Pool Adjacent Violator Algorithm (PAVA) on negative log loss

Zihao Zheng

Version: February 18, 2022

1 Problem description

Suppose the $\boldsymbol{y} : y_1, y_2, \cdots, y_n$ are observed frequencies with $y_i \in [0, 1]$ and $\sum_{i=1}^{n} y_i = 1$. Consider the following optimization problem:

$$\min_{\boldsymbol{p} \in \mathbf{R}^n} \quad -\sum_{i=1}^{n} y_i \log p_i \tag{1}$$

$$\text{s.t.} \quad p_i \leq p_{i+1} \; 1 \leq i \leq n-1 \tag{2}$$

$$\sum_{i=1}^{n} p_i = 1 \tag{3}$$

This is similar to the isotonic regression problem where the objective function of Equation 1 is replaced by the normal squared error loss $\sum_{i=1}^{n}(y_i - p_i)^2$ and without the regularization constraint in Equation 3[1].

The pool adjacent violator algorithm (PAVA) solves the isotonic regression with squared error loss efficiently, within finite number of iterations. At each iteration, it examine the violator, pool the observation into the same block, and take the average within each block. The average solver within the

[1]This regularization constraint is necessary for negative log loss but not necessary for the mean square loss, even though the solution from PAVA would guarantee this condition hold.

pooled block could be also generalized to other internal solvers [De Leeuw et al., 2010]. This algorithm terminates in finite step until there is no further violators. I am trying to argue the same algorithm, taking the average inside each pooled block, solves the optimization problem with negative log loss.

2 Karush–Kuhn–Tucker condition for the problem

Define dual variable $\lambda_i, 1 \leq i \leq n-1$ are dual variables associated those $n-1$ inequality constraints as in Equation 2 and μ is the dual variable associated with the equality constraint as in Equation 3. Then the Lagrangian equation is:

$$L(\boldsymbol{p}, \boldsymbol{\lambda}, \mu) = -\sum_{i=1}^{n} y_i \log p_i - \sum_{i=1}^{n-1} \lambda_i (p_{i+1} - p_i) - \mu(1 - \sum_{i=1}^{n} p_i) \quad (4)$$

And the following four conditions are so-called Karush–Kuhn–Tucker (KKT) conditions.

1. 0 partial derivatives:

$$\partial_{\boldsymbol{p}} \nabla L(\boldsymbol{p}, \boldsymbol{\lambda}, \mu) = 0 \quad (5)$$

2. Primal feasibility (a):

$$p_i \leq p_{i+1}, \ , 1 \leq i \leq n-1 \quad (6)$$

3. Primal feasibility (b):

$$\sum_{i=1}^{n} p_i = 1 \tag{7}$$

4. Dual feasibility:

$$\boldsymbol{\lambda} \geq 0 \tag{8}$$

5. Complementary slackness:

$$\lambda_i(p_{i+1} - p_i) = 0, \ , 1 \leq i \leq n-1 \tag{9}$$

In the following section, I am going to prove the solution from the PAVA algorithm satisfies the KKT condition 1-5.

3 Sketch of the prove

Following from Dijkstra et al. [1976] and Feijen and Van Gasteren [2013], there are two components to prove the computational algorithm, in the iterative based, is correct.

- Loop invariant and
- Termination of the algorithm

The loop invariant checks the KKT condition 1 (0 partial derivatives), 4 (dual feasibility) and 5 (complementary slackness) and the termination of

the algorithm checks the KKT condition 2 and 3 (primal feasibility). More specifically, we need to check the termination of the algorithm satisfies the primal feasibility and the other KKT conditions (1, 4, and 5) are satisfied in the initialization step and during the PAV algorithm.

The termination of the algorithm is straightforward. It is easy to see at the end of the PAV algorithm, we will get answer satisfying $p_i \leq p_{i+1}, 1 \leq i \leq n-1$ and $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} y_i = 1$. The rest of the work is to check:

- The initialization step establishes the loop invariant
- The PAV step (pool violator blocks and take the average) maintains the loop invariant

4 Prove of the problem

I will prove the initialization step established the loop invariant (i.e., satisfying KKT condition 1, 4 and 5) and the PAV step step also maintains the loop invariant.

4.1 Initialization step

I claim take $p_i = y_i, \lambda_i = 0$ and $\mu = 1$ establishes the KKT condition 1, 4 and 5.

It is easy to verity condition 4 and 5 by simply taking $\lambda_i = 0$. For condition 1, each entry of the partial derivative of the Lagrangian function is (for

notation simplicity, denote $\lambda_0 = \lambda_n = 0$):

$$\partial_i \nabla L(\boldsymbol{p}, \boldsymbol{\lambda}, \mu) = -\frac{y_i}{p_i} + \lambda_i - \lambda_{i-1} + \mu = 0$$

4.2 PAV step

Here I claim the pool adjacent violator (and take average within block) step maintains the KKT condition 1, 4 and 5. We prove $\mu = 1$ will help us find a solution satisfying the KKT condition.

During the algorithm, at any vector $\boldsymbol{p}$, we divide it up into blocks of equal consecutive components. Such blocks may have length one if some p_j is not equal to the values on either side. So we consider a block:

$$p_{j^*-1} \neq p_{j^*} = \cdots = p_{j^{**}-1} \neq p_{j^{**}}$$

By complementary slackness (KKT condition 5), we have $\lambda_{j^*-1} = \lambda_{j^{**}-1} = 0$. And by 0 derivatives of Lagrangian function (KKT condition 1) and taking $\mu = 1$, we have:

$$\begin{aligned} 0 &= \sum_{j=j^*}^{j^{**}-1} \left(-\frac{y_j}{p_j} + \lambda_j - \lambda_{j-1} + 1 \right) \\ &= \sum_{j=j^*}^{j^{**}-1} (1 - \frac{y_j}{p_j}) \end{aligned}$$

If denote $\neq p_{j^*} = \cdots = p_{j^{**}-1} = p$, then $p = \frac{1}{j^{**}-j^*} \sum_{j=j^*}^{j^{**}-1} y_j$, which is the average of observed frequencies in the block. This calculation verified taking

the average within block would maintain the loop invariant of KKT condition 1 and 5. The remaining part is to verify the dual feasibility (KKT condition 4, $\boldsymbol{\lambda} \geq 0$).

To prove the dual feasibility, we need revisit the block. For any $j^* \leq r < j^{**}$, we have

$$0 \;=\; \sum_{j=j^*}^{r} \left(-\frac{y_j}{p_j} + \lambda_j - \lambda_{j-1} + 1 \right)$$

Therefore we have

$$\lambda_r = \sum_{j=j^*}^{r} \left(\frac{y_j}{p_j} + 1 \right)$$

We would like to check the non-negativity of this running sum to prove dual feasibility.

Recall that we have to do the pooling step because of observing the violation of isotonic order. In other words, we would like to argue if running sums are nonnegative for all blocks before pooling, they remain to be nonnegative for the new pooled block that is created in the PAV step of the algorithm.

Let's look at the adjacent violator blocks. Suppose for some j^*, j^{**}, j^{***}, we have:

$$p_{j^*-1} \neq p_{j^*} = \cdots = p_{j^{**}-1} > p_{j^{**}} = \cdots = p_{j^{***}-1} \neq p_{j^{***}}$$

We also know the mean values for each block are the averages of the observed

frequencies for each block. When we pool the mean values for the block starting at j^* will decrease and the mean values for the other block will increase. Therefore the λ for the block starting at j^* will increase, and the λ for the other block will decrease. So we only have to check what happens in the other block.

Let's denote the average of the left block is $p^* = \frac{1}{j^{**}-j^*}\sum_{j=j^*}^{j^{**}-1} y_j$ and the average of the right block is $p^{**} = \frac{1}{j^{***}-j^{**}}\sum_{j=j^{**}}^{j^{***}-1} y_j$ and the pooled new average is $p^{\text{new}} = \frac{1}{j^{***}-j^*}\sum_{j=j^*}^{j^{***}-1} y_j$.

The Lagrange multiplier $\lambda_r = \sum_{j=j^*}^{r}\left(\frac{y_j}{p^{\text{new}}}+1\right)$ for $j^* \le r < j^{***}$. The non-negativity of λ_r is equivalent of showing for $j^{**} \le r < j^{***}$:

$$\begin{aligned} p^{\text{new}} &\le \frac{1}{r-j^*+1}\sum_{j=j^*}^{r} y_j \\ &= \frac{1}{r-j^*+1}\left(p^*(j^{**}-j^*)+p^{**}(r-j^{**}+1)\right) \end{aligned}$$

This is then, by definition of p^{new}, equivalent to

$$\frac{(j^{**}-j^*)p^* + (j^{***}-j^{**})p^{**}}{j^{***}-j^*} \le \frac{(j^{**}-j^*)p^* + (r-j^{**}+1)p^{**}}{r-j^*+1}$$

This is equivalent to check

$$(j^{**}-j^*)(j^{***}-r+1)(p^*-p^{**}) \ge 0$$

This is simply true by definition of r and observing $p^* > p^{**}$.

References

J. De Leeuw, K. Hornik, and P. Mair. Isotone optimization in r: pool-adjacent-violators algorithm (pava) and active set methods. *Journal of statistical software*, 32:1–24, 2010.

E. W. Dijkstra, E. W. Dijkstra, E. W. Dijkstra, and E. W. Dijkstra. *A discipline of programming*, volume 613924118. prentice-hall Englewood Cliffs, 1976.

W. H. Feijen and A. J. Van Gasteren. *On a method of multiprogramming*. Springer Science & Business Media, 2013.