THE PAV ALGORITHM OPTIMIZES BINARY PROPER SCORING RULES

NIKO BRÜMMER[†‡] AND JOHAN DU PREEZ[‡]

Abstract. There has been much recent interest in application of the *pool-adjacent-violators (PAV) algorithm* for the purpose of *calibrating* the probabilistic outputs of automatic pattern recognition and machine learning algorithms. Special cost functions, known as *proper scoring rules* form natural objective functions to judge the goodness of such calibration. We show that for binary pattern classifiers, the non-parametric optimization of calibration, subject to a monotonicity constraint, can solved by PAV and that this solution is optimal for all regular binary proper scoring rules. This extends previous results which were limited to *convex* binary proper scoring rules. We further show that this result holds not only for calibration of probabilities, but also for calibration of *log-likelihood-ratios*, in which case optimality holds independently of the prior probabilities of the pattern classes.

Key words. pool-adjacent-violators algorithm, proper scoring rule, calibration

AMS subject classifications. 26A48, 62G08, 62G30, 68W40, 90C99

1. Introduction.

There has been much recent interest in using the *pool-adjacent-violators*[1] (PAV) algorithm for the purpose of *calibration* of the outputs of machine learning or pattern recognition systems [31, 7, 24, 30, 17, 15]. Our contribution is to point out and prove some previously unpublished results concerning the optimality of using the PAV algorithm for such calibration.

In the rest of the introduction, §1.1 defines calibration; §1.2 introduces *regular binary proper scoring rules*, the class of objective functions which we use to judge the goodness of calibration; and §1.3 gives more specific details of how this calibration problem forms the non-parametric, monotonic optimization problem which is the subject of this paper.

The rest of the paper is organized as follows: In §2 we state the main optimization problem under discussion; §3 summarizes previous work related to this problem; §4, the bulk of this paper, presents our proof that PAV solves this problem; and finally §5 shows that the PAV can be adapted to a closely related calibration problem, which has the goal of assigning calibrated *log-likelihood-ratios*, rather than probabilities. We conclude in §6 with a short discussion about applying PAV calibration in pattern recognition.

The results of this paper can be summarized as follows: The PAV algorithm, when used for supervised, monotonic, non-parametric calibration is (i) optimal for all regular binary proper scoring rules and is moreover (ii) optimal at any prior when calibrating log-likelihood-ratios.

1.1. Calibration.

In this paper, we are interested in the calibration of *binary* pattern classification systems which are designed to discriminate between two classes, by outputting a scalar *confidence score*[2]. Let x denote a to-be-classified input pat-

[†]Spescom DataVoice, Stellenbosch, South Africa.

[‡]Digital Signal Processing Group, Department of Electrical and Electronic Engineering, University of Stellenbosch.

[1]a.k.a *pair*-adjacent-violators

[2]The reader is cautioned not to confuse *score* as defined here, with *proper scoring rule* as defined in the next subsection.

tern[3], which is known to belong to one of two classes: the *target* class θ_1, or the *non-target* class θ_2. The pattern classifier under consideration performs a mapping $x \mapsto s$, where s is a real number, which we call the *uncalibrated confidence score*. The only assumption that we make about s is that it has the following *sense*: *The greater the score, the more it favours the target class—and the smaller, the more it favours the non-target class.*

In order for the pattern classifier output to be more generally useful, it can be processed through a calibration transformation. We assume here that the calibrated output will be used to make a minimum-expected-cost Bayes decision [12, 29]. This requires that the score be transformed to act as *posterior probability* for the target class, given the score. We denote the transform of the uncalibrated score s to calibrated target posterior thus: $s \mapsto P(\theta_1|s)$. In the first (and largest) part of this paper, we consider this calibration transformation as an atomic step and show in what sense the PAV algorithm is optimal for this transformation.

In most machine-learning contexts, it is assumed that the object of calibration is (as discussed above) to assign *posterior probabilities* [26, 31, 24]. However, the calibration of *log-likelihood-ratios* may be more appropriate in some pattern recognition fields such as automatic speaker recognition [14, 7]. This is important in particular for *forensic* speaker recognition, in cases where a Bayesian framework is used to represent the weight of the speech evidence in likelihood-ratio form [17]. With this purpose in mind, in §5, we decompose the transformation $s \mapsto P(\theta_1|s)$ into two consecutive steps, thus: $s \mapsto \log \frac{P(s|\theta_1)}{P(s|\theta_2)} \mapsto P(\theta_1|s)$, where the intermediate quantity is known as the *log-likelihood-ratio* for the target, relative to the non-target. The first stage, $s \mapsto \log \frac{P(s|\theta_1)}{P(s|\theta_2)}$, is now the calibration transform and it is performed by an adapted PAV algorithm (denoted PAV-LLR), while the second stage, $\log \frac{P(s|\theta_1)}{P(s|\theta_2)} \mapsto P(\theta_1|s)$, is just standard application of Bayes' rule. One of the advantages of this decomposition is that the log-likelihood-ratio is independent of $P(\theta_1)$, the prior probability for the target class—and that therefore the pattern classifier (which does $x \mapsto s$) and the calibrator (which does $s \mapsto \log \frac{P(s|\theta_1)}{P(s|\theta_2)}$) can both be independent of the prior. The target prior need only be available for the final step of applying Bayes' rule. Our important contribution here is to show that the PAV-LLR calibration is optimal *independently* of the prior $P(\theta_1)$.

1.2. Regular Binary Proper Scoring Rules. We have introduced calibration as a tool to map uncalibrated scores to posterior probabilities, which may then be used to make minimum-expected-cost Bayes decisions. We next ask how the quality of a given calibrator may be judged. Since the stated purpose of calibration is to make cost-effective decisions, the goodness of calibration may indeed be judged by decision cost. For this purpose, we consider a class of special cost functions known as *proper scoring rules* to quantify the cost-effective decision-making ability of posterior probabilities, see e.g. [18, 12, 13, 11, 9, 16], or our previous work [7]. Since this paper is focused on the PAV algorithm, a detailed introduction to proper scoring rules is out of scope. Here we just need to define the class of regular binary proper scoring rules in a way that is convenient to our purposes. (Appendix A gives some notes to link this definition to previous work.)

We define a *regular binary proper scoring rule* (RBPSR) to be a function, C_ρ :

[3]The nature of x is unimportant here, it can be an image, a sound recording, a text document etc.

$\{\theta_1, \theta_2\} \times [0,1] \to [0,\infty]$, such that

$$\mathrm{C}_\rho(\theta_1, q) = \int_q^1 \frac{1}{\eta}\rho(\eta)\, d\eta, \qquad \mathrm{C}_\rho(\theta_2, q) = \int_0^q \frac{1}{1-\eta}\rho(\eta)\, d\eta \tag{1}$$

for which the following conditions must hold:

(i) These integrals exist and are *finite*, except[4] possibly for $\mathrm{C}_\rho(\theta_1, 0)$ and $\mathrm{C}_\rho(\theta_2, 1)$, which may assume the value ∞.

(ii) $\rho(\eta)$ is a probability distribution[5] over $[0,1]$, i.e. $\rho(\eta) \geq 0$ for $0 \leq \eta \leq 1$, and $\int_0^1 \rho(\eta)\, d\eta = 1$.

In other words the RBPSR's are a family of functions parametrized by ρ. If $\rho(\eta) > 0$ almost everywhere, then the RBPSR is denoted *strict*, otherwise it is *non-strict*. We list some examples, which will be relevant later:

1. If $\rho(\eta) = \delta(\eta - \eta')$, where δ denotes Dirac-delta, then $\mathrm{C}_\rho(\cdot, q)$ represents the misclassification cost of making binary decisions by comparing probability q to a threshold of η'. Note that this proper scoring rule is *non-strict*. Moreover it is discontinuous and therefore *not convex* as a function of q. This is but one example of many non-convex proper scoring rules. A more general example is obtained by convex combination[6] of multiple Dirac-deltas: $\rho(\eta) = \sum_i \alpha_i \delta(\eta - \eta'_i)$.

2. If $\rho(\eta) = 6\eta(1-\eta)$, then C_ρ is the (*strict*) quadratic[7] proper scoring rule, also known as the Brier scoring rule [6].

3. If $\rho(\eta) = 1$, then C_ρ is the (*strict*) logarithmic scoring rule, originally proposed by [18].

The salient property of a binary proper scoring rule is that for any $0 \leq p, q \leq 1$, its expectations w.r.t q are minimized at q, so that: $q\,\mathrm{C}_\rho(\theta_1, q) + (1-q)\,\mathrm{C}_\rho(\theta_2, q) \leq q\,\mathrm{C}_\rho(\theta_1, p) + (1-q)\,\mathrm{C}_\rho(\theta_2, p)$. For a strict RBPSR, this minimum is unique. We show below in lemma 6 how this property derives from (1).

1.3. Supervised, monotonic, non-parametric calibration. We have thus far established that we want to find a calibration method to map scores to probabilities and that we then want to judge the goodness of these probabilities via RBPSR. We can now be more specific about the calibration problem that is optimally solvable by PAV:

1. Firstly, we constrain the calibration transformation $s \mapsto P(\theta_1|s)$ to be a *monotonic non-decreasing* function: $\mathbb{R} \to [0,1]$. This is to preserve the above-defined *sense* of the score s. This monotonicity constraint is discussed further in §6. See also [7, 31, 24, 17].

2. Secondly, we assume that we are given a finite number, T, of *trials*, for each of which the to-be-calibrated pattern classifier has produced a score. We denote these scores $s_1, s_2, \ldots s_T$. We need only to map each of these scores to a probability. In other words, we do not have to find the calibration function itself, we only have to *non-parametrically* assign the T function output values $p_1, p_2, \ldots, p_T$, while respecting the above monotonicity constraint. To simplify notation, we assume without loss of generality, that $s_1 \leq s_2 \leq \cdots \leq s_T$. (In practice one has to sort the scores to make

[4] This exception accommodates cases like the logarithmic scoring rule, which is obtained at $\rho(\eta) = 1$, see [11, 16].

[5] It is easily shown that if $\rho(\eta)$ cannot be normalized (i.e. $\int_0^1 \rho(\eta)\, d\eta \to \infty$), then one or both of $\mathrm{C}_\rho(\theta_1, q)$ or $\mathrm{C}_\rho(\theta_2, q)$ must also be infinite for every value of q, so that a useful proper scoring rule is not obtained.

[6] The $\alpha_i > 0$ and sum to 1.

[7] In this context the average of the Brier proper scoring is just a mean-squared-error.

it so.) This now means that monotonicity is satisfied if $0 \leq p_1 \leq p_2 \leq \cdots \leq p_T \leq 1$. Notice that the input scores now only serve to define the order. Once this order is fixed, one does not need to refer back to the scores. The output probabilities can now be independently assigned, as long as they respect the above chain of inequalities.

3. Finally, we assume that the problem is *supervised*: For every one of the T trials the true class is known and is denoted: $\ell_1, \ell_2, \ldots, \ell_T \in \{\theta_1, \theta_2\}$. This allows evaluation of the RBPSR for every trial t as $\mathrm{C}_\rho(\ell_t, p_t)$. A weighted combination of the RBPSR costs for every trial can now be used as the objective function which needs to be be minimized.

In summary the problem which is solved by PAV is that of finding $p_1, p_2, \ldots, p_T$, subject to the monotonicity constraints, so that the RBPSR objective is minimized. This problem is succinctly restated in the following section:

2. Main optimization problem statement. The problem of interest may be stated as follows:

1. We are given as input:

(i) A sequence of T indices, denoted $(1, T) = 1, 2, \ldots, T$ with a corresponding sequence of labels $\ell_1, \ell_2, \ldots, \ell_T \in \{\theta_1, \theta_2\}$.

(ii) A pair of positive weights, $v_1, v_2 > 0$.

2. We use the notation $v(\ell_t)$ to assign a weight to every index, by letting $v(\theta_1) = v_1$ and $v(\theta_2) = v_2$.

3. The problem is now to find the sequence of T probabilities, denoted $\mathbf{p}_{1,T} = p_1, p_2, \ldots, p_T$, which minimizes the following *objective*:

$$\mathcal{O}_{1,T}(\mathbf{p}_{1,T}) = \sum_{t=1}^{T} v(\ell_t)\, \mathrm{C}_\rho(\ell_t, p_t), \tag{2}$$

subject to the *monotonicity constraint*:

$$0 \leq p_1 \leq p_2 \leq \cdots \leq p_T \ \leq 1 \tag{3}$$

We require the solution to hold (be a feasible minimum) *simultaneously* for every RBPSR C_ρ. We already know that if such a solution exists, it must be unique, because the original PAV algorithm as published in [4] in 1955, was shown to give a unique optimal solution for the special case of $\rho(\eta) = 1$, for which $\big(\mathrm{C}_\rho(\theta_1, p), \mathrm{C}_\rho(\theta_2, p)\big) = \big(-\log(p), -\log(1-p)\big)$. See theorem 1 and corollary 2 below for details.

3. Relationship of our proof to previous work. Although not stated explicitly in terms of a proper scoring rule, the first publication of the PAV algorithm [4], was already proof that it optimized the logarithmic proper scoring rule. It is also known that PAV optimizes the quadratic (Brier) scoring rule [31], and indeed that it optimizes combinations of more general convex functions [5, 2]. However as pointed out above, there are proper scoring rules that are not convex.

In our previous work [7], where we made use of calibration with the PAV algorithm, we did mention the same results presented here, but without proof. This paper therefore complements that work, by providing proofs.

We also note that independently, in [15], it was stated "it can be proved that the same [PAV algorithm] is obtained when using any proper scoring function", but this was also without proof or further references[8].

[8] Notes to reviewers: Note 1: We contacted Fawcet and Niculescu-Mizil to ask if they had a proof.

We construct a proof that the PAV algorithm solves the problem as stated in §2, by roughly following the pattern of the unpublished document [1], where the optimality of PAV was proved for the case of *strictly convex* cost functions. That proof is not applicable as is for our purposes, because as pointed out above, some RBPSR's are not convex. We will show however in lemma 6 below, that all RBPSR's and their expectations are *quasiconvex* and that the proof can be based on this quasiconvexity, rather than on convexity. Note that when working with convex cost functions, one can use the fact that positively weighted combinations of convex functions are also convex, but this is not true in general for quasiconvex functions. For our case it was therefore necessary to prove explicitly that expectations of RBPSR's are also quasiconvex. A further complication that we needed to address was that non-strict RBPSR's lead to unidirectional implications, in places where the strictly convex cost functions of the proof in [1] gave *if and only if* relationships.

Finally, we note that although the more general case of PAV for non-strict convex cost functions was treated in [5], we could not base our proof on theirs, because they used properties of convex functions, such as subgradients, which are not applicable to our quasiconvex RBPSR's.

4. Proof of optimality of PAV. This section forms the bulk of this paper and is dedicated to prove that a version of the PAV algorithm solves the optimization problem stated in §2.

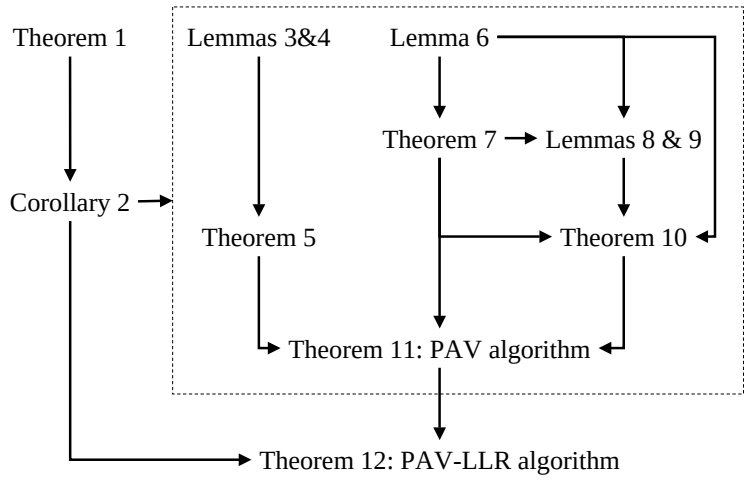

FIG. 1. *Proof structure: PAV is optimal for all RBPSR's and PAV-LLR is optimal for all RBPSR's and priors.*

They replied that their statement was based on the assumption that proper scoring rules are convex, which by [5] is then optimized by PAV. Since we include here also non-convex proper scoring rules, our results are more general. Note 2: The paper [28] has the word 'quasi-convex' in the title and employs the PAV algorithm for a solution. This could suggest that our problem was solved in that paper, but a different problem was solved there, namely: "the approximation problem of fitting n data points by a quasi-convex function using the least squares distance function."

See figure 1 for a roadmap of the proof: Theorem 1 and corollary 2 give the closed-form solution for the logarithmic RBPSR. For the PAV algorithm, we use corollary 2 just to show that there is a unique solution, but we re-use it later to prove the prior-independence of the PAV-LLR algorithm. Inside the dashed box, theorem 5 shows how multiple optimal *subproblem* solutions can constitute the optimal solution to the whole problem. Theorems 7 and 10 respectively show how to find and combine optimal subproblem solutions, so that the PAV algorithm can use them to meet the requirements of theorem 5.

4.1. Unique solution. In this section, we use the work of Ayer et al, reproduced here as theorem 1, to show via corollary 2 that, if our problem does have a solution for every RBPSR, then it must be unique, because the special case of the logarithmic scoring rule (when $\rho(\eta) = 1$) does have a unique solution.

THEOREM 1 (Ayer et al., 1955). *Given non-negative real numbers a_t, b_t, such that $a_t + b_t > 0$ for every $t = 1, 2, \ldots, T$, the maximization of the objective $\mathcal{O}'_{1,T}(\mathbf{p}_{1,T}) = \prod_{t=1}^{T}(p_t)^{a_t}(1-p_t)^{b_t}$, subject to the monotonicity constraint* (3)*, has the unique solution, $\mathbf{p}_{1,T} = p_1, p_2, \ldots, p_T$, where:*

$$\begin{aligned} p_t &= \max_{1\le i\le t}\ \min_{t\le j\le T} r'_{i,j} \\ &= \min_{t\le j\le T}\ \max_{1\le i\le t} r'_{i,j}, \end{aligned} \tag{4}$$

where

$$r'_{i,j} = \frac{\sum_{k=i}^{j} a_k}{\sum_{k=i}^{j} a_k + b_k} \tag{5}$$

Proof. See[9] [4], theorem 2.2 and its corollary 2.1. In that work, the monotonicity constraint was non-increasing, rather than the non-decreasing constraint (3) that we use here. The solution that they give therefore has to be transformed by letting the index t go in reverse order, which means exchanging the roles of the subsequence endpoints i, j, which then has the result of exchanging the roles of max and min in the solution. □

We now show that this theorem supplies the solution for the special case of the logarithmic RBPSR:

COROLLARY 2. *If $\big(\mathrm{C}_\rho(\theta_1, p), \mathrm{C}_\rho(\theta_2, p)\big) = \big(-\log(p), -\log(1-p)\big)$, then the problem of minimizing objective* (2)*, subject to constraint* (3)*, has the unique solution, $\mathbf{p}_{1,T} = p_1, p_2, \ldots, p_T$, where:*

$$\begin{aligned} p_t = \mathrm{PAV}_t\big((\ell_1, \ell_2, \ldots, \ell_T), (v_1, v_2)\big) &= \max_{1\le i\le t}\ \min_{t\le j\le T} r_{i,j} \\ &= \min_{t\le j\le T}\ \max_{1\le i\le t} r_{i,j}, \end{aligned} \tag{6}$$

where

$$r_{i,j} = \frac{m_{i,j} v_1}{m_{i,j} v_1 + n_{i,j} v_2} \tag{7}$$

where $m_{i,j}$ is the number of θ_1-labels and $n_{i,j}$ the number of θ_2-labels in subsequence $\ell_i, \ell_{i+1}, \ldots, \ell_j$.

[9] Available online (with open access) at `http://projecteuclid.org/euclid.aoms/1177728423`.

Proof. Observe that if we let

$$(a_t, b_t) = \begin{cases} (v_1, 0), & \text{if } \ell_t = \theta_1, \\ (0, v_2), & \text{if } \ell_t = \theta_2, \end{cases}$$

then $r'_{i,j} = r_{i,j}$, so that $\mathcal{O}'_{1,T}(\mathbf{p}_{1,T}) = \exp\bigl(-\mathcal{O}_{1,T}(\mathbf{p}_{1,T})\bigr)$, so that the constrained maximization of theorem 1 and the constrained minimization of this corollary have the same solution. □

This corollary gives a closed-form solution, (6), to the problem, and from [4] we know that this is the same solution which is calculated by the iterative PAV algorithm[10]. As noted above, it has so far [4, 1, 5] only been shown that this solution is valid for logarithmic and other RBPSR's which have *convex* expectations. In the following sections we show that this solution is also optimal for all other RBPSR's.

4.2. Decomposition into subproblems. We need to consider *subsequences* of $(1, T)$: For any $1 \le i \le j \le T$, we denote as (i, j) the subsequence of $(1, T)$ which starts at index i and ends at index j. We may compute a partial objective function over a subsequence (i, j) as:

$$\mathcal{O}_{i,j}(\mathbf{p}_{i,j}) = \sum_{t=i}^{j} v(\ell_t)\, \mathrm{C}_\rho(\ell_t, p_t). \tag{8}$$

where $\mathbf{p}_{i,j} = p_i, p_{i+1}, \ldots, p_j$. We can now define the *subproblem* (i, j) as the problem of minimizing $\mathcal{O}_{i,j}(\mathbf{p}_{i,j})$, simultaneosly for every RBPSR, and subject to the monotonicity constraint $0 \le p_i \le p_{i+1} \le \cdots \le p_j \le 1$. In what follows, we shall use the following notational conventions:

1. The subproblem $(1, T)$ is equivalent to the original problem.

2. We shall denote a subproblem solution, $\mathbf{p}_{i,j}$, as *feasible* when the monotonicity constraint is met and *non-feasible* otherwise.

3. By *subproblem solution* we mean just a sequence $\mathbf{p}_{i,j}$, feasible or not, such that $p_i, p_{i+1}, \ldots, p_j \in [0, 1]$.

4. Since any subproblem is isomorphic to the original problem, corollary 2 also shows that if[11] it has a feasible minimizing solution for every RBPSR, then that solution must be unique. Hence, by *the optimal subproblem solution*, we mean the unique feasible solution that minimizes $\mathcal{O}_{i,j}(\cdot)$, for every RBPSR.

5. By a *partitioning* of the problem $(1, T)$ into a set, $\mathcal{S}$, of adjacent, non-overlapping subproblems, we mean that every index occurs exactly once in all of the subproblems, so that:

$$\mathcal{O}_{1,T}(\mathbf{p}_{1,T}) = \sum_{(i,j)\in\mathcal{S}} \mathcal{O}_{i,j}(\mathbf{p}_{i,j}) \tag{9}$$

Our first important step is to show with theorem 5, proved via lemmas 3 and 4, how the optimal total solution may be constituted from optimal subproblem solutions:

LEMMA 3. *For a given RBPSR and for a given partitioning, $\mathcal{S}$, of $(1, T)$ into subproblems, let:*

[10] The PAV algorithm, if efficiently implemented, is known [25, 2, 30] to have *linear* computational load (of order T), which is superior to a straight-forward implementation of the explicit form (6).

[11] The object of this whole exercise is to prove that the optimal solution exists for every subproblem and is given by the PAV algorithm, but until we have proved this, we cannot assume that the optimal solution exists for every subproblem.

(i) $\mathbf{p}^*_{1,T} = p^*_1, p^*_2, \ldots, p^*_T$ *be a feasible solution to the whole problem, with minimum total objective* $\mathcal{O}_{1,T}(\mathbf{p}^*_{1,T})$*; and*

(ii) for every subproblem $(i,j) \in \mathcal{S}$*, let* $\mathbf{q}^*_{i,j} = q^*_i, q^*_{i+1}, \ldots, q^*_j$ *denote a feasible subproblem solution with minimum partial objective* $\mathcal{O}_{i,j}(\mathbf{q}^*_{i,j})$*; and*

(iii) $\mathbf{q}^*_{1,T} = q^*_1, q^*_2, \ldots, q^*_T$ *denote the concatenation of all the subproblem solutions* $\mathbf{q}^*_{i,j}$*, in order, to form a (not necessarily feasible) solution to the whole problem* $(1,T)$*, then*

$$\mathcal{O}_{1,T}(\mathbf{q}^*_{1,T}) = \sum_{(i,j)\in\mathcal{S}} \mathcal{O}_{i,j}(\mathbf{q}^*_{i,j}) \leq \sum_{(i,j)\in\mathcal{S}} \mathcal{O}_{i,j}(\mathbf{p}^*_{i,j}) = \mathcal{O}_{1,T}(\mathbf{p}^*_{1,T}). \tag{10}$$

Proof. Follows by recalling (9) and by noting that for every (i,j), $\mathcal{O}_{i,j}(\mathbf{q}^*_{i,j}) \leq \mathcal{O}_{i,j}(\mathbf{p}^*_{i,j})$, because (except at $i = 1$ and $j = T$) minimization of the RHS is subject to the extra constraints $p^*_{i-1} \leq p^*_i$ and $p^*_j \leq p^*_{j+1}$. □

LEMMA 4. *For a given RBPSR and for a given partitioning,* $\mathcal{S}$*, of* $(1,T)$ *into subproblems, let* $\mathbf{p}^*_{1,T} = p^*_1, p^*_2, \ldots, p^*_T$ *be a feasible solution to the whole problem, with minimum total objective* $\mathcal{O}_{1,T}(\mathbf{p}^*_{1,T})$*; and let* $\mathbf{q}_{1,T} = q_1, q_2, \ldots, q_T$ *be any feasible solution to the whole problem, with total objective* $\mathcal{O}_{1,T}(\mathbf{q}_{1,T})$*. Then*

$$\mathcal{O}_{1,T}(\mathbf{q}_{1,T}) = \sum_{(i,j)\in\mathcal{S}} \mathcal{O}_{i,j}(\mathbf{q}_{i,j}) \geq \mathcal{O}_{1,T}(\mathbf{p}^*_{1,T}). \tag{11}$$

Proof. Follows directly from (9) and the premise. □

THEOREM 5. *Let* $\mathbf{q}^*_{1,T} = q^*_1, q^*_2, \ldots, q^*_T$ *be a* feasible *solution for* $(1,T)$ *and let* $\mathcal{S}$ *be a partitioning of* $(1,T)$ *into subproblems, such that for every* $(i,j) \in \mathcal{S}$*, the subsequence* $\mathbf{q}^*_{i,j} = q^*_i, q^*_{i+1}, \ldots, q^*_j$ *is the optimal solution to subproblem* (i,j)*, then* $\mathbf{q}^*_{1,T}$ *is the optimal solution to the whole problem* $(1,T)$*.*

Proof. The premises make lemmas 3 and 4 applicable, for every RBPSR. Since both inequalities (10) and (11) are satisfied, $\mathcal{O}_{1,T}(\mathbf{q}^*_{1,T}) = \mathcal{O}_{1,T}(\mathbf{p}^*_{1,T})$, where $\mathbf{p}^*_{1,T}$ is an optimal solution for each RBPSR. Hence $\mathbf{q}^*_{1,T}$ is optimal for every RBPSR and is by corollary 2 the unique optimal solution. □

4.3. Constant subproblem solutions. In what follows constant subproblem solutions will be of central importance. A solution $\mathbf{p}_{i,j}$ is constant if $p_i = p_{i+1} = \cdots = p_j = q$, for some $0 \leq q \leq 1$. In this case, we use the short-hand notation $\mathcal{O}_{i,j}(q) = \mathcal{O}_{i,j}(\mathbf{p}_{i,j})$ to denote the subproblem objective, and this may be expressed as:

$$\begin{aligned} \mathcal{O}_{i,j}(q) = \mathcal{O}_{i,j}(\mathbf{p}_{i,j}) &= \sum_{t=i}^{j} v(\ell_t)\, \mathrm{C}_\rho(\ell_t, q) \\ &= mv_1\, \mathrm{C}_\rho(\theta_1, q) + nv_2\, \mathrm{C}_\rho(\theta_2, q), \end{aligned} \tag{12}$$

where m is the number of θ_1-labels and n the number of θ_2-labels. Note:

1. A constant subproblem solution is always *feasible*.
2. If it exists, the optimal solution to an arbitrary subproblem may or may not be constant.

Whether optimal or not, it is important to examine the behaviour of subproblem solutions that are constrained to be constant. This behaviour is governed by the quasiconvex[12] properties of $\mathcal{O}_{i,j}(q)$ as summarized in the following lemma:

[12] A real-valued function $f(p)$, defined on a real interval is *quasiconvex*, if every sublevel set of the form $\{p|f(p) < a\}$ is convex (i.e. a real interval) [3]. Lemma 6 shows that $\mathcal{O}_{i,j}(q)$ is *quasiconvex*.

LEMMA 6. *Let $r_{i,j} = \frac{v_1 m}{v_1 m + v_2 n}$, where m is the number of θ_1-labels and n the number of θ_2-labels in the subsequence (i, j), and let $\mathcal{O}_{i,j}(q) = m v_1 \, \mathrm{C}_\rho(\theta_1, q) + n v_2 \, \mathrm{C}_\rho(\theta_2, q)$ be the objective for the constant subproblem solution, $p_i = p_{i+1} = \cdots = p_j = q$, then the following properties hold, where C_ρ is* any *RBPSR, and where we also note the specialization for strict RBPSR's:*

1. *If $q \le q' \le r_{i,j}$, then $\mathcal{O}_{i,j}(q) \ge \mathcal{O}_{i,j}(q') \ge \mathcal{O}_{i,j}(r_{i,j})$.*

strict case: *If $q < q' \le r_{i,j}$, then $\mathcal{O}_{i,j}(q) > \mathcal{O}_{i,j}(q')$.*

2. *If $q' \ge q \ge r_{i,j}$, then $\mathcal{O}_{i,j}(q') \ge \mathcal{O}_{i,j}(q) \ge \mathcal{O}_{i,j}(r_{i,j})$.*

strict case: *If $q' > q \ge r_{i,j}$, then $\mathcal{O}_{i,j}(q') > \mathcal{O}_{i,j}(q)$.*

3. $\min_q \mathcal{O}_{i,j}(q) = \mathcal{O}_{i,j}(r_{i,j})$,

strict case: *$q = r_{i,j}$ is the unique minimum.*

(This is the salient property *of binary proper scoring rules, which was mentioned above.)*

Proof. For convenience in this proof, we drop the subscripts i, j, letting $r = r_{i,j} = \frac{m v_1}{m v_1 + n v_2}$. The expected value of $\mathrm{C}_\rho(\theta, q)$ w.r.t. probability r is:

$$\begin{aligned} e(q) = \mathbb{E}_{\theta|r}\big\{\mathrm{C}_\rho(\theta, q)\big\} &= \tfrac{1}{m v_1 + n v_2} \mathcal{O}_{i,j}(q) \\ &= r \, \mathrm{C}_\rho(\theta_1, q) + (1 - r) \, \mathrm{C}_\rho(\theta_2, q) \end{aligned} \tag{13}$$

Clearly, if the above properties hold for $e(q)$, then they will also hold for $\mathcal{O}_{i,j}(q)$. We prove these properties for $e(q)$ by letting $q \le q'$ and by examining the sign of $\Delta_e = e(q') - e(q)$: If $q' = q$, then $\Delta_e = 0$. If $q < q'$, then (1) gives:

$$\Delta_e = \int_q^{q'} (\eta - r) \frac{\rho(\eta)}{\eta(1 - \eta)} \, d\eta \tag{14}$$

The non-strict versions of properties 1,2 and 3 now follow from the following observation: Since $\rho(\eta) \ge 0$ for $0 \le \eta \le 1$, the sign of the integrand and therefore of Δ_e depends solely on the sign of $(\eta - r)$, giving:

(i) $\Delta_e \ge 0$, if $r \le q < q'$.
(ii) $\Delta_e \le 0$, if $q < q' \le r$.

If more specifically, $\rho(\eta) > 0$ *almost everywhere*, then for any $0 \le q < q' \le 1$, we have $|\Delta_e| > 0$. In this case, the RBPSR is denoted *strict* and we have:

(i) $\Delta_e > 0$, if $r \le q < q'$.
(ii) $\Delta_e < 0$, if $q < q' \le r$.

which concludes the proof also for the strict cases. □

For now, we need only property 3 to proceed. We use the other properties later. The optimal constant subproblem solution is characterized in the following theorem:

THEOREM 7. *If the optimal solution to subproblem (i, j) is constant, then:*

1. *The constant is $r_{i,j}$.*
2. *For any index k, such that $i \le k \le j$, the following are both true:*

(i) $r_{i,k} \ge r_{i,j}$
(ii) $r_{k,j} \le r_{i,j}$

where $r_{i,k}$ and $r_{k,j}$ are defined in a similar way to $r_{i,j}$, but for the subproblems (i, k) and (k, j).

Proof. Property 1 of this theorem follows directly from property 3 of lemma 6. To prove property 2, we use contradiction: If the negation of 2(i) were true, namely $r_{i,k} < r_{i,j}$, then the non-constant solution $p_i = \cdots = p_k = r_{i,k} < p_{k+1} = \cdots = p_j = r_{i,j}$ would be feasible and (by property 3 of lemma 6) would have lower objective,

namely $\mathcal{O}_{i,k}(r_{i,k}) + \mathcal{O}_{k+1,j}(r_{i,j})$, for any strict RBPSR, than that of the constant solution, namely $\mathcal{O}_{i,k}(r_{i,j}) + \mathcal{O}_{k+1,j}(r_{i,j})$. This contradicts the premise that the optimal solution is constant, so that 2(i) must be true. Property 2(ii) is proved by a similar contradiction. □

4.4. Pooling adjacent constant solutions. This section shows (using lemmas 8 and 9 to prove theorem 10) when and how optimal constant subproblem solutions may be assembled by pooling smaller adjacent constant solutions:

LEMMA 8. *Given a subproblem (i, j), for which the optimal solution is constant (at $r_{i,j}$), we can form the* augmented subproblem, *with the additional constraint that the solution at j must satisfy $p_j \leq \alpha$, for some α such that $0 \leq \alpha < r_{i,j}$. That is, the solution to the augmented subproblem must satisfy $0 \leq p_i \leq p_{i+1} \leq \cdots \leq p_j \leq \alpha < r_{i,j}$. Then the augmented subproblem solution is optimized, for every RBPSR, by the constant solution $p_i = p_{i+1} = \cdots = p_j = \alpha$.*

Proof. Feasible solutions to the augmented subproblem must satisfy either (i) $p_i = \cdots = p_j = \alpha$, or (ii) $p_i < \alpha$. We need to show that there is no feasible solution of type (ii), which has a lower objective value, for any RBPSR, than solution (i).

For a given solution, let k be an index such that $i \leq k \leq j$ and $p_i = p_{i+1} = \cdots = p_k$. By combining the premises of this lemma with property 2(i) of theorem 7, we find: $p_i = \cdots = p_k \leq \alpha < r_{i,j} \leq r_{i,k}$, or more succinctly: $p_i = \cdots = p_k \leq \alpha < r_{i,k}$. Now the monotonicity property 1 of lemma 6 shows that the value of $p_i = \cdots = p_k$, which is optimal for all BPSRs must be as large as allowed by the constraints. This means if we start at $k = i$, then p_i is optimized at the constraint $p_i = p_{i+1}$. Next we set $k = i + 1$ to see that $p_i = p_{i+1}$ is optimized at the next constraint $p_i = p_{i+1} = p_{i+2}$. We keep incrementing k, until we find the optimum for the augmented subproblem at the constant solution $p_i = \cdots = p_j = \alpha$. □

LEMMA 9. *Given a subproblem (i, j), for which the optimal solution is constant (at $r_{i,j}$), we can form the* augmented subproblem, *with the additional constraint that the solution at i must satisfy $\alpha \leq p_i$, for some α such that $r_{i,j} \leq \alpha \leq 1$. That is, the solution to the augmented subproblem must satisfy $r_{i,j} < \alpha \leq p_i \leq p_{i+1} \leq \cdots \leq p_j \leq 1$. Then the augmented subproblem solution is optimized, for every RBPSR, by the constant solution $p_i = p_{i+1} = \cdots = p_j = \alpha$.*

Proof. The proof is similar to that of lemma 8, but here we invoke property 2(ii) of theorem 7, to find: $r_{k,j} < \alpha \leq p_k = \cdots = p_j$ and we use the monotonicity property 2 of lemma 6 to show that the value of $p_k = \cdots = p_j$, which is optimal for all RBPSR's, must be as small as allowed by the constraints. □

THEOREM 10. *Given indices $i \leq k \leq j$ such that the optimal subproblem solutions for the two* adjacent *subproblems, (i, k) and $(k+1, j)$, are both constant and therefore (by theorem 7) have the respective values $r_{i,k}$ and $r_{k+1,j}$, then, whenever $r_{i,k} \geq r_{k+1,j}$, the optimal solution for the pooled subproblem (i, j) is also constant, and has the value $r_{i,j}$.*

Proof. First consider the case $r_{i,k} = r_{k+1,j}$. Since this forms a constant solution to subproblem (i, j), by theorem 7, the optimal solution is $r_{i,j}$.

Next consider $r_{i,k} > r_{k+1,j}$. The solution $p_i = \cdots = p_k = r_{i,k} > p_{k+1} = \cdots = p_j = r_{k+1,j}$ is not feasible. A feasible solution must obey $p_k \leq \alpha \leq p_{k+1}$, for some α. There are three possibilities for the value of α: (i) $\alpha \leq r_{k+1,j}$; (ii) $r_{k+1,j} < \alpha < r_{i,k}$; or (iii) $r_{i,k} \leq \alpha$. We examine each in turn:

(i) If $\alpha \leq r_{k+1,j} < r_{i,k}$, then the left subproblem (i, k) is augmented by the constraint $\alpha < r_{i,k}$, so that lemma 8 applies and it is optimized at the constant solution α, while the right subproblem $(k+1, j)$ is not further constrained and is still optimized

at $r_{k+1,j}$. We can now optimize the total solution for (i,j) by adjusting α: By the monotonicity property 1 of lemma 6, the left subproblem objective and therefore also the total objective for (i,j) is optimized at the upper boundary $\alpha = r_{k+1,j}$. In other words, in this case, the optimum for subproblem (i,j) is a constant solution.

(ii) If $r_{k+1,j} < \alpha < r_{i,k}$, then lemma 8 applies to the left subproblem and lemma 9 applies to the right subproblem, so that both subproblems and therefore also the total objective for (i,j) are all optimized at α. In this case also we have a constant solution for (i,j).

(iii) If $r_{k+1,j} < r_{i,k} \leq \alpha$, then the right subproblem is augmented while the left subproblem is not further constrained. We can now use lemma 9 and property 2 of lemma 6, in a similar way to case (i) to show that in this case also, the optimum solution is constant.

Since the three cases exhaust the possibilities for choosing α, the optimal solution is indeed constant and by theorem 7 the optimum is at $r_{i,j}$. □

4.5. The PAV algorithm. We can now use theorems 5, 7 and 10 to construct a proof that a version of the *pool-adjacent-violators* (PAV) algorithm solves the whole problem $(1,T)$.

THEOREM 11. *The PAV algorithm solves the problem stated in §2.*

Proof. The proof is constructive. The strategy is to satisfy the conditions for theorem 5, by starting with optimal constant subproblem solutions of length 1 and then to iteratively combine them via theorem 10, into longer optimal constant solutions until the total solution is feasible. The algorithm proceeds as follows:

input:

(i) labels, $\ell_1, \ell_2, \ldots, \ell_T \in \{\theta_1, \theta_2\}$.

(ii) weights, $v_1, v_2 > 0$.

variables:

(i) $\mathcal{S}$, a partitioning of problem $(1,T)$ into adjacent, non-overlapping subproblems.

(ii) $\mathbf{q}^*_{1,T} = q^*_1, q^*_2, \ldots, q^*_T$, a tentative (not necessarily feasible) solution for problem $(1,T)$.

loop invariant: For every subproblem $(i,j) \in \mathcal{S}$:

(i) The optimal subproblem solution is constant.

(ii) The partial solution $\mathbf{q}^*_{i,j} = q^*_i, q^*_{i+1}, \ldots, q^*_j$ is equal to the optimal subproblem solution, i.e. constant, with value $r_{i,j}$ (by theorem 7).

initialization: Let $\mathcal{S}$ be the finest partitioning into subproblems, so that there are T subproblems, each spanning a single index. Clearly every subproblem (i,i) has a constant solution, optimized at $q^*_i = r_{i,i}$, which is 1, if $\ell_t = \theta_1$, or 0, if $\ell_t = \theta_2$. This initial solution $\mathbf{q}^*_{1,T}$ respects the loop invariant, but is most probably not feasible.

iteration: While $\mathbf{q}^*_{1,T}$ is not feasible:

1. Find any pair of adjacent subproblems, $(i,k),(k+1,j) \in \mathcal{S}$, for which the solutions are equal or violate monotonicity: $r_{i,k} \geq r_{k+1,j}$.

2. Pool (i,k) and $(k+1,j)$ into one subproblem (i,j), by adjusting $\mathcal{S}$ and by assigning the constant solution $r_{i,j}$ to $\mathbf{q}^*_{i,j}$, which by theorem 10 is optimal for (i,j), thus maintaining the loop invariant.

termination: Clearly the iteration must terminate after at most $T-1$ pooling steps, at which time $\mathbf{q}^*_{1,T}$ is now feasible and is still optimal for every subproblem. By theorem 5, $\mathbf{q}^*_{1,T}$ is then the unique optimal solution to problem $(1,T)$. □

5. The PAV-LLR algorithm. The PAV algorithm as presented above finds solutions in the form of *probabilities*. Here we show how to use it to find solutions

in terms of *log-likelihood-ratios*. It will be convenient here to express Bayes' rule in terms of the logit function, $\text{logit}(p) = \log \frac{p}{1-p}$. Note logit is a monotonic rising bijection between $[0, 1]$ and the extended real line. Its inverse is the sigmoid function, $\sigma(w) = \frac{1}{1+e^{-w}}$. Bayes' rule is now [19]:

$$\text{logit}\, P(\theta_1|s_t) = w_t + \pi \tag{15}$$

where the LHS is the *posterior log-odds*, $w_t = \log \frac{P(s_t|\theta_1)}{P(s_t|\theta_2)}$ is the *log-likelihood-ratio*, and $\pi = \text{logit}\, P(\theta_1)$ is the *prior log-odds*.

The problem that is solved by the PAV-LLR algorithm can now be described as follows:

1. There is given:

(i) Labels, $\ell_1, \ell_2, \ldots, \ell_T \in \{\theta_1, \theta_2\}$. We denote as T_1 and T_2 the respective numbers of θ_1 and θ_2 labels in this sequence, so that $T_1 + T_2 = T$.

(ii) Prior log-odds π, where $-\infty < \pi < \infty$. This determines a prior probability distribution for the two classes, namely $\big(P(\theta_1), P(\theta_2)\big) = \big(\sigma(\pi), 1-\sigma(\pi)\big)$, which may be *different* from the label proportions $\big(\frac{T_1}{T}, \frac{T_2}{T}\big)$.

(iii) An RBPSR C_ρ

2. There is required a solution $\mathbf{w}_{1,T} = w_1, w_2, \ldots, w_T$, which minimizes the following objective:

$$\mathcal{O}_{1,T}(\mathbf{w}_{1,T}) = \sum_{t=1}^{T} v(\ell_t)\, \text{C}_\rho(\ell_t, p_t), \tag{16}$$

$$p_t = \sigma(w_t + \pi), \tag{17}$$

$$v_1 = v(\theta_1) = \frac{\sigma(\pi)}{T_1} = \frac{P(\theta_1)}{T_1}, \tag{18}$$

$$v_2 = v(\theta_2) = \frac{1-\sigma(\pi)}{T_2} = \frac{P(\theta_2)}{T_2} \tag{19}$$

(The weights v_1, v_2 are chosen thus[13] to cancel the influence of the proportions of label types, and to re-weight the optimization objective with the given prior probabilities for the two classes, but we show below that this re-weighting is irrelevant when optimizing with PAV.)

3. The minimization is subject to the monotonicity constraint:

$$-\infty \leq w_1 \leq w_2 \leq \cdots \leq w_T \leq \infty, \tag{20}$$

which by the monotonicity of (15) and the logit transformation is equivalent to (3). This problem is solved by first finding the probabilities $p_1, p_2, \ldots, p_T$ via the PAV algorithm and then inverting (17) to find $w_t = \text{logit}(p_t) - \pi$. We already know that the solution is independent of the RBPSR, but remarkably, it is *also* independent of the prior π. This is shown in the following theorem:

THEOREM 12. *Let* $p_t = \text{PAV}_t\big((\ell_1, \ell_2, \ldots, \ell_T), (v_1, v_2)\big)$ *be given by* (6)*, then the problem of minimizing objective* (16)*, subject to monotonicity constraint* (20) *has the*

[13]This kind of class-conditional weighting has been used in several formal evaluations of the technologies of *automatic speaker recognition* and *automatic language recognition*, to weight the error-rates of hard recognition decisions [20, 22] and more recently to also weight logarithmic proper scoring of recognition outputs in log-likelihood-ratio form [7, 27, 23].

unique solution:

$$w_t = \text{logit}\,\text{PAV}_t\big((\ell_1, \ell_2, \ldots, \ell_T), (1,1)\big) - \text{logit}\,\frac{T_1}{T} \tag{21}$$

This solution is simultaneously optimal for every RBPSR, C_ρ*, and any prior log-odds,* $-\infty < \pi < \infty$*.*

Proof. By the properties of the PAV as proved in §4.5 and since logit is a strictly monotonic rising bijection, it is clear that for all RBPSR's and for a given π, this minimization is solved as

$$w_t = \text{logit}\,\text{PAV}_t\big((\ell_1, \ell_2, \ldots, \ell_T), (v_1, v_2)\big) - \pi \tag{22}$$

where π determines v_1 and v_2 via (18) and (19). By corollary 2, we can write component t of this solution, in closed form:

$$\begin{aligned} w_t &= \text{logit}\left(\max_{1\le i\le t}\ \min_{t\le j\le T} r_{i,j}\right) - \pi \\ &= \max_{1\le i\le t}\ \min_{t\le j\le T} \text{logit}\, r_{i,j} - \pi \end{aligned} \tag{23}$$

Now observe that:

$$\begin{aligned} \text{logit}\, r_{i,j} &= \text{logit}\,\frac{v_1 m_{i,j}}{v_1 m_{i,j} + v_2 n_{i,j}} \\ &= \text{logit}\,\frac{m_{i,j}}{m_{i,j} + n_{i,j}} - \text{logit}\,\frac{T_1}{T} + \pi, \end{aligned} \tag{24}$$

which shows that w_t is independent of π. Now the prior may be conveniently chosen to equal the label proportion, $\pi = \text{logit}\,\frac{T_1}{T}$, to give an un-weighted PAV, with $v_1 = v_2 = 1$. □

6. Discussion. We have shown that the problem of monotonic, non-parametric calibration of binary pattern recognition scores is optimally solved by PAV, for all regular binary proper scoring rules. This is true for calibration in posterior probability form and also in log-likelihood-ratio form.

We conclude by addressing some concerns that readers may have about whether the optimization problem solved here is actually useful in real pattern recognition practice, where a calibration transform is trained in a supervised way (as here) on some training data, but is then utilized later on *new unsupervised* data.

The first concern we address is about the non-parametric nature of the PAV mapping, because for general real scores there will be new unmapped score values. An obvious solution is to map new values by interpolating between the (input,output) pairs in the PAV solution and this was indeed done in several of the references cited in this paper (see e.g. [30] for an interpolation algorithm).

Another concern is that the PAV mapping from scores to calibrated outputs has flat regions (all those constant subproblem solutions) and is therefore not an invertible transformation. Invertible transformations are information-preserving, but non-invertible transformations may lose some of the relevant information contained in the input score. This concern is answered by noting that expectations of proper scoring rules are generalized information measures [12, 11] and that in particular the expectation of the logarithmic scoring rule is equivalent to Shannon's cross-entropy

information measure [10]. So by optimizing proper scoring rules, we are indeed optimizing the information relevant to discriminating between the two classes. Also note that a *strictly* monotonic (i.e. invertible) transformation can be formed by adding an arbitrarily small strictly monotonic perturbation to the PAV solution. The PAV solution can be viewed as the argument of the infimum of the RBPSR objective, over all strictly rising monotonic transformations.

In our own work on calibration of speaker recognition log-likelihood-ratios [8], we have chosen to use strictly monotonic rising *parametric* calibration transformations, rather than PAV. However, we then do use the PAV calibration transformation in the supporting role of *evaluating* how well our parametric calibration strategies work. In this role, the PAV forms a well-defined reference against which other calibration strategies can be compared, since it is the best possible monotonic transformation that can be found on a given set of supervised evaluation data. It is in this evaluation role, that we consider the optimality properties of the PAV to be particularly important.

For details on how we employ PAV as an evaluation tool[14], see [7, 21].

Acknowledgments. We wish to thank Daniel Ramos for hours of discussing PAV and calibration, and without whose enthusiastic support this paper would not have been written.

Appendix A. Note on RBPSR family. Some notes follow, to place our definition of the RBPSR family, as defined in §1.2 in context of previous work. Our *regularity* condition (i), directly below (1), is adapted from [11, 16]. General families of binary proper scoring rules have been represented in a variety of ways (see [16] and references therein), including also integral representations that are very similar (but not identical in form) to our (1). See for example [13], where the form $\int_q^1 \rho'(\eta)\,d\eta$, $\int_0^q \frac{\eta}{1-\eta}\rho'(\eta)\,d\eta$ was used; or [9, 16] where $\int_q^1 (1-\eta)\rho''(\eta)\,d\eta$, $\int_0^q \eta\rho''(\eta)\,d\eta$ was used. Equivalence to (1) is established by letting $\rho'(\eta) = \frac{\rho(\eta)}{\eta}$ and $\rho''(\eta) = \frac{\rho(\eta)}{\eta(1-\eta)}$. The advantage of the form (1) which we adopt here, is that the weighting function $\rho(\eta)$ is always in the form of a normalized probability density, which gives the natural interpretation of *expectation* to these integrals.

The reader may notice that it is easy (e.g. by applying an affine transform to (1)) to find a binary proper scoring rule which satisfies the properties of lemma 6, but which is not in the family defined by (1). There are however equivalence classes of proper scoring rules, where the members of a class are all equivalent for making minimum-expected-cost Bayes decisions [12, 11]. Elimination of this redundancy allows normalization of arbitrary proper scoring rules in such a way that the family (1) becomes representative for the members of these equivalence classes [7].

REFERENCES

[1] R.K. Ahuja and J.B. Orlin, "Solving the Convex Ordered Set Problem with Applications to Isotone Regression", Sloan School of Management, MIT, SWP#3988, February 1998, retrieved online from `http://www.mit.edu/bitstream`.

[2] R.K. Ahuja and J.B. Orlin, "A fast scaling algorithm for minimizing separable convex functions subject to chain constraints," Operations Research, 49, 2001, pp. 784–789.

[3] M. Avriel, W.E. Diewert, S. Schaible and I. Zang, *Generalized Concavity*, Plenum Press, 1988.

[14] Our PAV-based evaluation tools are available as a free MATLAB toolkit here: `http://www.dsp.sun.ac.za/~nbrummer/focal/`

[4] Miriam Ayer, H.D. Brunk, G.M. Ewing, W.T. Reid and Edward Silverman, "An Empirical Distribution Function for Sampling with Incomplete Information", Ann. Math. Statist. Volume 26, Number 4, 1955, pp.641–647.
[5] M.J. Best et al, "Minimizing Separable Convex Functions Subject to Simple Chain Constraints", SIAM J. Opim., Vol. 10, No. 3, pp. 658–672, 2000.
[6] G.W. Brier,"Verification of forecasts expressed in terms of probability.", Monthly Weather Review, 78, 1950, pp.1–3.
[7] N. Brümmer and J.A. du Preez, "Application-independent evaluation of speaker detection", Computer Speech & Language, Volume 20, Issues 2-3, April–July 2006, pp.230–275.
[8] N. Brümmer et al., "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006", IEEE Transactions on Audio, Speech, and Language Processing, vol.15, no.7, 2007, pp.2072–2084.
[9] A. Buja, W. Stuetzle, Yi Shen, "Loss Functions for Binary Class Probability Estimation and Classification: Structure and Applications", 2005, online at `www.wharton.upenn.edu/buja`.
[10] T.M. Cover and J.A. Thomas, *Elements of information theory*, 1st Edition. New York: Wiley-Interscience, 1991.
[11] A.P. Dawid, "Coherent Measures of Discrepancy, Uncertainty and Dependence, with Applications to Bayesian Predictive Experimental Design", Technical Report, online at `http://www.ucl.ac.uk/Stats/research/Resrprts/abs94.html#139`, 1998.
[12] M.H. DeGroot, *Optimal Statistical Decisions*. New York: McGraw-Hill, 1970.
[13] M.H. DeGroot and S. Fienberg, "The Comparison and Evaluation of Forecasters", The Statistician 32, 1983.
[14] G. Doddington, "Speaker recognition—a research and technology forecast", in Proceedings Odyssey 2004: The ISCA Speaker and Language Recognition Workshop, Toledo, 2004.
[15] T. Fawcett and A. Niculescu-Mizil, "PAV and the ROC Convex Hull", Machine Learning, Volume 68, Issue 1, July 2007, pp. 97–106.
[16] T. Gneiting and A.E. Raftery, "Strictly Proper Scoring Rules, Prediction, and Estimation", Journal of the American Statistical Association, Volume 102, Number 477, March 2007 , pp. 359–378.
[17] J. Gonzalez-Rodriguez, P. Rose, D. Ramos, D. T. Toledano and J.Ortega-Garcia, "Emulating DNA: Rigorous Quantification of Evidential Weight in Transparent and Testable Forensic Speaker Recognition", IEEE Transactions on Audio, Speech and Language Processing, Vol. 15, no.7, September 2007, pp. 2104–2115.
[18] I.J. Good, "Rational Decisions", Journal of the Royal Statistical Society, 14, 1952, pp.107–114.
[19] E.T. Jaynes, *Probability Theory: The Logic of Science*, Cambridge University Press, 2003.
[20] D.A. van Leeuwen, A.F. Martin, M.A. Przybocki and J.S. Bouten, "NIST and NFI-TNO evaluations of automatic speaker recognition", Computer Speech and Language, Volume 20, Numbers 2–3, April–July 2006, pp. 128–158.
[21] D.A. van Leeuwen and N. Brümmer, "An Introduction to Application-Independent Evaluation of Speaker Recognition Systems", in Christian Müller (Ed.): *Speaker Classification I: Fundamentals, Features, and Methods. Lecture Notes in Computer Science 4343*, Springer 2007, pp.330–353.
[22] A.F. Martin and A.N. Le, "The Current State of Language Recognition: NIST 2005 Evaluation Results", in Proceedings of IEEE Odyssey 2006: The Speaker and Language Recognition Workshop, June 2006.
[23] A.F. Martin and A.N. Le, "NIST 2007 Language Recognition Evaluation", to appear Proceedings of Odyssey 2008: The Speaker and Language Recognition Workshop, January 2008.
[24] A. Niculescu-Mizil and R. Caruana, "Predicting Good Probabilities With Supervised Learning", in Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 2005.
[25] P.M. Pardalos and G. Xue, "Algorithms for a class of isotonic regression problems", Algorithmica, 23, 1999, pp.211–222.
[26] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods", in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans, eds., MIT Press, 1999, pp.61–74.
[27] M.A. Przybocki and A.N. Le, "NIST Speaker Recognition Evaluation Chronicles—Part 2", in Proceedings of IEEE Odyssey 2006: The Speaker and Language Recognition Workshop, June 2006.
[28] V.A. Ubhaya, "An O(n) algorithm for least squares quasi-convex approximation", Computers & Mathematics with Applications, Volume 14, Issue 8, 1987, pp.583–590.
[29] A. Wald, *Statistical Decision Functions*. Wiley, New York, 1950.

[30] W.J. Wilbur, L. Yeganova and Won Kim, "The Synergy Between PAV and AdaBoost", Machine Learning, Volume 61, Issue 1–3, November 2005, pp.71–103.

[31] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates", In: Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining (KDD02), 2002.