all its members. It repeats the process till no violations exist and the mapping has become monotonically increasing with respect to the classifier score.

When combining classifiers, the traditional pipeline involves normalizing the scores, $\{S_1, S_2, ...S_{n_c}\}$, from the involved classifiers using a score normalization technique such as min-max or tan-h normalization and then using the normalized scores, $\left\{S'_1, S'_2, ...S'_{n_c}\right\}$ for fusion. Here, n_c denotes the number of classifiers involved in the fusion. A simple and yet effective fusion strategy utilized in literature is sum rule fusion [4] which essentially aggregates the normalized scores from each classifier to obtain the fused score U for each data instance:

$$U = \sum_{i=1}^{n_c} S'_i \tag{5}$$

We propose the addition of another step in the pipeline (refer Fig. 2), where the fused scores U are then converted to PAV calibrated probability values, P, before being used for determining class label and evaluating classifier performance. Since the PAV algorithm has been shown to be functionally equivalent to the ROCCH algorithm [10], it transforms the scores such that they are both interpretable and optimal as per the ROCCH algorithm [16]. It has been shown in [16] that only a classifier which lies on the convex hull of the points in ROC space can potentially be optimal. Applying the PAV algorithm to transform the fused score helps optimize the performance of the fused classifier and by design makes it lie on the convex hull. However, it is to be noted that while a perfect function mapping would indeed determine an optimal

classifier using the scores, the entire distribution of scores is not known during training. Therefore, the optimality of the learned classifier after applying score normalization and PAV depends on how well the training data represents the actual distribution during testing.

It is to be noted that PAV normalization can not only be applied to the fused score at the end of the pipeline, but also in place of a traditional score normalization technique. As a replacement for a traditional score normalization, it provides robustness to outliers due to the averaging process and also converts data to a common numeric range for each classifier. Applying PAV at both the individual classifier and fused score levels produces a purely PAV normalized score, which we assess and compare with the other alternative normalization techniques in the following section.

III. Results and Analysis

We first provide the details of the database and experimental protocol utilized in the evaluation of the proposed framework. Then, we present the results along with our observations and analysis.

A. Database and Experimental Protocol

In order to evaluate the performance of the proposed approach, we utilize the publicly available NIST Biometrics Scores Set Release 1 (BSSR1) database. This database contains three partitions. Set 1 contains face and fingerprint scores from the same set of 517 individuals using two fingerprint scores (two left index fingerprints (L) and two right index fingerprints (R)) and two separate face matchers (C and G). It contains a total of 534,578 scores. Set 2 contains fingerprint score data for 6000 subjects with two fingerprint scores per subject, one from the comparison of two left index fingerprints (L) and another from the comparison of two right index fingerprints (R). It contains a total of 72,000,000 scores. Set 3 contains face score data for 3000 subjects with two different face matching algorithms (C and G). It contains a total of 36,000,000 scores. We compute and present classifier fusion results on all three partitions of the BSSR1 database. We utilize the symbols C, G, L, and R to denote the two face matchers and two fingerprint scores respectively in the ROC curves presented in the results section.

To emulate a real scenario where not all of the data is available to the score normalization technique at once, we perform 10 fold cross validation on each set of the BSSR1 database. We divide the available scores into 10 partitions, keeping the number of genuine and impostor scores same across each fold. 9 such partitions are used for training each score normalization method, and the remaining partition is used for testing. The process is repeated 10 times and the test partition scores thus obtained are used to evaluate the performance. We report results in form of Receiver Operating Characteristic (ROC) curves for each set, evaluating the genuine accept rate (GAR) at varying false accept rate (FAR) values. For min-max normalization, training involves learning the minimum and maximum value of the score distribution from the training data and using these learned values to perform normalization during test time. For the *tan-h* based normalization technique, learning involves computing the Hampel estimator and its mean and standard deviation from the training data and using those values for the test data. The PAV algorithm learns the calibration function in the form of a mapping between the original scores for each classifier and classifier combination and the corresponding PAV calibrated scores assigned to them during training. In order to estimate the PAV score of an unseen score value during test time, its nearest neighbor is searched in the original score and the corresponding PAV score is assigned to it. Since the number of scores is quite large, especially in the case of set 2 and set 3, this operation can be computationally expensive. We, therefore, down-sample the mapping by retaining only the mean of all the original score values that correspond to a single PAV normalized value. Since the PAV algorithm produces a stepwise function, the number of unique values in the learned PAV scores are much lesser than the entirety of the original scores, leading to a large reduction in the computational requirement at the cost of being highly accurate for boundary cases.

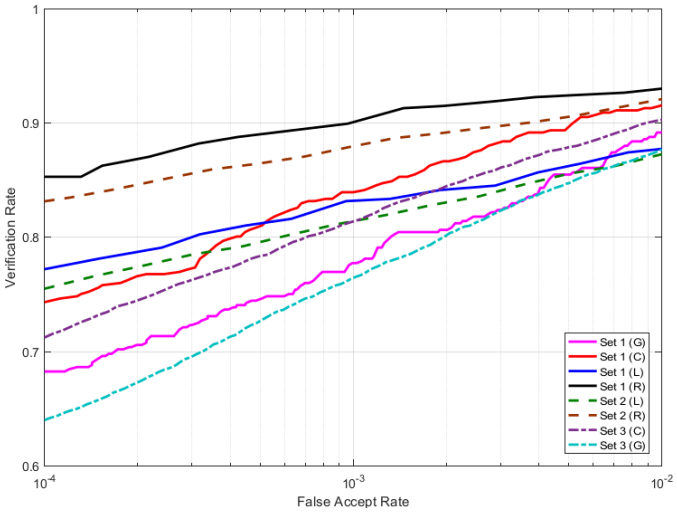


Fig. 3. Baseline performance for individual matchers on all three sets of the NIST BSSR1 database without utilizing any normalization techniques.

B. Experimental Results

Fig. 3 presents the baseline performance by the individual matchers (C, G, L, and R) across all the three sets of the BSSR1 database. The baseline performance for each individual classifier is computed using the original scores as provided in the database without any scaling or normalization technique applied.

Fig. 4 presents the ROC curve for the fusion of all four classifiers (2 face matchers (C and G) + 2 fingerprint scores (L and R)) using the sum rule and different normalization techniques. We observe that the two best performing fusion schemes are those that utilize the PAV normalization on the fused score. We also observe that applying the PAV algorithm boosts the performance of min-max normalization when it is

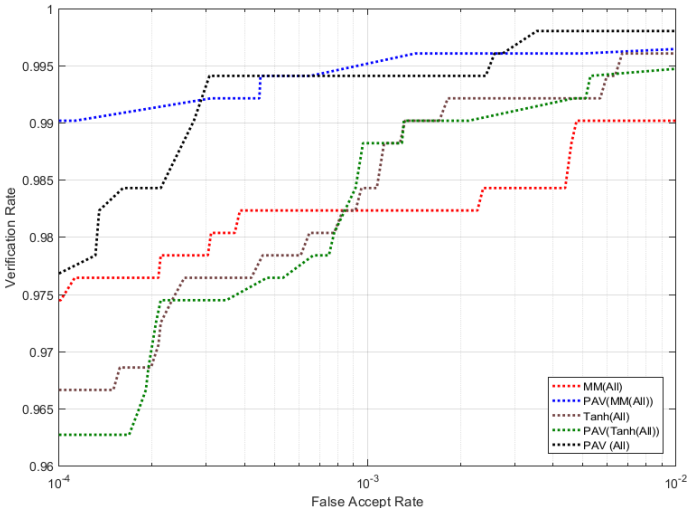


Fig. 4. Fusion of all classifiers on Set 1 of the NIST BSSR1 database.

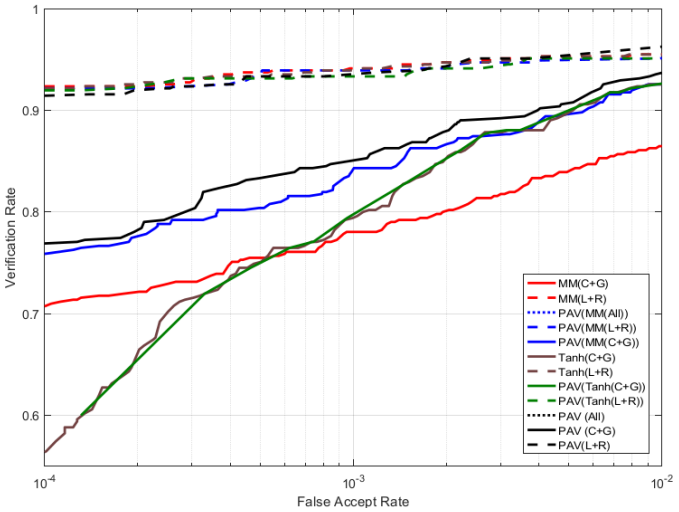
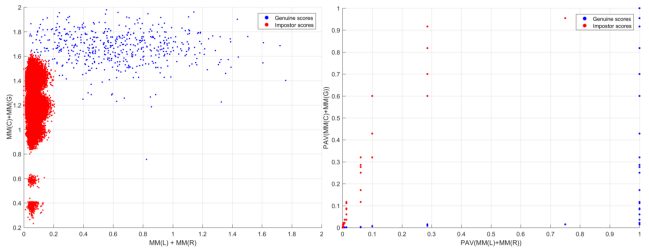


Fig. 6. Fusion of two classifiers on Set 1 of the NIST BSSR1 database.



(a) Min-max (b) PAV + Min-max

Fig. 5. Scatter plots depicting the score distributions for set 1 of the BSSR1 database, (a) before and (b) after applying PAV normalization.

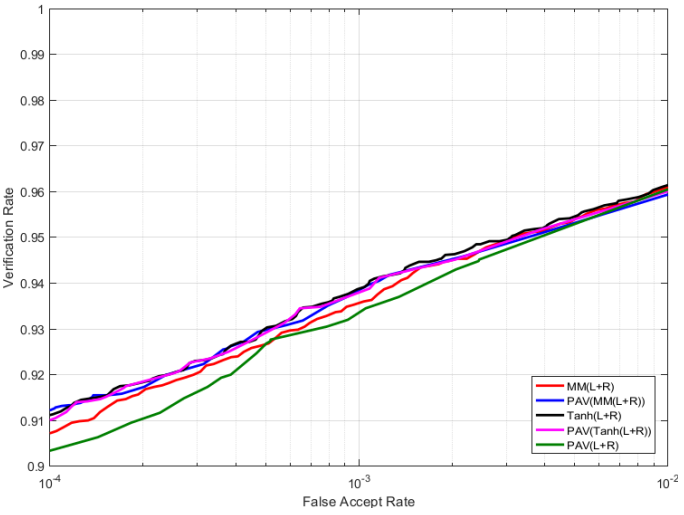


Fig. 7. Fusion of both classifiers on Set 2 of the NIST BSSR1 database.

applied on the individual classifiers. However, we do also note that the tan-h normalization algorithm does not benefit as much from the application of PAV in this case. Fig. 6 presents the results when only two of the face scores or two of the fingerprint scores are fused together. We observe here that the min-max normalization is unable to perform well on the test data due to not knowing the entire score distribution beforehand. On the other hand, applying the proposed PAV normalization improves the performance; utilizing a purely PAV based normalization at both the individual and fused score levels offers the best results for combining the scores from the face matchers. We can also clearly note the convex hull behavior from PAV normalized scores in the case of tan-h normalization for the face matchers, even though it clearly does not fare well with the face matcher scores as compared to using the fingerprint scores. The performance for all combinations is comparable when the two fingerprint matchers are fused together. A scatter plot of the scores after and before PAV normalization for a combination of the two face matchers (C+G) and the two fingerprint matchers (L+R) with each other is presented in Fig. 5. We observe that the general separability of data instances improves and the overlap between the genuine and impostor scores reduces for most instances after applying PAV normalization.

Fig. 7 presents the results for Set 2 with the two fingerprint scores, denoted by L and R. While purely PAV normalization does not perform the best for this scenario, using min-max or *tan-h* normalization first, and applying PAV afterwards to the fused score achieves the best performance while displaying the convex-hull behavior in the ROC space. We note that even if the effectiveness of the purely PAV normalization depends on the type of data distribution, similar to other existing score normalization techniques, PAV normalization applied on any other score normalization technique consistently offers improvements in performance besides converting the fused score to interpretable probability estimates. The performance improvement obtained depends on the training of the PAV mapping and how close the classifier can reach to its convex hull version without the normalization.

Fig. 8 presents the results for Set 3 which contains scores from two face matchers, denoted by C and G. Again, we observe that PAV normalization helps boost the fused perfor-

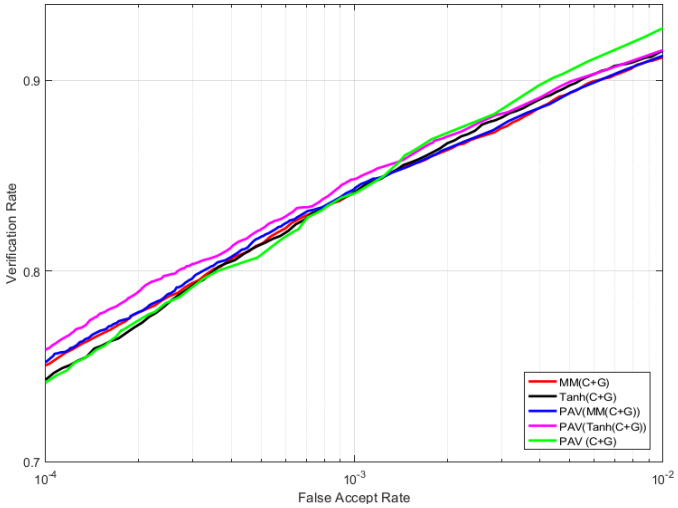


Fig. 8. Fusion of both classifiers on Set 3 of the NIST BSSR1 database.

mance. For the best performing curve, PAV applied over *tan-h* normalization, higher gains are achieved when the original curve is far from its convex hull before the two curves converge at about 0.01 FAR. Purely PAV normalization performs competitively with the other alternatives and becomes the best option for a system looking to operate at 0.01 FAR. PAV does not offer much of a gain in performance for the min-max normalization in this set, but still achieves small gains across FAR intervals where the min-max curve is separated from its convex hull. The key observations are summarized below:

- When PAV is utilized to normalize scores from two classifiers individually, before fusing them with the sum rule, it is similar to any other score normalization algorithm and its performance depends on the quality of its learned mapping and the distribution of the data itself. It offers competitive performance at all times and is the best performing normalization approach for combining the face matchers in Set 1 and at the 0.01 FAR operating point for Set 3.
- When PAV is utilized to normalize fused scores that have been obtained using another normalization algorithm, it offers gains in performance depending on the nature of the curve obtained using the original fused scores. PAV normalization boosts the performance of the fusion, especially when the normalization approach applied to the individual classifiers performs poorly, as we consistently observe from all of the presented results.
- The interpretation learned by the PAV normalization which it utilizes to convert scores to probability estimates can display robustness towards unseen data when the original distribution might change substantially from training to testing. While the scaling learned by the tan-h and min-max normalization algorithms does not hold well for set 1, the PAV normalization algorithm still performs much better.

IV. Conclusion and Future Research Directions

In this paper, we have explored the utility of the PAV algorithm as a score normalization technique for classifier fusion at the score level. The PAV algorithm offers several advantages besides the convenience of normalizing data to a common numeric range. It converts scores to interpretable probability estimates and normalizes scores to achieve a curve closer to the convex hull of the performance curve obtained using the original scores in ROC space, by design. Through a series of experiments on the NIST BSSR1 database, we observe that it also showcases robustness and consistency when utilized as either a normalization technique for the individual classifiers or the fused score obtained using existing normalization techniques. The current approach may be further augmented by exploiting the probabilistic nature of the normalized scores when fusing two classifiers.

References

[1] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE T Pattern Anal.*, vol. 20, no. 3, pp. 226–239, 1998.

[2] K. Nandakumar, Y. Chen, S. C. Dass, and A. K. Jain, "Likelihood ratio-based biometric score fusion," *IEEE T Pattern Anal.*, vol. 30, no. 2, pp. 342–347, 2008.

[3] D. Miao, M. Zhang, Haiqingli, Z. Sun, and T. Tan, "Bin-based weak classifier fusion of iris and face biometrics," in *BTAS*, 2015, pp. 1–6.

[4] A. A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of Multibiometrics (International Series on Biometrics)*, 2006.

[5] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern recogn.*, vol. 38, no. 12, pp. 2270–2285, 2005.

[6] P. Rousseeuw, F. Hampel, E. Ronchetti, and W. Stahel, "Robust statistics: the approach based on influence functions," *J. Wiley, New York*, 1986.

[7] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digit Signal Process.*, vol. 10, no. 1, pp. 42–54, 2000.

[8] M. Montague and J. A. Aslam, "Relevance score normalization for metasearch," in *CIKM*, 2001, pp. 427–433.

[9] J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez, "Target dependent score normalization techniques and their application to signature verification," *IEEE Sys Man Cybern.*, vol. 35, no. 3, pp. 418–425, 2005.

[10] T. Fawcett and A. Niculescu-Mizil, "PAV and the ROC convex hull," *Machine Learning*, vol. 68, no. 1, pp. 97–106, 2007.

[11] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *SIGKDD*, 2002, pp. 694–699.

[12] H. Aronowitz, M. Li, O. Toledo-Ronen, S. Harary, A. Geva, S. Ben-David, A. Rendel, R. Hoory, N. Ratha, S. Pankanti, and D. Nahamoo, "Multi-modal biometrics for mobile authentication," in *IJCB*, 2014, pp. 1–8.

[13] N. Brümmer and G. Doddington, "Likelihood-ratio calibration using prior-weighted proper scoring rules," *arXiv:1307.7981*, 2013.

[14] M. I. Mandasari, R. Saeidi, M. McLaren, and D. A. van Leeuwen, "Quality measure functions for calibration of speaker recognition systems in various duration conditions," *IEEE T Audio Speech.*, vol. 21, no. 11, pp. 2425–2438, 2013.

[15] S. Kim, L. Yeganova, and W. J. Wilbur, "Summarizing topical contents from pubmed documents using a thematic analysis," in *EMNLP*, 2015, pp. 805–810.

[16] F. Provost and T. Fawcett, "Robust classification for imprecise environments," *Machine learning*, vol. 42, no. 3, pp. 203–231, 2001.