Big Data and
Information Analytics

BDIA, 3(2): 54–67.
DOI: 10.3934/bdia.2018007
Received: 03 December 2018
Accepted: 18 March 2019
Published: 22 March 2019

http://www.aimspress.com/journal/BDIA

Research article

Resolutions to flip-over credit risk and beyond-least squares estimates and maximum likelihood estimates with monotonic constraints

Bill Huajian Yang*

Royal Bank of Canada, 155 Wellington St W, Toronto, ON M5V 3H6, Canada

* **Correspondence:** Email: h_y02@yahoo.ca.

Abstract: Given a risk outcome y over a rating system $\{R_i\}_{i=1}^{k}$ for a portfolio, we show in this paper that the maximum likelihood estimates with monotonic constraints, when y is binary (the Bernoulli likelihood) or takes values in the interval $0 \leq y \leq 1$ (the quasi-Bernoulli likelihood), are each given by the average of the observed outcomes for some consecutive rating indexes. These estimates are in average equal to the sample average risk over the portfolio and coincide with the estimates by least squares with the same monotonic constraints. These results are the exact solution of the corresponding constrained optimization. A non-parametric algorithm for the exact solution is proposed. For the least squares estimates, this algorithm is compared with "pool adjacent violators" algorithm for isotonic regression. The proposed approaches provide a resolution to flip-over credit risk and a tool to determine the fair risk scales over a rating system.

Keywords: risk scale; maximum likelihood; least squares; isotonic regression; flip-over credit risk

1. Introduction

Flip-over is a phenomenon where a low risk segment has a larger value of risk estimate than a high risk segment. It is usually caused by over-segmentation when practitioners seek discriminatory power greedily in the model development stage. This means that a segment is forced to split further into several small segments for a seemly in-sample increase of the discriminatory power, but they have no obvious difference from the population perspectives. When flip-over occurs, practitioners typically combine segments manually, or through hierarchical clustering.

We show in this paper that the flip-over phenomenon can be resolved by approaches based on least squares estimates or maximum likelihood estimates with monotonic constraints.

Let $\{R_i\}_{i=1}^{k}$ denote a segmentation or the non-default risk ratings for a risk-rated portfolio. Let y, $-\infty < y < +\infty$, be a general risk outcome, for example, the loan loss, the exposure at default, or the default indicator. A monotonicity rule is assumed: a higher index R_i is expected to carry higher risk, i.e., the expected value of y is higher for a higher index rating.

Monotonic constraints are widely used in learning processes. Examples of learnings, where monotonic constraints are imposed, include isotonic regression [2,3,5,8] risk scale estimation for a rating system [18] classification tree [11] rule learning [6] binning [1,4] and deep lattice network [19].

We use the following notations: For a given a sample S, let y_{ij} denote the j^{th} observation of the risk outcome over R_i and n_i the total number of observations for R_i. We assume $n_i > 0$. Let $d_i = \sum_{j=1}^{n_i} y_{ij}$ be the sum of all the observed y-values, and $r_i = d_i/n_i$, the average observed risk for R_i.

We are interested in the least squares estimates $\{p_i\}_{i=1}^{k}$that minimize the sum squared error (1.1) subject to monotonic constraints (1.2) below:

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(y_{ij} - p_i\right)^2, \tag{1.1}$$

$$p_1 \le p_2 \le \cdots \le p_k. \tag{1.2}$$

When y is binary (e.g. the default indicator) or takes values in the interval $0 \le y \le 1$, we are interested in the maximum likelihood estimates $\{p_i\}_{i=1}^{k}$ that maximize the log-likelihood (1.3) below subject to (1.2):

$$LL = \sum_{i=1}^{k} [d_i \log(p_i) + (n_i - d_i) \log(1 - p_i)], \tag{1.3}$$

where the additive term $d_i \log(p_i) + (n_i - d_i)\log(1 - p_i)$ corresponds to the Bernoulli log-likelihood when y is binary, i.e., we assume that the risk outcome y over rating R_i follows a Bernoulli distribution with probability p_i. It corresponds to the quasi-Bernoulli log-likelihood when y takes values in the interval $0 \le y \le 1$ [10].

Main results. In this paper, we show that (see Propositions 3.1 and 4.1), for a given sample $S = \{y_{ij} \mid 1 \le i \le k, 1 \le j \le n_i\}$, there exist partition integers $\{k_i\}_{i=0}^{m}$, where $0 = k_0 < k_1 < \cdots < k_m = k$, such that the values $\{p_j\}_{j=1}^{k}$, given by (1.4) below, minimize (1.1) and maximize (1.3), subject to (1.2):

$$p_j = \frac{d_{k_{i-1}+1} + d_{k_{i-1}+2} + \cdots + d_{k_i}}{n_{k_{i-1}+1} + n_{k_{i-1}+2} + \cdots + n_{k_i}}, \; k_{i-1} + 1 \le j \le k_i, \tag{1.4}$$

These $\{p_j\}_{j=1}^{k}$ satisfy the equation below:

$$\frac{n_1p_1+n_2p_2+\cdots+n_kp_k}{n}=\frac{d}{n} \tag{1.5}$$

where

$$n=n_1+n_2+\cdots+n_k, \tag{1.6}$$
$$d=d_1+d_2+\cdots+d_k. \tag{1.7}$$

These results are the exact solution for the corresponding constrained optimization and are proved in a more general setting under weighted least squares and weighted maximum likelihood.

Given the above results, flip-over credit risk can be resolved by combining each group with indexes in $k_{i-1}+1\le j\le k_i$, and replacing their estimates by the average of the risk over the group.

One of the most important estimations with monotonic constraints is the isotonic regression [2] Given values $\{r_j\}_{j=1}^{k}$, the goal of isotonic regression is to find $\{p_i\}_{i=1}^{k}$, subject to (1.2), that minimize the weighted sum squares $\sum_{i=1}^{k}w_i(r_i-p_i)^2$, where $\{w_i\}_{i=1}^{k}$ are the given weights. A unique exact solution to the isotonic regression problem exists and can be obtained by a non-parametric algorithm called Pool Adjacent Violators (PAV) [2,3,5,8].

A non-parametric algorithm (Algorithm 5.1) with time complexity $O(k^2)$ is proposed in Section. 5 for finding these partition integers in (1.4), hence the estimates. For estimates with general monotonic constraints, we propose a parametric algorithm (Algorithm 5.2) for least squares estimates with constraints: $p_i\le p_{i+1}+\epsilon_i$ for $1\le i\le k$ and $\epsilon_i\ge 0$, and for maximum likelihood estimates with constraints: $p_{i+1}/p_i\ge 1+\epsilon$ for $1\le i\le k$ and $\epsilon\ge 0$. A detailed comparison between the PAV algorithm and the non-parametric algorithm proposed in this paper can be found in Section. 6.1.

The key ideas to the proof of (1.4) and the algorithms proposed in this paper are the re-parameterization of the estimates so that (1.2) is automatically satisfied. Consequently, the constrained programming is transformed into a tractable non-constrained mathematical programming problem (see Section. 3, 4 and 5).

The paper is organized as follows: In Section. 2, we define the partition integer for a given sample. A formula like (1.4) is shown in Section. 3 for weighted maximum likelihood estimates and in Section. 4 for weighted least squares estimates. The non-parametric algorithm for the exact solution is proposed in Section. 5. In Section. 6, we illustrate how this proposed non-parametric algorithm can be used to determine the fair risk scales over a rating system. Applications to risk-supervised monotonic binning are also discussed.

2. The partition integers

For a given sample $S=\{y_{ij}\mid 1\le i\le k, 1\le j\le n_i\}$, let $\{w_i\}_{i=1}^{k}$ denote the given weights, where $\mathrm{w_i}>0$ is the weight assigned to the observed outcomes $\{y_{ij}\}_{j=1}^{n_i}$ for R_i. We use the notations introduced in Section. 1 and let $r_i=d_i/n_i$ and $d_i=\sum_{j=1}^{n_i}y_{ij}$.

For $1\le i\le j\le k$, let

$$u(i,j) = \frac{r_i n_i w_i + r_{i+1} n_{i+1} w_{i+1} + \cdots + r_j n_j w_j}{n_i w_i + n_{i+1} w_{i+1} + \cdots + n_j w_j} \tag{2.1}$$

$$= \frac{d_i w_i + d_{i+1} w_{i+1} + \cdots + d_j w_j}{n_i w_i + n_{i+1} w_{i+1} + \cdots + n_j w_j}. \tag{2.2}$$

By (2.1), $u(i,j)$ is the weighted average of $\{r_i, r_{i+1}, \ldots, r_j\}$ where r_h is weighted by $n_h w_h$. Specifically, we have

$$u(1,k) = \frac{d_1 w_1 + d_2 w_2 + \cdots + d_k w_k}{n_1 w_1 + n_2 w_2 + \cdots + n_k w_k} = \frac{D}{N}, \tag{2.3}$$

the weighted average of $\{r_i\}_{i=1}^{k}$ over the portfolio, where N and D are defined respectively by (2.4) and (2.5) below:

$$N = n_1 w_1 + n_2 w_2 + \cdots + n_k w_k, \tag{2.4}$$

$$D = d_1 w_1 + d_2 w_2 + \cdots + d_k w_k. \tag{2.5}$$

Let $\{k_i\}_{i=0}^{m}$ be the partition integers, where $0 = k_0 < k_1 < \cdots < k_m = k$, such that (2.6) and (2.7) below hold for $0 < i \le m$:

$$u(k_{i-1}+1, k_i) = min\{u(k_{i-1}+1, j) \mid k_{i-1}+1 \le j \le k\}, \tag{2.6}$$

$$u(k_{i-1}+1, k_i) < u(k_{i-1}+1, k_i+1). \tag{2.7}$$

That is, given k_{i-1}, the integer k_i is the largest index such that $u(k_{i-1}+1, j)$ reaches its minimum at $j = k_i$ within all remaining indexes $j \ge k_{i-1}+1$. When $\{r_i\}_{i=1}^{k}$ are strictly increasing, we have $m = k$ and $\{k_i\}_{i=1}^{m} = \{1, 2, \ldots, k\}$.

By (2.6) and (2.7), we have the following inequalities:

$$u(1, k_1) < u(k_1+1, k_2) < \cdots < u(k_{m-1}+1, k_m). \tag{2.8}$$

This is because if, for example, $u(1,k_1) \ge u(k_1+1, k_2)$, then we have:

$$\begin{aligned} u(1,k_2) &= \frac{n_1 w_1 + n_2 w_2 + \cdots + n_{k_1} w_{k_1}}{n_1 w_1 + n_2 w_2 + \cdots + n_{k_2} w_{k_2}} u(1,k_1) + \frac{n_{k_1+1} w_{k_1+1} + n_{k_1+2} w_{k_1+2} + \cdots + n_{k_2} w_{k_2}}{n_1 w_1 + n_2 w_2 + \cdots + n_{k_2} w_{k_2}} u(k_1+1, k_2) \\ &\le \frac{n_1 w_1 + n_2 w_2 + \cdots + n_{k_1} w_{k_1}}{n_1 w_1 + n_2 w_2 + \cdots + n_{k_2} w_{k_2}} u(1,k_1) + \frac{n_{k_1+1} w_{k_1+1} + n_{k_1+2} w_{k_1+2} + \cdots + n_{k_2} w_{k_2}}{n_1 w_1 + n_2 w_2 + \cdots + n_{k_2} w_{k_2}} u(1,k_1) \\ &= u(1,k_1). \end{aligned}$$

This contradicts the fact that k_1 is the largest index where $u(1,j)$ reaches its minimum at $j = k_i$ for all $j \ge k_{i-1}+1$.

3. Weighted maximum likelihood estimates with monotonic constraints

Under weighted maximum likelihood framework, log-likelihood (1.3) becomes

$$LL = \sum_{i=1}^{k} w_i [d_i \log(p_i) + (n_i - d_i)\log(1-p_i)]. \tag{3.1}$$

We are interested in the weighted maximum likelihood estimates $\{p_i\}_{i=1}^{k}$ that maximize (3.1) subject to (1.2).

Let $f_i(p_i) = d_i \log(p_i) + (n_i - d_i)\log(1 - p_i)$ be an additive term of (3.1). Values $f_i(1)$ and $f_i(0)$ are defined as follows: By taking the limit of $f_i(p_i)$ when p_i approaches 1 from the left, we can assume $f_i(1) = 0$ if $d_i = n_i$, and $f_i(1) = -\infty$ if $d_i < n_i$. Similarly, by taking the limit of $f_i(p_i)$ when p_i approaches 0 from the right, we can assume $f_i(0) = 0$ if $d_i = 0$, and $f_i(0) = -\infty$ if $d_i > 0$. In absence of (1.2), the sample means $\{r_i\}_{i=1}^{k}$ maximize (3.1), because each $f_i(p_i)$ is maximized at $p_i = r_i$. This is true when $r_i = 0, 1$. For $0 < \mathrm{r_i} < 1$, one can see it by taking the derivative for the additive term with respect to p_i and set it to zero (see [18]).

Proposition 3.1. With the partition integers $\{k_i\}_{i=0}^{m}$ defined by (2.6) and (2.7), the values $\{p_j\}_{j=1}^{k}$ given by (3.2) below maximize (3.1) subject to (1.2):

$$p_j = u(k_{i-1} + 1, k_i)$$

$$= \frac{d_{k_{i-1}+1} w_{k_{i-1}+1} + d_{k_{i-1}+2} w_{k_{i-1}+2} + \cdots + d_{k_i} w_{k_i}}{n_{k_{i-1}+1} w_{k_{i-1}+1} + n_{k_{i-1}+2} w_{k_{i-1}+2} + \cdots + n_{k_i} w_{k_i}}, \text{ where } k_{i-1} + 1 \le j \le k_i, \qquad (3.2)$$

In addition, the following equation holds:

$$\frac{n_1 w_1 p_1 + n_2 w_2 p_2 + \cdots + n_k w_k p_k}{n_1 w_1 + n_2 w_2 + \cdots + n_k w_k} = \frac{D}{N}. \qquad (3.3)$$

Proof. First, by (2.8), the estimates $\{p_i\}_{i=1}^{k}$ given specifically by (3.2) satisfy (1.2). By (3.2) and (2.2), the sum of $\{n_j w_j p_j \mid k_{i-1} + 1 \le j \le k_i\}$ is equal to the sum of $\{d w_j \mid k_{i-1} + 1 \le j \le k_i\}$. Thus, we have:

$$n_1 w_1 p_1 + n_2 w_2 p_2 + \cdots + n_k w_k p_k = d_1 w_1 + d_2 w_2 + \cdots + d_k w_k = D.$$

Therefore, with these specific values for $\{p_i\}_{i=1}^{k}$, Eq (3.3) holds.

For $i \le j$, let

$$LL = \sum_{i=1}^{k} w_i [d_i \log(p_i) + (n_i - d_i)\log(1 - p_i)] = \sum_{i=1}^{m} LL(k_{i-1} + 1, k_i)$$

where

$$LL(k_{i-1} + 1, k_i) = \sum_{h=k_{i-1}+1}^{k_i} w_h [d_h \log(p_h) + (n_h - d_h)\log(1 - p_h)].$$

Because of (2.8), it suffices to show that each log-likelihood $LL(k_{i-1} + 1, k_i)$ is maximized at $p_j = u(k_{i-1} + 1, k_i)$ for $k_{i-1} + 1 \le j \le k_i$, subject to (1.2) within the range $k_{i-1} + 1 \le j \le k_i$. We show only the case $i = 1$ where $LL(k_{i-1} + 1, k_i)$ is $LL(1, k_1)$. The proof for other cases is similar. Without loss of generality, we assume $k_1 = k$. In this case, $m = 1$, $k_1 = k$, and $LL(1, k) = LL$.

As the maximum likelihood estimates for probabilities, $0 \le p_j \le 1$ for $1 \le j \le k$. Consider the following four cases: (a) $p_k = 1$. Then the additive term $f_k(p_k)$, hence LL, takes value $-\infty$ if

$d_k < n_k$. Hence $d_k = n_k$, and $r_k = 1$. Because $u(1, j)$ reaches its minimum at $j = k$ for $1 \leq j \leq k$, we must have $r_j = 1$ for all $1 \leq j \leq k$, by (2.1). Therefore $u(1, k) = 1$ and, by (3.2), $p_j = 1$ for all $1 \leq j \leq k$. These values of $\{p_j\}_{j=1}^{k}$ do maximize LL subject to (1.2). (b) $p_1 = 0$. As for case (a), we have $p_j = 0$ for all $1 \leq j \leq k$. We must have $d_j = 0$ for all $1 \leq j \leq k$, and $u(1, k) = 0$ (thus the proposition holds). Otherwise $f_k(p_k)$, hence LL, takes value $-\infty$. (c) $u(1, k) = 1$. Then $r_j = 1$ for all $1 \leq j \leq k$. As for case (a), the proposition holds. (d) $u(1, k) = 0$. Then $r_j = 0$ for all $1 \leq j \leq k$. As for case (b), the proposition holds.

Therefore, we can assume $0 < u(1, k) < 1$, $p_k < 1$, and $p_1 > 0$. Then we can parameterize p_j for $1 \leq j \leq k$ by letting

$$p_{k+1-j} = \exp[-(b_1 + b_2 + \cdots + b_j)],\ b_j = a_j^2, \tag{3.4}$$

where $-\infty < a_j < +\infty$ for $1 \leq j \leq k$. With this parameterization, (1.2) is satisfied. By plugging (3.4) into LL, we transform the constrained optimization problem to a non-constrained mathematical programming problem. The partial derivative of LL with respect to a_j is given by:

$$\frac{\partial LL}{\partial a_j} = \sum_{i=1}^{k} \frac{\partial}{\partial a_j} w_{k+1-i}[d_{k+1-i} \log(p_{k+1-i}) + (n_{k+1-i} - d_{k+1-i}) \log(1 - p_{k+1-i})]$$

$$= \sum_{i=j}^{k} w_{k+1-i} \left[-2a_j d_{k+1-i} + \frac{2a_j(n_{k+1-i} - d_{k+1-i})p_{k+1-i}}{1 - p_{k+1-i}}\right]$$

$$= \sum_{i=j}^{k} w_{k+1-i} \left[-2a_j d_{k+1-i} - 2a_j(n_{k+1-i} - d_{k+1-i}) + \frac{2a_j(n_{k+1-i} - d_{k+1-i})}{1 - p_{k+1-i}}\right]$$

$$= 2a_j \sum_{i=j}^{k} w_{k+1-i} \left[\frac{(n_{k+1-i} - d_{k+1-i})}{1 - p_{k+1-i}} - n_{k+1-i}\right]$$

$$= 2a_j \sum_{i=1}^{k+1-j} w_i \left[\frac{(n_i - d_i)}{1 - p_i} - n_i\right] = 2a_j g(j)$$

where

$$g(j) = \sum_{i=1}^{k+1-j} w_i \left[\frac{n_i - d_i}{1 - p_i} - n_i\right]. \tag{3.5}$$

Setting this partial derivative to zero we have either $a_j = 0$ or $g(j) = 0$. For $j = 1$, we have $a_1 \neq 0$, otherwise $p_k = 1$, contrary to our assumption. Thus, we have

$$0 = g(1) = \sum_{i=1}^{k} w_i \left[\frac{n_i - d_i}{1 - p_i} - n_i\right]. \tag{3.6}$$

We claim that $a_j = 0$ for all $1 < j \leq k$. If this is true, then $p_1 = p_2 = \cdots = p_k$. Then by (3.6) we have $p_1 = \frac{D}{N} = u(1, k)$, and the proof follows. Suppose $1 = i_1 < \cdots < i_H$ (where $1 < H$ and $i_H \leq k$) are all the indexes such that $g(i_h) = 0$ and $a_{i_h} \neq 0$ for $1 \leq h \leq H$. For $1 < h < H$, we have:

$$0 = g(i_{h-1}) - g(i_h)$$
$$= \sum_{i=1}^{k+1-i_{h-1}} w_i \left[\frac{n_i - d_i}{1-p_i} - n_i\right] - \sum_{i=1}^{k+1-i_h} w_i \left[\frac{n_i - d_i}{1-p_i} - n_i\right]$$
$$= \sum_{i=k+2-i_h}^{k+1-i_{h-1}} w_i \left[\frac{n_i - d_i}{1-p_i} - n_i\right]. \tag{3.7}$$

Since $a_j = 0$ when $i_{h-1} < j < i_h$, all $\{p_i\}_{i=k+2-i_0}^{k+1-i_{h-1}}$ are equal to $p_{k+1-i_{h-1}}$. Thus (3.7) becomes

$$0 = \sum_{i=k+2-i_h}^{k+1-i_{h-1}} w_i[(n_i - d_i) - n_i(1 - p_i)]. \tag{3.8}$$

Solving (3.8) for $p_{k+1-i_{h-1}}$, we have:

$$p_{k+1-i_{h-1}} = \frac{d_{k+2-i}w_{k+2-i} + d_{k+3-i}w_{k+3-i} + \cdots + d_{k+1-j}w_{k+1-j}}{n_{k+2-i}w_{k+2-i} + n_{k+3-i}w_{k+3-i} + \cdots + n_k + j - iw_{k+j-i}}$$
$$= u(k + 2 - i, k + 1 - j). \tag{3.9}$$

where $i = i_h$ and $j = i_{h-1}$. Similarly for i_H, we have $g(i_H) = 0$, thus:

$$0 = \sum_{i=1}^{k+1-i_H} w_i \left[\frac{n_i - d_i}{1-p_i} - n_i\right]. \tag{3.10}$$

Since all $a_j = 0$ when $i_H < j \le k$, all $\{p_i\}_{i=1}^{k+1-i_H}$ are equal to p_1. By (3.10) we have $p_1 = u(1, k + 1 - i_H)$. Consequently, each of $\{p_i\}_{i=1}^{k}$ is either $u(1, k + 1 - i_H)$ or is given by one of $\{u(k + 2 - i_h, k + 1 - i_{h-1})\}$. Thus Eq (3.3) holds, because by (2.2) the sum of $\{n_j w_j p_j \mid k + 2 - i_h \le j \le k + 1 - i_{h-1}\}$ is equal to the sum of $\{dw_j \mid k + 2 - i_h \le j \le k + 1 - i_{h-1}\}$, and the sum of $\{n_j w_j p_j \mid 1 \le j \le k + 1 - i_H\}$ is equal to the sum of $\{dw_j \mid 1 \le j \le k + 1 - i_H\}$.

Now that the weighted sum in the right-hand-side of (3.3) must be larger than p_1, because $p_i > p_1$ for all $i \ge k + 2 - i_H$. However, the left-hand-side of (3.3) is $u(1, k)$, therefore we have:

$$u(1, k + 1 - i_H) = p_1 < u(1, k).$$

This contradicts to the assumption that $i = k$ is the largest index within $1 \le i \le k$ such that $u(1, i)$ reaches the minimum.

4. Weighted least squares estimates with monotonic constraints

We use the notations introduced in Section. 1. Under weighted least squares framework, (1.1) changes to (4.1) below:

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n_i} w_i (y_{ij} - p_i)^2, \tag{4.1}$$

$$SSE = \sum_{i=1}^{k}\sum_{j=1}^{n_i} w_i\left(y_{ij} - r_i\right)^2 + \sum_{i=1}^{k} n_i w_i (r_i - p_i)^2 = SSE_1 + SSE_2$$

where $SSE_1 = \sum_{i=1}^{k}\sum_{j=1}^{n_i} w_i\left(y_{ij} - r_i\right)^2$ and

$$SSE_2 = \sum_{i=1}^{k} n_i w_i (r_i - p_i)^2. \qquad (4.2)$$

Since SSE_1 is a constant term, the weighted least squares estimates are the estimates $\{p_i\}_{i=1}^{k}$ that minimize (4.2) subject to (1.2). Note that, in absence of (1.2), $\{r_i\}_{i=1}^{k}$ minimize (4.1).

Proposition 4.1. With the partition integers $\{k_i\}_{i=0}^{m}$ defined by (2.6) and (2.7), the values $\{p_i\}_{i=1}^{k}$ given as in Proposition 3.1 by (4.3) below, minimize (4.2) subject to (1.2):

$$p_j = u(k_{i-1} + 1, k_i), \text{where } k_{i-1} + 1 \le j \le k_i, \qquad (4.3)$$

In addition, the following equation holds:

$$\frac{n_1 w_1 p_1 + n_2 w_2 p_2 + \cdots + n_k w_k p_k}{n_1 w_1 + n_2 w_2 + \cdots + n_k w_k} = \frac{D}{N}. \qquad (4.4)$$

Proof. As shown the proof of Proposition 3.1, values $\{p_i\}_{i=1}^{k}$ given by (4.3) satisfy (1.2) and (4.4). Next, for $i \le j$, let $SSE = \sum_{i=1}^{m} SSE(k_{i-1} + 1, k_i)$, where

$$SSE(k_{i-1} + 1, k_i) = \sum_{h=k_{i-1}+1}^{k_i}\sum_{g=1}^{n_h} w_h\left(y_{hg} - p_h\right)^2.$$

Because of (2.8), it suffices to show $SSE(k_{i-1} + 1, k_i)$ is minimized at $p_j = u(k_{i-1} + 1, k_i)$ subject to (1.2), where $k_{i-1} + 1 \le j \le k_i$. We show only the case $i = 1$ where $SSE(k_{i-1} + 1, k_i)$ is $SSE(1, k_1)$. The proof for other cases is similar. Without loss of generality, we assume $k_i = k$. In this case, $m = 1$ and $k_1 = k$, and $SSE(1, k) = SSE$.

Parameterize p_j by letting $p_1 = a_1$, and for 2≤ $j \le k$,

$$p_j = a_1 + \left(b_2 + \cdots + b_j\right), \;\; b_j = a_j^2 \qquad (4.5)$$

where $-\infty < \; a_j < +\infty$ for $1 \le j \le k$. With this parametrization, (1.2) is satisfied. By plugging (4.5) into (4.1), we transform the constrained optimization problem to a non-constrained mathematical programming problem. We take the partial derivative of SSE with respect to a_j. For $j \ge 2$, we have

$$\frac{\partial SSE}{\partial a_j} = \sum_{i=j}^{k}\sum_{g=1}^{n_i} -4a_j w_i\left(y_{ig} - p_i\right) = -4a_j \sum_{i=j}^{k} w_i(d_i - n_i p_i) = -4a_j f(j)$$

where $f(j) = \sum_{i=j}^{k} w_i(d_i - n_i p_i)$. Setting this derivative to zero, we have either $a_j = 0$ or $f(j) = 0$.

For $j = 1$, we have

$$\frac{\partial SSE}{\partial a_1} = \sum_{i=1}^{k}\sum_{g=1}^{n_i} -2w_i\left(y_{ig} - p_i\right) = -2\sum_{i=1}^{k} w_i(d_i - n_i p_i) = -2f(1).$$

Setting this derivative to zero, we have

$$0 = f(1) = \sum_{i=1}^{k} w_i(d_i - n_i p_i).$$

This implies:

$$\frac{n_1 w_1 p_1 \quad + n_2 w_2 p_2 + \cdots + n_k w_k p_k}{n_1 w_1 + n_2 w_2 + \cdots + n_k w_k} = \frac{D}{N} = u(1, k).$$

This shows that the weighted least squares estimates $\{p_i\}_{i=1}^{k}$, before their true values are found, satisfy (4.4). We claim that $a_j = 0$ for all $1 < j \le k$. If this is true, then $p_1 = p_2 = \cdots = p_k$. Then by (4.4) we have $p_1 = \frac{D}{N} = u(1, k)$, and the proof follows. Otherwise, let $i_0 > 1$ be the smallest integer such that $a_{i_0} \ne 0$, and $a_j = 0$ whenever $1 < j < i_0$. Then we have $f(1) = 0$ and $f(i_0) = 0$. Thus

$$0 = f(1) - f(i_0) = \sum_{i=1}^{i_0 - 1} w_i(d_i - n_i p_i). \tag{4.6}$$

Since $a_j = 0$ when $1 < j < i_0$, all $\{p_j\}_{j=1}^{i_0 - 1}$ are equal to p_1. Thus by (4.6) and (2.2), we have

$$p_1 = \frac{d_1 w_1 + d_2 w_2 + \cdots + d_{i_0 - 1} w_{i_0 - 1}}{n_1 w_1 + n_2 w_2 + \cdots + n_{i_0 - 1} w_{i_0 - 1}} = u(1, i_0 - 1). \tag{4.7}$$

However, $a_{i_0} \ne 0$, thus $p_1 < p_{i_0}$. Thus by (4.4), (1.2), and (4.7), we have

$$\frac{D}{N} = \sum_{i=1}^{k} \frac{n_i w_i}{N} p_i > \sum_{i=1}^{k} \frac{n_i w_i}{N} p_1 = p_1 = u(1, i_0 - 1).$$

Thus, we have $u(1, i_0 - 1) < \frac{D}{N} = u(1, k)$. This contradicts the fact that $j = k$ is the largest index where $u(1, j)$ reaches its minimum for all $j \ge 1$. Therefore, we have $a_2 = a_3 = \cdots = a_k = 0$, and all $\{p_i\}_{i=1}^{k}$ are equal to p_1.

5. Algorithms for least squares estimates or maximum likelihood estimates with monotonic constraints

First, we propose a non-parametric search algorithm, with time complexity $O(k^2)$, for finding the partition integers $0 = k_0 < k_1 < \cdots < k_m = k$ defined by (2.6) and (2.7), and then calculate by (3.2) or (4.3) for the estimates $\{p_i\}_{i=1}^{k}$ subject to (1.2).

Algorithm 5.1 (Non-parametric). Set $k_0 = 0$. Assume that partition integers $\{k_h\}_{h=1}^{i-1}$ have been found for an integer $i > 0$, and that $\{p_j\}_{j=1}^{k_{i-1}}$ have been calculated by (3.2) or (4.3).

(a) Scan into the remaining indexes range $k_{i-1} + 1 \le j \le k$ for a value $j = k_i$ such that

$$u(k_{i-1} + 1, j) = \frac{d_{k_{i-1}+1} w_{k_{i-1}+1} + d_{k_{i-1}+2} w_{k_{i-1}+2} + \cdots + d_j w_j}{n_{k_{i-1}+1} w_{k_{i-1}+1} + n_{k_{i-1}+2} w_{k_{i-1}+2} + \cdots + n_j w_j}$$

reaches its minimum in the range $k_{i-1}+1 \leq j \leq k$, and $j = k_i$ is the largest index for this minimum.

(b) Calculate $p_j,\ k_{i-1}+1 \leq j \leq k_i$, by (3.2) or (4.3) as $u(k_{i-1}+1, k_i)$.

Repeat steps (a) and (b) until there are no more remaining indexes to partition.

For the optimization problems of (1.1) and (1.3), when general monotonic constraints are required (including strictly monotonic constraints), we propose the following parametric algorithm, which can be implemented by using SAS procedure PROC NLMIXED [14].

Algorithm 5.2 (Parametric). For problem (1.1), parameterize p_i by letting $p_1 = a_1$, and for 2$\leq i \leq k$,

$$p_i = a_1 + \left(b_2 + \cdots + b_j\right),\ \ b_i = a_i^2 + \epsilon_i,\ 2 \leq i \leq k \tag{5.1}$$

where $\{\epsilon_i \geq 0\}_{i=1}^{k}$ are the given constants. Then $p_i - p_{i-1} \geq \epsilon_i$. For problem (1.3), let $b_1 = a_1^2$, and $b_i = a_i^2 + \epsilon$ for $2 \leq i \leq k$, where $\epsilon \geq 0$. Parameterize p_i by letting

$$p_{k+1-i} = \exp\left(-(b_1 + b_2 + \cdots + p_i)\right). \tag{5.2}$$

Then $p_i/p_{i-1} \geq \exp(\epsilon)$. Plug the corresponding parameterization into (1.1) or (1.3) and perform the non-constrained mathematical programming to obtain the estimates $\{a_i\}_{i=1}^{k}$, hence $\{p_i\}_{i=1}^{k}$ by (5.1) and (5.2).

6. Applications

6.1. Isotonic regression

Given real numbers $\{r_i\}_{i=1}^{k}$, the task of isotonic regression is to find $\{p_i\}_{i=1}^{k}$ that minimize the weighted sum squares $\sum_{i=1}^{k} w_i(r_i - p_i)^2$, where $\{w_i\}_{i=1}^{k}$ are the given weights. When w_i is 1 and r_i takes value 0 or 1 for all i's, it is known [13] that the results for isotonic regression coincide with the maximum likelihood estimates subject to (1.2) for log-likelihood $\sum_{i=1}^{k}[r_i \log(p_i) + (1-r_i)\log(1-p_i)]$.

A unique exact solution to the isotonic regression exists and can be found by a non-parametric algorithm called Pool Adjacent Violators (PAV) [2]. The basic idea as described in [5] is the following: Starting with r_1, we move to the right and stop at the first place where $r_i > r_{i+1}$. Since r_{i+1} violates the monotonic assumption, we pool r_i and r_{i+1} replacing both with their weighted average. Call this average $r_i^* = r_{i+1}^*$=$(w_i r_i + w_{i+1} r_{i+1})/(w_i + w_{i+1})$. We then move to the left to make sure that $r_{i-1} \leq r_i^*$—if not, we pool r_{i-1} with r_i^* and r_{i+1}^* replacing these three with their weighted average. We continue to the left until the monotonic requirement is satisfied, then proceed again to the right (see [2,3,5,8]). This algorithm finds the exact solution via forward and backward averaging. Another parametric algorithm, called Active Set Method, approximates the solution using the Karush-Kuhn-Tucker (KKT) conditions for linearly constrained optimization [3,9].

The algorithm PAV repeatedly searches both backward and forward for violators and takes average whenever a violator is found. In contrast, Algorithm 5.1 determines explicitly the groups of consecutive indexes by a forward search for the partition integers. Average is to be taken over each of these groups. For Algorithm 5.2, the constrained optimization is transformed into a non-constrained mathematical programming, through a re-parameterization. No KKT conditions and active set method are used.

6.2. An empirical example: the fair risk scales over a rating system

In this section, we show an example how the non-parametric search algorithm (Algorithm 4.1, labelled as "NPSM") can be used for estimation of the default risk scales with monotonic constraints for a rating system. We use the following two benchmarks:

EXP-CDF—The method proposed by Burgt [17]. The rating level PD is estimated by $p_i = \exp(a + bx)$, where x denotes, for a rating R_i, the adjusted sample cumulative distribution:

$$x(i) = \frac{n_1+n_2+\cdots+n_i}{n_1+n_2+\cdots+n_k} \tag{6.1}$$

where $\{n_i\}_{i=1}^{k}$ are defined as in Section. 1. Instead of estimating parameters via cap ratio [17], we estimate parameters by maximizing the log likelihood (1.3).

LGST-INVCDF—The method proposed by Tasche [16]. The rating level PD is estimated by $p_i = \frac{1}{1+\exp(a+b\Phi^{-1}(x))}$, where x is as in (6.1), and Φ^{-1} is the inverse of the cumulative distribution for the standard normal distribution. Parameters are estimated by maximizing the log likelihood (1.3).

The sample consists of the default and non-default frequencies for six non-default ratings (labelled as "RTG" in Table 1 below). Table 1 shows the number of defaults by rating (labelled as "D") in the sample, the count by rating (labelled as "N"), and the default rate (labelled as "DFR"). The third row denotes the sample distribution (labelled as "Dist"). It is assumed that lower index ratings carry higher default risks. For the proposed method "NPSM" in table 1, we need to first reverse the indexes of ratings and then apply Algorithm 4.1.

The quality of an estimation is measured by log-likelihood (labelled as "LL", larger values are better), the sum squared error (labelled as "SSE" in the sense of (1.1), smaller values are better), the portfolio level count-weighted average of the estimates (labelled as "AVG", closer to the sample portfolio default rate is better).

As shown in Table 1, the sample default rate is not monotonic between ratings 2 and 3. The proposed non-parametric algorithm (NPSM) simply takes the average. It gets the highest log-likelihood, the lowest sum squared error, and its count-weighted average is the same as the sample portfolio default rate. While for the other two benchmarks, the sum squared error is higher. Both overestimate the risk for ratings 4, 5, and underestimate the risk for ratings 1, 2, 3 and 6.

Table 1. Smoothing rating level default rate.

RTG	1	2	3	4	5	6	LL	AVG	SSE
D	1	11	22	124	62	170			
N	5529	11566	29765	52875	4846	4318			
Dist	5%	11%	27%	49%	4%	4%			
DFR	0.0173%	0.0993%	0.0739%	0.2352%	1.2833%	3.9442%	-2208.01	0.003594	0
NPSM	0.0173%	0.0810%	0.0810%	0.2352%	1.2833%	3.9442%	-2208.33	0.003594	0.00053
EXP-CDF	0.0061%	0.0086%	0.0294%	0.3431%	1.9081%	2.5057%	-2264.46	0.003601	1.15966
LGST-INVCDF	0.0104%	0.0188%	0.0585%	0.2795%	1.5457%	3.4388%	-2223.17	0.003594	0.16221

6.3. *Risk-supervised monotonic binning for univariate data*

Given a sample, let $S_x = \{x_i\}_{i=1}^{k}$ be the order set of all the distinct sample values of an explanatory variable x ordered by $x_i < x_{i+1}$. Denote by $\{y_{ij}\}_{j=1}^{n_i}$ the set of all the observed y-values conditional on $x = x_i$. Discretization of continuous attributes are usually required in machine learning processes [7]. Binning is also widely used in retail portfolio credit scoring [1,4,15]. A discretization or binning of a numerical variable x consists of a list of partition numbers $\{c_i\}_{i=1}^{M}$ and intervals $\{I_i\}_{i=1}^{M}$, where

$$-\infty = c_0 < c_1 < \cdots < c_{M-1} < c_M = +\infty,$$
$$I_1 = (-\infty, c_1],\ I_2 = \ (c_1, c_2],\ \ldots,\ I_{M-1} = (c_{M-2},\ c_{M-1}],\ I_M = \ (c_{M-1}, +\infty),$$

where each intersection $B_i = S \cap I_i$ is non-empty for all $1 \le i \le M$. Let N_i denote the number of observations in B_i and $b_i = \frac{1}{N_i}\sum_{x_h \in B_i}\sum_{j=1}^{n_h} y_{hj}$, the sample average of y over B_i. A monotonic binning for the explanatory variable x is a binning where $\{b_i\}_{i=1}^{M}$ satisfy the monotonic condition (6.2) or (6.3) below:

$$b_1 < b_2 < \cdots < b_M, \tag{6.2}$$

$$b_1 > b_2 > \cdots > b_M. \tag{6.3}$$

The quality of a binning can be measured by its sum squared error (smaller values are better), which is defined as:

$$SSE = \sum_{i=1}^{M}\sum_{x_j \in B_i}\sum_{h=1}^{n_i}\left(y_{jh} - b_i\right)^2$$
$$= \sum_{i=1}^{M}\sum_{x_j \in B_i}\sum_{h=1}^{n_i}\left(y_{jh} - r_j\right)^2 + \sum_{i=1}^{M}\sum_{x_j \in B_i} n_i\left(r_j - b_i\right)^2 = SSE_A + SSE_B$$

where $SSE_A = \sum_{i=1}^{M}\sum_{x_j \in B_i}\sum_{h=1}^{n_i}\left(y_{jh} - r_j\right)^2$, and

$$SSE_B = \sum_{i=1}^{M}\sum_{x_j \in B_i} n_i\left(r_j - b_i\right)^2. \tag{6.4}$$

Because SSE_A does not depend on the binning, the minimization of the sum squared error SSE by binning depends only on the minimization of SSE_B.

When y is binary or takes values in the range $0 \le y \le 1$, the quality of the binning can also be measured by the log-likelihood (Bernoulli, or quasi-Bernoulli, or binomial) (high values are better) as

$$LL_B = \sum_{i=1}^{M}\sum_{x_j \in B_i}[d_j \log(b_i) + \left(n_j - d_j\right)\log(1 - b_i)]. \tag{6.5}$$

With the estimates given by Propositions 3.1 and 4.1 and in absence of the bin size requirements, a preliminary but the best monotonic binning, in the sense of maximum likelihood or minimum sum square error subject to (6.2), can be obtained as:

$$I_1 = \left(-\infty, x_{k_1}\right], I_2 = \left(x_{k_1}, x_{k_2}\right], \ldots, I_{m-1} = \left(x_{k_{m-2}}, x_{k_{m-1}}\right], I_m = \left(x_{k_{m-1}}, +\infty\right)$$

where $\{k_i\}_{i=0}^{m}$ are the partition integers by (2.6) and (2.7).

7. Conclusions

This paper shows that the maximum Bernoulli likelihood (or quasi-Bernoulli likelihood) estimates with monotonic constraints are each given by the average risk observed over some consecutive indexes. These estimates coincide with the least squares estimates with the same monotonic constraints. The proposed non-parametric algorithm provides a resolution to flip-over credit risk, and a tool to determine the fair risk scales over a rating system.

Acknowledgements

The author is very grateful to the second reviewer for suggesting the title and the removal of (1.5) as a condition in Propositions 3.1 and 4.1 in the original version, and his verification of Propositions 3.1 and 4.1 by alternative approaches. The first paragraph comes directly from the comments by the second reviewer.

The author also thanks Carlos Lopez for his supports for this research. Special thanks go to Clovis Sukam his critical reading for this manuscript, and Biao Wu, Glenn Fei, Wallace Law, Zunwei Du, Kaijie Cui, Lan Gong, Wilson Kan, and Amada Huang for many valuable conversations.

Conflict of interest:

The views expressed in this article are not necessarily those of Royal Bank of Canada or any of its affiliates. Please direct any comments to the author Bill Huajian Yang at: h_y02@yahoo.ca.

References

1. Anderson R, (2007) *The Credit Scoring Toolkit: Theory and Practice for Retail Credit Risk Management and Decision Automation,* Oxford: Oxford University Press.
2. Barlow RE, Bartholomew DJ, Bremner JM, et al. (1972) *Statistical inference under order restrictions: The theory and application of isotonic regression,* New York: Wiley.
3. Best MJ, Chakravarti N, (1990) Active set algorithms for isotonic regression; A unifying framework. *Math Program* 47: 425–439.
4. Eichenberg T, (2018) Supervised weight of evidence binning of numeric variables and factors, R-Package Woebinning.

5. Friedman J, Tibshirani R, (1984) The monotone smoothing of scatterplots. *Technometrics* 26: 243–250.
6. Kotlowski W, Slowinski R, (2009) Rule learning with monotonicity constraints. *Proceedings of the 26th Annual International Conference on Machine Learning*, 537–544.
7. Kotsiantis S, Kanellopoulos D, (2006) Discretization techniques: A recent survey. *GESTS Int Trans Comput Sci Engin* 32: 47–58.
8. Leeuw JD, Hornik K, Mai P, (2009) Isotone optimization in R: Pool-adjacent-violators algorithm (PAVA) and active set methods. *J stat software* 32.
9. Nocedal J, Wright SJ, (2006) *Numerical Optimization,* 2 Eds.,New York: Springer.
10. Papke LE, Wooldrige JM, (1996) Econometric methods for fraction response variables with application to 401 (k) plan participation rates. *J Appl Econometrics* 11: 619–632.
11. Potharst R, Feelders AJ, (2002) Classification trees for problems with monotonicity constraints. *SIGKDD Explor* 14: 1–10.
12. Ramsay JO, Wickham H, Graves S, et al. (2018) Package ‘fda’-CRAN.R-Project, 265–267
13. Robertson T, Wright FT, Dykstra RL, (1998) *Order Restricted Stat Inference*, New Jersey: John Wiley and Sons.
14. SAS Institute Inc (2014) SAS/STAT(R) 13.2 User’s Guide.
15. Siddiqi N, (2006) Credit risk scorecards: Developing and implementing intelligent credi scoring. *Hoboken,* New Jersey: John Wiley and Sons.
16. Tasche D, (2013) The art of PD curve calibration. *J Credit Risk* 9: 63–103.
17. Van der Burgt M, (2008) Calibrating low-default portfolios, using the cumulative accuracy profile. *J Risk Model validation* 1: 17–33.
18. Yang BH, (2018) Smoothing algorithms by constrained maximum likelihood. *J Risk Model Validation* 12: 89–102.
19. You S, Ding D, Canini K, et al. (2017) Deep lattice networks and partial monotonic functions. *31st Conf Neural Inf Process Syst (NIPS).*