1. Introduction


Somehow, it appears that approaches of all these authors share a common feature. Namely, it is always assumed that random variables for which one desires to compute estimates satisfying ordering constraints are independent. This author is not aware of any estimation procedure that accounts for possible correlation between such variables. Hence two possible choices are: constructing a procedure dedicated to correlated data or investigating the properties of existing estimation strategies when independence assumption is dropped. In this paper the original Pool-Adjacent-Violators algorithm (PAVA) of Ayer et al (1955) is revisited and its properties in the case of non-zero correlation are assessed in a simulation study.

2. Pool-Adjacent-Violators algorithm

Let \( \pi_1, \pi_2, \ldots, \pi_n \) be unknown probabilities satisfying a simple order:

\[
\pi_1 \leq \pi_2 \leq \ldots \leq \pi_n
\]
Let $N_i$ independent trials be made of an event with probability $\pi_i$ for $i = 1,\ldots,n$. Let $y_i$ denote the number of successes in the $i$-th trial and let $p_i^* = y_i / N_i$ for $i = 1,\ldots,n$. The PAVA procedure computes estimates $p_1, \ldots, p_n$ of $\pi_1, \ldots, \pi_n$ satisfying (1) by iteratively grouping (merging) initial estimates $p_1^*, \ldots, p_n^*$ into blocks and averaging them within each block. The procedure works through repeating following steps (see Härdle (1992), Ayer et al (1955) and de Leeuw et al (2009)).

1) Assign each component $\pi_i$ for $i = 1,\ldots,n$ to a separate group so initially $n$ groups $G_1^{(0)}, \ldots, G_n^{(0)}$ exist. Set initial estimate of mean probability in each $i$-th group to $\tilde{p}_g^{(0)} = p_g^*$ for $g = 1,\ldots,n$.

2) While there exist some groups in the $k$-th step of algorithm such that associated estimates of mean probability violate the ordering constraint, find maximum-length sequence of such groups (say $G_g^{(k)}, G_{g+1}^{(k)}, \ldots, G_{h}^{(k)}$) and merge them into a single group $G_g^{(k+1)} = G_g^{(k)} \cup G_{g+1}^{(k)} \cup \ldots \cup G_h^{(k)}$ while $G_j^{(k+1)} = G_j^{(k)}$ for $j > g$ and assign a mean probability estimate $\tilde{p}_g^{(k+1)} = \left( \sum_{i \in G_g^{(k+1)}} y_i \right) / \sum_{i \in G_g^{(k+1)}} N_i$ to the group $G_g^{(k+1)}$ while $\tilde{p}_j^{(k+1)} = \tilde{p}_{j+h-g}^{(k)}$ for $j \geq g$.

3) When iteration stops after the last – say $K$-th – step (where $K \in \{0,1,\ldots\}$), with $H$ groups remaining assign a mean probability estimate computed for a group to each of its member components so that the final estimate for the component $\pi_i$ is $p_i = \tilde{p}_g^{(k)}$ for $i \in G_g$, $g = 1,2,\ldots,H$.

If $y_1,\ldots,y_n$ are independent, this procedure leads to a vector of restricted maximum likelihood estimates for probabilities $\pi_1, \pi_2, \ldots, \pi_n$. We will now abandon the independence assumption and allow for some correlation among $y_i$’s.

### 3. A simple correlation model

To investigate properties of PAVA estimator in the case when variables are correlated we will assume a simple model stating that correlation coefficient between individual binary variables is the same for all pairs of subsequent variables: $(y_1, y_2)$, $(y_2, y_3)$, $(y_3, y_4)$, $(y_{k-1}, y_k)$. Hence a procedure generating binary random vectors in the form $y = [y_1,\ldots,y_k]$ satisfying $E(y) = m$ and $\text{Cov}(y_i, y_{i+1}) / \sqrt{\text{Var}(y_i)\text{Var}(y_{i+1})} = r$ for $i = 2,\ldots,k$ and some arbitrarily chosen
m ∈ (0, 1)^k, r ∈ (0, 1) is needed. Let a vector \( U = [U_1, \ldots, U_k]' \) consist of independent components: \( U_i \sim \text{Unif}(0, 1) \) and denote:

\[
p_{11} = r \left( m_i m_{i-1} \left( 1 - m_i \right) \left( 1 - m_{i-1} \right) \right)^{0.5} + m_i m_{i-1} \quad (2)
\]

\[
p_{01} = m_i - p_{11} \quad (3)
\]

The first component of \( y \) may be generated as \( y_1 = J_i(m_i) \) and subsequent components are obtained according to the formula:

\[
y_i = \begin{cases} 
J_i \left( \frac{p_{11}}{m_{i-1}} \right) & \text{for } y_{i-1} = 1 \\
J_i \left( \frac{p_{01}}{1 - m_{i-1}} \right) & \text{for } y_{i-1} = 0
\end{cases} \quad (4)
\]

where

\[
J_i(a) = \begin{cases} 1 & \text{for } U_i < a \\
0 & \text{for } U_i \geq a
\end{cases} \quad (5)
\]

for \( i = 1, \ldots, k \) so that \( E(J_i(a)) = a \).

Such a simple procedure yields a vector \( y \) satisfying desired constraints since:

\[
E(y_i) = E(y_i \mid y_{i-1} = 1) \Pr(y_{i-1} = 1) + E(y_i \mid y_{i-1} = 0) \Pr(y_{i-1} = 0) = \\
= E \left( J_i \left( \frac{p_{11}}{m_{i-1}} \right) \right) m_{i-1} + E \left( J_i \left( \frac{p_{01}}{1 - m_{i-1}} \right) \right) (1 - m_{i-1}) = \\
= \frac{p_{11}}{m_{i-1}} \cdot m_{i-1} + \frac{p_{01}}{1 - m_{i-1}} \cdot (1 - m_{i-1}) = p_{11} - p_{01} = p_{11} - (m_i - p_{11}) = m_i
\]

and

\[
\text{Cov}(y_{i-1}, y_i) = E(y_{i-1}y_i) - E(y_{i-1})E(y_i) = P(y_{i-1} = 1, y_i = 1) - m_{i-1}m_i = \\
= P(y_{i-1} = 1 \mid y_{i-1} = 1)P(y_{i-1} = 1) - m_{i-1}m_i = \frac{p_{11}}{m_{i-1}} - m_{i-1}m_i = \\
= p_{11} - m_{i-1}m_i = r \left( m_im_{i-1} \left( 1 - m_i \right) \left( 1 - m_{i-1} \right) \right)^{0.5} = rV^{0.5}(y_{i-1})V^{0.5}(y_i)
\]

The procedure depends on the ability to generate pseudo-random numbers \( U_1, \ldots, U_k \) imitating independent random variables having uniform distribution on \( (0, 1) \). Many such generators are widely available including the fast implementation of Mersenne-Twister algorithm by Matsumoto and Nishimura (1998) implemented in the R package. This generator will be used in
our study. Some sample output of the proposed procedure will now be presented. For \( m = [1/40, 2/40, 3/40, ..., 1] \) and \( r = 0 \) we got a typical realization of a binary random vector:

\[
y_1 = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ ]
\]

For \( m = [0.5, ..., 0.5] \) and \( r = 0 \) we got a typical realization:

\[
y_2 = [1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ ]
\]

For \( m = [0.5, ..., 0.5] \) and \( r = 0.8 \) we got a typical realization:

\[
y_3 = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ ]
\]

For \( m = [0.5, ..., 0.5] \) and \( r = -0.8 \) we got a typical realization:

\[
y_4 = [0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 ]
\]

4. Simulation results

A simulation study was carried out in order to assess how the bias and mean square error of PAVA estimates for ordered probabilities depend on the sample size when variables are correlated. Three simulation experiments were carried out. In each experiment the sequence of \( n = 20, 40, ..., 200 \) binary vectors was generated independently \( h = 30000 \) times using the procedure described in previous section. All the experiments were carried out using scripts in R (R Development Core Team (2011)). PAVA estimates were computed using the ‘gmpava’ function implemented in the R ‘isotone’ package (see de Leeuw et al (2009) for a description). In the first experiment marginal probabilities were set to:

\[
m_1 = [0.48, 0.49, 0.5, 0.51, 0.52]
\]

with \( r = 0.0, 0.2, 0.5, 0.8 \). In the second experiment they were set to

\[
m_2 = [0.33, 0.33, 0.35, 0.37, 0.37]
\]

with \( r = 0.0, 0.2, 0.5, 0.8 \). In the third experiment marginal probabilities amounted to:

\[
m_3 = m_1 = [0.48, 0.49, 0.5, 0.51, 0.52]
\]
with \( r = 0.0, -0.2, -0.5, -0.8 \). Marginal probabilities were chosen close to each other in order to make the effects of correcting breached constraints by PAVA clearly visible. The bias and mean square error observed in the first experiment are shown in figures 1 and 2. The bias and mean square error observed in the second experiment are shown in figures 3 and 4. The bias and mean square error observed in the third experiment are shown in figures 5 and 6.

![Fig. 1. The bias of PAVA estimates for \( m = m_1 \) and \( r = 0.0, 0.2, 0.5, 0.8 \)](image)

In all three experiments the scope of observed bias depends on the position of a variable in the simple order (1). The bias for estimates \( p_4 \) and \( p_5 \) of rightmost probabilities \( \pi_4 \) and \( \pi_5 \) tends to be positive while for leftmost probabilities \( \pi_1 \) and \( \pi_2 \) the bias of estimators \( p_1 \) and \( p_2 \) tends to be negative.
Meanwhile, estimator $p_3$ of the innermost probability $\pi_3$ seems to be approximately unbiased. The introduction of strong positive correlation in the first and second experiment seems to reduce the bias to some extent. The effect of negative correlation assessed in the third experiment is more complex: estimates of outermost variables $\pi_1$ and $\pi_5$ seem to be unaffected while the bias of estimates for $\pi_2$ and $\pi_4$ is slightly reduced. Anyway, in all experiments and for all parameters $\pi_1,...,\pi_5$ the bias of estimates apparently tends to zero when sample size $n$ increases. Hence there is no evidence that the asymptotic unbiasedness of PAVA estimates which was proven by Ayer et al (1955) under assumption of independence ceases to hold in the presence of correlation.

Fig. 2. The MSE of PAVA estimates for $m = m_1$ and $r = 0.0, 0.2, 0.5, 0.8$
The mean square error of estimates depends on a position of a variable in the order (1) as well. In all three experiments the MSE for estimators $p_1$ and $p_5$ is clearly the highest of all the five and only in the second experiment the observed difference between these two is more pronounced (with $p_1$ being somehow more accurate than $p_5$). Anyway, in all experiments the MSE of all estimators $p_1, \ldots, p_5$ apparently tends to zero with growing sample size which suggests that consistency is retained under correlation.
Fig. 4. The MSE of PAVA estimates for \( m = m_2 \) and \( r = 0.0, 0.2, 0.5, 0.8 \).
Fig. 5. The bias of PAVA estimates for $m = m_3$ and $r = 0.0, -0.2, -0.5, -0.8$
Fig. 6. The MSE of PAVA estimates for $m = m_3$ and $r = 0.0, -0.2, -0.5, -0.8$

5. Conclusion

Simulation experiments carried out during this study covered several multivariate distributions of a binary vector $y$, involving dependencies between its individual components. Even for very strong correlations, no evidence of any departures from the consistency property was found. Hence, presented results suggest that PAVA estimates may retain consistency in the situation when binary variables are correlated.

Obviously, those promising simulation results do not constitute a formal proof of consistency as they cover only a few of infinitely many possible combinations of parameters. However they justify theoretical efforts aimed at establishing properties of PAVA-based estimates under correlation. Such efforts may significantly widen the range of possible applications for the PAVA procedure.
References


---

O WŁASNOŚCIACH ALGORYTMU PAVA
DLA ZMIENNYCH ZALEŻNYCH

Streszczenie

Algorytm PAVA (od ang. Pool-Adjacent-Violators Algorithm) jest popularnym narzędziem estymacji wykorzystywanym do szacowania wartości oczekiwanych ciągu zmiennych losowych w sytuacji, gdy dostępna informacja dodatkowa pozwala stwierdzić, że między tymi wartościami oczekiwanymi zachodzi relacja porządku. Uzyskane za pomocą tego algorytmu oszacowania maksymalizują (warunkowo) funkcję wiarygodności przy założeniu, że relacja ta

---

ŁASNOŚCIACH ALGORYTMU PAVA
DLA ZMIENNYCH ZALEŻNYCH

Streszczenie

Algorytm PAVA (od ang. Pool-Adjacent-Violators Algorithm) jest popułarnym narzędziem estymacji wykorzystywanym do szacowania wartości oczekiwanych ciągu zmiennych losowych w sytuacji, gdy dostępna informacja dodatkowa pozwala stwierdzić, że między tymi wartościami oczekiwanymi zachodzi relacja porządku. Uzyskane za pomocą tego algorytmu oszacowania maksymalizują (warunkowo) funkcję wiarygodności przy założeniu, że relacja ta
jest spełniona oraz poszczególne zmienne są niezależne. Wydaje się, że żadna z przedstawionych w literaturze przedmiotu modyfikacji tej procedury estymacji nie uwzględnia możliwości wystąpienia zależności pomiędzy poszczególnym zmiennymi. W niniejszym artykule przedstawiono rezultaty eksperymentów symulacyjnych których celem było zbadanie własności oszacowań uzyskanych za pomocą tej procedury gdy zmienne są skorelowane.