Monotonic Estimation for Probability Distribution and Multivariate Risk Scales by Constrained Minimum Generalized Cross-Entropy

Yang, Bill Huajian

March 2019

Monotonic Estimation for Probability Distribution and Multivariate Risk Scales by Constrained Minimum Generalized Cross-Entropy

Bill Huajian Yang

***Abstract* - Minimum cross-entropy estimation is an extension to the maximum likelihood estimation for multinomial probabilities. Given a probability distribution $\{r_i\}_{i=1}^{k}$, we show in this paper that the monotonic estimates $\{p_i\}_{i=1}^{k}$ for the probability distribution by minimum cross-entropy are each given by the simple average of the given distribution values over some consecutive indexes. Results extend to the monotonic estimation for multivariate outcomes by generalized cross-entropy. These estimates are the exact solution for the corresponding constrained optimization and coincide with the monotonic estimates by least squares. A non-parametric algorithm for the exact solution is proposed. The algorithm is compared to the "pool adjacent violators" algorithm in least squares case for the isotonic regression problem. Applications to monotonic estimation of migration matrices and risk scales for multivariate outcomes are discussed.**

***Index terms* –maximum likelihood, cross-entropy, least squares, isotonic regression, constrained optimization, multivariate risk scales**

I. INTRODUCTION

Utilizing prior knowledge is important for a learning process. A common prior is the monotone relationship between input and output. For example, we expect the loss for a loan to be lower when collateral value and quality of collateral type are higher; and people tend to buy less of a product when price increases. Examples of learnings, where monotonic constraints are imposed, include isotonic regression ([1], [2], [3], [4]), rating migration models ([5]), classification trees ([6]), rule learning ([7]), binning ([8]), and deep lattice network ([9]).

For a random vector $(y_1, y_2, \ldots, y_k)$, let p_i be the expected value of y_i, and $S = \{(y_{i1}, y_{i2}, \ldots, y_{ik})\}_{i=1}^{n}$ a given sample for the random vector, where $(y_{i1}, y_{i2}, \ldots, y_{ik})$ denotes the i^{th} observation, and y_{ij} its j^{th} component. We assume:

$$0 \le y_{ij} \le 1,\ 1 \le j \le k, 1 \le i \le n, \quad (1.1)$$
$$y_{i1} + \ y_{i2} + \ \ldots + y_{ik} = 1. \quad (1.2)$$

That is, each observation $(y_{i1}, y_{i2}, \ldots, y_{ik})$ is a percentage distribution over k ordinal indexes. We use the following notations: $d_j = \sum_{i=1}^{n} y_{ij}, r_j = d_j/n, D = d_1 + d_2 + \cdots + d_k$, and $R = \frac{D}{n}$.

Given an observed distribution $q = \{q_i\}_{i=1}^{k}$ and the predicted distribution $p = \{p_i\}_{i=1}^{k}$, the cross-entropy between q and p is defined as:

$$H(q, p) = -\sum_{i=1}^{k} q_i \log(p_i).$$

By using the Kullback-Leibler (KL) divergence (also called relative entropy) between q and p ([10]), one can show that cross-entropy $H(q, p)$ measures the dissimilarity between q and p (see Appendix, or [11], [12]).The cross-entropy for the given sample S is defined as:

$$\begin{aligned} CE &= -\sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log(p_j) \\ &= -\sum_{j=1}^{k} (\sum_{i=1}^{n} y_{ij}) \log(p_j) \\ &= -\sum_{j=1}^{k} d_j \log(p_j). \end{aligned} \quad (1.3)$$

The monotonic minimum cross-entropy estimates are the values $\{p_j\}_{j=1}^{k}$ that minimize (1.3) subject to (1.4) and (1.5) below:

$$0 \le p_1 \le p_2 \le \cdots \le p_k, \quad (1.4)$$
$$p_1 + p_2 + \cdots + p_k = 1. \quad (1.5)$$

Because $-\sum_{j=1}^{k} d_j \log(p_j) = -n \sum_{j=1}^{k} r_j \log(p_j)$, the measure CE defined by (1.3) is the same as the cross-entropy between $\{r_j\}_{j=1}^{k}$ and $\{p_j\}_{j=1}^{k}$, up to a scalar n. Therefore, CE measures the dissimilarity between $\{r_j\}_{j=1}^{k}$ and $\{p_j\}_{j=1}^{k}$.

When each observation $(y_{i1}, y_{i2}, \ldots, y_{ik})$ is multinomial, i.e. all $y_{i1}, y_{i2}, \ldots$, and y_{ik} are zero but one, which is 1, then d_i becomes the frequency that y_i takes the value 1. Therefore, (1.3) is the negative multinomial log-likelihood, up to a constant given by the logarithm of some multinomial coefficient, which is independent of $\{p_j\}_{j=1}^{k}$. In this case, the minimum cross-entropy estimates are the maximum multinomial likelihood estimates.

Generalization to multivariate outcomes. Given a sample $S = \{(y_{i1}, y_{i2}, \ldots, y_{ik})\}_{i=1}^{n}$ of the random vector $(y_1, y_2, \ldots, y_k)$, where $y_{ij} \geq 0$ for $1 \leq j \leq k$ and $1 \leq i \leq n$ (in absence of (1.2)), the generalized cross-entropy is defined, similarly to CE, as:

$$GCE = -\sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log(p_j)$$
$$= -\sum_{j=1}^{k} d_j \log(p_j). \qquad (1.6)$$

The monotonic minimum generalized cross-entropy estimates are the values $\{p_j\}_{j=1}^{k}$ that minimize (1.6) subject to (1.4) and (1.7) below:

$$p_1 + p_2 + \ldots + p_k = R. \qquad (1.7)$$

Recall $R = \frac{D}{n}$. Clearly, minimizing (1.3) for CE subject to (1.4) and (1.5) is a special case of minimizing (1.6) for GCE subject to (1.4) and (1.7).

Main results. We show in this paper that for a given sample $S = \{(y_{i1}, y_{i2}, \ldots, y_{ik})\}_{i=1}^{n}$, where $y_{ij} \geq 0$ for $1 \leq j \leq k$ and $1 \leq i \leq n$, there exist partition integers $\{k_i\}_{i=0}^{m}$, where $0 = k_0 < k_1 < \cdots < k_m = k$, such that the monotonic minimum generalized cross-entropy estimates $\{p_j\}_{j=1}^{k}$ that minimize (1.6) subject to (1.4) and (1.7) are given by the simple average (see Proposition 3.3) below:

$$p_j = \frac{1}{k_i - k_{i-1}} \sum_{j=k_{i-1}+1}^{k_i} r_j. \qquad (1.8)$$

One of the most important monotonic estimations is by least squares, i.e. the isotonic regression ([1]). The goal of isotonic regression is to find $\{p_i\}_{i=1}^{k}$, subject to (1.4), 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 exists and can be obtained by a non-parametric algorithm called Pool Adjacent Violators (PAV) ([1], [2], [4], [8]).

Results by (1.8) are the exact solution to the constrained optimization problem corresponding to (1.6) and are proved to be also the least squares estimates subject only to (1.4) (see Proposition 3.4), which links to isotonic regression. That is, for monotonic least squares estimates, (1.7) is an implication, while it is a condition (i.e. a constraint) for monotonic generalized cross-entropy estimates.

A non-parametric algorithm (Algorithm 4.1) is proposed in section IV for the partition integers in (1.8), hence the monotonic estimates. This algorithm is compared in section V to the PAV algorithm.

The key ideas to the proof of (1.8) and the algorithms proposed in this paper are the re-parameterization of the estimates so that (1.4) is automatically satisfied. Consequently, the constrained programming is transformed into a tractable non-constrained mathematical programming problem (see sections III and IV).

The paper is organized as follows: Partition integers are defined in section II. Equation (1.8) is proved in section III. We propose in section IV a non-parametric algorithm for finding these partition integers. In section V, we compare this non-parametric algorithm, in least squares case, with the Pool Adjacent Violators algorithm for isotonic regression. Two examples are provided in section V, where monotonic estimation for long-run rating migration matrices and loss rate time series are discussed.

II. THE PARTITION INTEGERS

Given a sample $S = \{(y_{i1}, y_{i2}, \ldots, y_{ik})\}_{i=1}^{n}$, where y_{ij} are real numbers, let where $d_j = \sum_{i=1}^{n} y_{ij}$ and $r_j = d_j/n$. Define:

$$v(i,j) = \frac{r_i + r_{i+1} + \cdots + r_j}{(j-i+1)} \qquad (2.1)$$
$$= \frac{d_i + d_{i+1} + \cdots + d_j}{n(j-i+1)}. \qquad (2.2)$$

Then $v(i,j)$ is the simple average for the consecutive values of $\{r_i, r_{i+1}, \ldots, r_j\}$, and $v(1,k) = \frac{D}{nk}$, where $d_1 + d_2 + \cdots + d_k = D$. Let $\{k_i\}_{i=0}^{m}$ be partition integers, where $0 = k_0 < k_1 < \cdots < k_m = k$, such that (2.3) and (2.4) below hold for each $i > 0$:

$$v(k_{i-1}+1, k_i)$$
$$= min\{v(k_{i-1}+1, j) \mid k_{i-1}+1 \leq j \leq k\}, \quad (2.3)$$
$$v(k_{i-1}+1, k_i) < v(k_{i-1}+1, k_i+1). \qquad (2.4)$$

That is, given k_{i-1}, the partition integer k_i is the largest index where $v(k_{i-1}+1, j)$ reaches its minimum at $j = k_i$ within the remaining range $k_{i-1}+1 \leq j \leq k$. By definition, when $\{r\}_{i=1}^{k}$ are strictly increasing, we have $m = k$ and $\{k_i\}_{i=1}^{m} = \{1, 2, \ldots, k\}$. By (2.3) and (2.4), we have:

$$v(1,k_1) < v(k_1+1, k_2) < \cdots <$$
$$v(k_{m-1}+1, k_m). \qquad (2.5)$$

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

$$v(1,k_2)$$
$$= \frac{k_1}{k_2} v(1,k_1) + \frac{k_2 - k_1}{k_2} v(k_1+1, k_2)$$
$$\leq v(1,k_1).$$

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

III. MONOTONIC ESTIMATION BY MINIMUM CROSS-ENTROPY

In this section, we prove equation (1.8), first for the minimum cross-entropy estimates subject to (1.4) and (1.5), then for the minimum generalized cross-entropy estimates subject to (1.4) and (1.7). At the end of the section, we show that these estimates are also the monotonic least squares estimates, in absence of (1.7).

Lemma 3.1. In absence of (1.4), the sample rates $\{r_i\}_{i=1}^{k}$ minimize (1.3) subject to (1.5). Similarly, in absence of (1.4), the sample rates $\{r_i\}_{i=1}^{k}$ minimize (1.6) subject to (1.7).

Proof. First, we show that the 1st statement implies the 2nd statement. The second statement in the lemma holds if $R = \frac{D}{n} = 0$ because, in this case, $d_i = 0$ and $r_i = 0$ for all $i's$. If $R > 0$, then:

$$GCE = -\sum_{j=1}^{k} d_j \log(p_j) \qquad (3.1)$$
$$= -R\sum_{j=1}^{k} d_j' [\log(p_j') + \log(R)]$$
$$= c - R\sum_{j=1}^{k} d_j' \log(p_j') \qquad (3.2)$$

where $d_j' = d_j/R, p_j' = p_j/R$, and $c = -\sum_{j=1}^{k} d_j \log(R)$. By (1.7), $\{p_j'\}_{j=1}^{k}$sum to one. Since $\{d_j'\}_{j=1}^{k}$ sum to n, the function $-R\sum_{j=1}^{k} d_j' \log(p_j')$ differs from (1.3), the formulation of CE, only by a constant scalar R. Therefore, if the first statement in the lemma holds, then $\{p_j' = r_j'\}_{j=1}^{k}$minimize (3.2), because R and c are constants, where $r_j' = \frac{d_j'}{n} = \frac{d_j}{nR} = r_j/R$. Since $p_j = p_j'R = r_j$, the sample rates $\{r_j\}_{j=1}^{k}$ minimize (3.1) subject to (1.7).

We now show the first statement. We consider the following three cases. Case (a). $0 < r_i < 1$ for all $1 \leq i \leq k$. Take the derivative of CE with respect to p_i in the range $0 < p_i < 1$ and set it to zero, using the relation $p_k = 1 - (p_1 + p_2 + \cdots + p_{k-1})$. We have $\frac{d_i}{p_i} - \frac{d_k}{p_k} = 0$. This holds for all $i's$. Thus the vector $(p_1, p_2, \ldots, p_k)$ is in proportion to $(d_1, d_2, \ldots, d_k)$, hence in proportion to $(r_1, r_2, \ldots, r_k)$. Because of (1.5), we must have $p_i = r_i$.

Case (b). $r_i = 1$ for some i. Then r_j are all zero but this r_i. In this case, CE reduces to $-d_i \log(p_i)$, which is minimized at $p_i = 1 (= r_i)$ within $0 \leq p_i \leq 1$.

Case (c). $r_i = 0$ for some $i's$ and $0 < r_j < 1$ for all other $r_j's$. Without loss of generality, we assume that i_0 is the integer where $0 < r_i < 1$ for $i \leq i_0$ and $r_i = 0$ for $i > i_0$. Then CE reduces to $-\sum_{i=1}^{i_0} d_i \log(p_i)$. Setting the derivatives with respect to p_i in the range $0 < p_i < 1$, $i \leq i_0$, to zero, and using the relation $p_{i_0} = 1 - (p_1 + p_2 + \cdots + p_{i_0 - 1} + p_{i_0 + 1} + \cdots + p_k)$, we have $\frac{d_i}{p_i} - \frac{d_{i_0}}{p_{i_0}} = 0$. This implies, $(p_1, p_2, \ldots, p_{i_0})$ is in proportion to $(r_1, r_2, \ldots, r_{i_0})$. Thus $p_i = sr_i$ for $i \leq i_0$ for a scalar $s > 0$. Because $r_i = 0$ for $i > i_0$, we have $\sum_{i=1}^{i_0} r_i$=1. Hence by (1.5), we have $0 < s \leq 1$. With $p_i = sr_i, 0 < s \leq 1$, for $i \leq i_0$, the function $-\sum_{i=1}^{i_0} d_i \log(p_i)$ reaches its minimum at $s = 1$, because $d_i \log(sr_i)$ is an increasing function of s. Therefore, $p_i = r_i$ for $i \leq i_0$. By (1.5), we must have $p_i = 0 = r_i$ for $i > i_0$. □

Proposition 3.2. Given a sample $S = \{(y_{i1}, y_{i2}, \ldots, y_{ik})\}_{i=1}^{n}$ subject to (1.1) and (1.2), let $\{k_i\}_{i=0}^{m}$ be the partition integers defined by (2.3) and (2.4). Then the minimum cross-entropy estimates $\{p_i\}_{i=1}^{k}$ that minimize (1.3) subject to (1.4) and (1.5) are given by:

$$p_j = v(k_{i-1} + 1, k_i)$$
$$= \frac{r_{k_{i-1}+1} + r_{k_{i-1}+2} + \cdots + r_{k_i}}{(k_i - k_{i-1})} \qquad (3.3)$$

where $k_{i-1} + 1 \leq j \leq k_i$.

Proof. First, with the values given by (3.3), (1.4) holds by (2.5), and $p_{k_{i-1}+1} + p_{k_{i-1}+2} + \cdots + p_{k_i} = r_{k_{i-1}+1} + r_{k_{i-1}+2} + \cdots + r_{k_i}$. Thus (1.5) holds as well for these specific values.

Next, with the partition integers $\{k_i\}_{i=0}^{m}$ given by (2.3) and (2.4), we have:

$$CE = -\sum_{j=1}^{k} d_j \log(p_j)$$
$$= CE(1, k_1) + CE(k_1 + 1, k_2) + \cdots +$$
$$CE(k_{m-1} + 1, k_m) \qquad (3.4)$$

where:

$$CE(k_{i-1} + 1, k_i) = -\sum_{j=k_{i-1}+1}^{k_i} d_j \log(p_j).$$

Case (a). $r_1 = 0$. In this case, $r_j = 0$ and $d_j = 0$ for all $1 \leq j \leq k_1$, this is because $j = k_1$ is the largest index such that $v(1, j)$ reaches its minimum within the range $1 \leq j \leq k$. Thus $CE(1, k_1) = 0$. Set $p_j = 0$

by (3.3) for all $1 \leq j \leq k_1$, drop out $CE(1, k_1)$ from (3.4), and focus only on $CE = E(k_1 + 1, k_2) + \cdots + CE(k_{m-1} + 1, k_m)$. By the definition of partition integers, we have $r_{k_1+1} > 0$. Essentially, dropping out indexes $1 \leq j \leq k_1$ is the same as assuming that the index starts from $k_1 + 1$. Therefore, the problem reduces to case (b) below.

Case (b). $r_1 > 0$. Let $p(k_{i-1} + 1, k_i) = p_{k_{i-1}+1} + p_{k_{i-1}+2} + \cdots + p_{k_i}$ and $d(k_{i-1} + 1, k_i) = d_{k_{i-1}+1} + d_{k_{i-1}+1} + \cdots + d_{k_i}$. Normalize $p_j's$ for $k_{i-1} + 1 \leq j \leq k_i$ by letting:

$$p_j^0 = \frac{p_j}{p(k_{i-1}+1, k_i)}, \; k_{i-1} + 1 \leq j \leq k_i.$$

Then $\{p_j^0 \mid k_{i-1} + 1 \leq j \leq k_i\}$ sum up to 1, and we have:

$$\begin{aligned} &CE(k_{i-1} + 1, k_i) \\ &\quad = -\textstyle\sum_{j=k_{i-1}+1}^{k_i} d_j \{\log(p_j^0) + \\ &\qquad\qquad \log[p(k_{i-1} + 1, k_i)]\}. \qquad (3.5) \end{aligned}$$

Then by (3.4) and (3.5), we have:

$$\begin{aligned} CE &= -\textstyle\sum_{i=1}^{m} \sum_{j=k_{i-1}+1}^{k_i} d_j \{\log(p_j^0) + \\ &\qquad\qquad \log[p(k_{i-1} + 1, k_i)]\} \\ &= -\textstyle\sum_{i=1}^{m} \sum_{j=k_{i-1}+1}^{k_i} d_j \log(p_j^0) \; - \\ &\quad \textstyle\sum_{i=1}^{m} d(k_{i-1} + 1, k_i) \log[p(k_{i-1} + 1, k_i)] \\ &= CE_1 + CE_2 \end{aligned}$$

where $CE_1 = -\sum_{i=1}^{m} \sum_{j=k_{i-1}+1}^{k_i} d_j \log(p_j^0)$, $CE_2 = -\sum_{i=1}^{m} d(k_{i-1} + 1, k_i) \log[p(k_{i-1} + 1, k_i)]$. By Lemma 3.1, CE_2 is minimized at:

$$p(k_{i-1} + 1, k_i) = d(k_{i-1} + 1, k_i)/n. \quad (3.6)$$

Let $CE_1(k_{i-1} + 1, k_i) = -\sum_{j=k_{i-1}+1}^{k_i} d_j \log(p_j^0)$. Then $CE_1 = CE_1(1, k_1) + CE_1(k_1 + 1, k_2) + \cdots + CE_1(k_{m-1} + 1, k_m)$. It suffices to show that each $CE_1(k_{i-1} + 1, k_i)$ is minimized at:

$$p_j^0 = \frac{1}{k_i - k_{i-1}}. \quad (3.7)$$

This is because, if (3.7) is true, then by (3.6), we have:

$$\begin{aligned} p_j &= p_j^0 p(k_{i-1} + 1, k_i) \\ &= \frac{1}{k_i - k_{i-1}} \frac{d(k_{i-1} + 1, \; k_i)}{n} \\ &= v(k_{i-1} + 1, k_i) \qquad (3.8) \end{aligned}$$

by (2.2). The proof is then complete.

We prove (3.7) only for $CE_1(1, k_1)$. The proof for other $CE_1(k_{i-1} + 1, k_i)$ is similar. Without loss of generality, we assume $m = 1$. In this case, $CE_1(1, k_1) = CE(1, k)$ and $p_j^0 = p_j$ for all $j's$.

For $1 \leq i \leq k$, parameterize p_i by:

$$p_i = \exp(b_1 + b_2 + \cdots + b_i) / \Delta \quad (3.9)$$

where $b_i = a_i^2$, $1 \leq i \leq k$, and $\Delta = \sum_{i=1}^{k} \exp(b_1 + b_2 + \cdots + b_i)$. Then by (3.9), $\{p_i\}_{i=1}^{k}$ satisfy (1.4) and (1.5). Let $c_0 = 0$ and $c_i = p_i + p_2 + \cdots + p_i$. The partial derivative of $d_i \log(p_i)$ with respect to a_j, when $j \leq i$, is:

$$\begin{aligned} &\frac{\partial d_i \log(p_i)}{\partial a_j} \\ &\quad = \left(\frac{2 d_i a_j}{p_i}\right) [p_i - p_i(p_j + p_{j+1} + \cdots + p_k)] \\ &\quad = 2 d_i a_j [1 - (p_j + p_{j+1} + \cdots + p_k)] \\ &\quad = 2 d_i a_j c_{j-1} \end{aligned}$$

using the relation $1 = c_k = p_1 + p_2 + \cdots + p_k$.

When $j > i$, we have:

$$\begin{aligned} &\frac{\partial d_i \log(p_i)}{\partial a_j} \\ &\quad = \left(\frac{2 d_i a_j}{p_i}\right) [-p_i(p_j + p_{j+1} + \cdots + p_k)] \\ &\quad = 2 d_i a_j [-(p_j + p_{j+1} + \cdots + p_k)] \\ &\quad = 2 d_i a_j (c_{j-1} - 1). \end{aligned}$$

Therefore, the partial derivative of $CE(1, k)$ with respect to a_j is:

$$\begin{aligned} \frac{\partial CE}{\partial a_j} &= -\textstyle\sum_{i=1}^{k} \frac{\partial d_i \log(p_i)}{\partial a_j} \\ &= -2 a_j \left(c_{j-1} \textstyle\sum_{i=1}^{k} d_i - \sum_{i=1}^{j-1} d_i\right) \\ &= -2 a_j \left(c_{j-1} n - \textstyle\sum_{i=1}^{j-1} d_i\right) = -2 a_j g(j) \end{aligned}$$

using the relation $\sum_{i=1}^{k} d_i = n$, where:

$$g(j) = \left(c_{j-1} n - \textstyle\sum_{i=1}^{j-1} d_i\right).$$

We claim $a_2 = a_3 = a_k = 0$. If this is true, then by (3.9), we have $p_1 = p_2 = \cdots = p_k = \frac{1}{k} = v(1, k)$. The proof follows. Otherwise, let $i_0 > 1$ be the smallest index such that $a_{i_0} \neq 0$. Then we have:

(i) $p_{i_0-1} < p_{i_0}$;
(ii) $p_1 = p_2 = \cdots = p_{i_0-1}$;
(iii) $g(i_0) = 0$.

Therefore, by (iii), we have:

$$0 = g(i_0) = c_{i_0-1}n - \sum_{i=1}^{i_0-1} d_i$$
$$\Rightarrow c_{i_0-1} = \sum_{i=1}^{i_0-1} \frac{d_i}{n} = (i_0 - 1)v(1, i_0 - 1).$$

By (ii), $c_{i_0-1} = (i_0 - 1)p_1$, thus we have $p_1 = v(1, i_0 - 1)$. This leads to the following:

$$1 = kv(1, k)$$
$$= p_1 + p_2 + .. + p_k > kp_1 = kv(1, i_0 - 1)$$

where the inequality follows from (i) and (1.4). Thus we have $v(1, i_0 - 1) < v(1, k)$.This contradicts the fact that $j = k$ is the largest index that $v(1, j)$ reaches it minimum for all $1 \le j \le k$. □

The following proposition generalizes the results of Proposition 3.2 to the case for the minimum generalized cross-entropy estimates.

Proposition 3.3. Given a sample $S = \{(y_{i1}, y_{i2}, \dots, y_{ik})\}_{i=1}^{n}$, where $y_{ij} \ge 0$, let $\{k_i\}_{i=0}^{m}$ be the partition integers defined by (2.3) and (2.4). The minimum generalized cross-entropy estimates $\{p_i\}_{i=1}^{k}$ that minimize (1.6) subject to (1.4) and (1.7) are given by:

$$p_j = \frac{r_{k_{i-1}+1} + r_{k_{i-1}+2} + \cdots + r_{k_i}}{(k_i - k_{i-1})} \qquad (3.10)$$

where $k_{i-1} + 1 \le j \le k_i$.

Proof. If $R = \frac{D}{n} = 0$, the proposition holds, because $d_j = 0$ for all $1 \le j \le k$. Assume $R > 0$. By (3.2), we have:

$$GCE = c - R \sum_{j=1}^{k} d'_j \log(p'_j)$$

where $d'_j = d_j/R, p'_j = p_j/R$, and $c = -R \sum_{j=1}^{k} d'_j \log(R)$. Since $\{p'_j\}_{j=1}^{k}$sum to one and $\{d'_j\}_{j=1}^{k}$ sum to n, the function $-R \sum_{j=1}^{k} d'_j \log(p'_j)$ differs from (1.3), the formulation of CE, only by a scalar R. Because R and c are constants, by Proposition 3.2, the minimum estimates of this function subject to (1.4) and (1.5) are given by $p'_j = \frac{r'_{k_{i-1}+1} + r'_{k_{i-1}+2} + \cdots + r'_{k_i}}{(k_i - k_{i-1})}$ for $k_{i-1} + 1 \le j \le k_i$, where $r'_j = \frac{d'_j}{n} = \frac{d_j}{nR} = r_j/R$. The equation (3.10) follows from the equations $p_j = Rp'_j$ and $r'_j = \frac{r_j}{R}$. □

Given a sample $S = \{(y_{i1}, y_{i2}, \dots, y_{ik})\}_{i=1}^{n}$, where y_{ij} are real numbers, we are interested in the least squares estimates $\{p_i\}_{i=1}^{k}$ that minimize (3.11) subject to (3.12) below:

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{n} \left(y_{ij} - p_j\right)^2, \qquad (3.11)$$
$$p_1 \le p_2 \le \cdots \le p_k. \qquad (3.12)$$

Proposition 3.4. *Let* $\{k_i\}_{i=0}^{m}$ be the partition integers defined by (2.3) and (2.4). The least squares estimates $\{p_j\}_{j=1}^{k}$ of (3.11) subject to (3.12) are given by:

$$p_j = v(k_{i-1} + 1, k_i)$$
$$= \frac{r_{k_{i-1}+1} + r_{k_{i-1}+2} + \cdots + r_{k_i}}{(k_i - k_{i-1})} \qquad (3.13)$$

where $k_{i-1} + 1 \le j \le k_i$. These estimates satisfy (1.7).

Proof. First, similarly to the proof of Proposition 3.2, these specific values for $\{p_j\}_{j=1}^{k}$ satisfy (1.7). By (2.5), (3.12) holds. Let:

$$SSE = \sum_{i=1}^{m} SSE(k_{i-1} + 1, k_i)$$

where:

$$SSE(k_{i-1} + 1, k_i)$$
$$= \sum_{j=k_{i-1}+1}^{k_i} \sum_{g=1}^{n} \left(y_{gj} - p_j\right)^2.$$

Because of (2.5), it suffices to show $SSE(k_{i-1} + 1, k_i)$ is minimized at $p_j = v(k_{i-1} + 1, k_i)$, where $k_{i-1} + 1 \le j \le k_i$. We show only the case when $i = 1$ for $SSE(1, k_1)$. The proof for other $SSE(k_{i-1} + 1, k_i)$ is similar. Without loss of generality, we assume $k_1 = k$. 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 $2 \le j \le k$:

$$p_j = a_1 + \left(b_2 + \cdots + b_j\right) \qquad (3.14)$$

where $b_i = a_i^2$. With this parametrization, (3.12) holds. Plug (3.14) into (3.11) and take the partial derivative of SSE with respect to a_j. For $j \ge 2$, we have:

$$\frac{\partial SSE}{\partial a_j} = -\sum_{i=j}^{k} \sum_{g=1}^{n} 4a_j\left(y_{gi} - p_i\right)$$
$$= -4a_j \sum_{i=j}^{k}(d_i - np_i) = -4a_j h(j)$$

where $h(j) = \sum_{i=j}^{k} (d_i - np_i)$. Setting this derivative to zero, we have either $a_j = 0$ or $h(j) = 0$. For $j = 1$, we have:

$$\frac{\partial SSE}{\partial a_1} = -\sum_{i=1}^{k} \sum_{g=1}^{n} 2\left(y_{gi} - p_i\right)$$
$$= -2\sum_{i=1}^{k} (d_i - np_i) = -2h(1).$$

Setting this derivative to zero, we have:

$$0 = h(1) = \sum_{i=1}^{k}(d_i - np_i)$$
$$\Rightarrow \sum_{i=1}^{k} p_i = \frac{d_1+d_2+\cdots+d_k}{n}$$
$$= \frac{D}{n} = R = kv(1,k). \quad (3.15)$$

We claim that $a_j = 0$ for all $1 < j \leq k$. If this is true, then $p_1 = p_2 = \cdots = p_k$. By (3.15), we have $p_1 = \frac{D}{nk} = v(1,k)$, and the proof follows. Otherwise, let $i_0, 1 < i_0 \leq k$, be the smallest index such that $a_j = 0\ when\ 1 < j < i_0$, and $a_{i_0} \neq 0$. Then we have $h(1) = 0$ and $h(i_0) = 0$. Thus:

$$0 = h(1) - h(i_0)$$
$$= \sum_{i=1}^{i_0-1}(d_i - np_i). \quad (3.16)$$

Since $a_j = 0$ for $1 < j < i_0$, we have $p_1 = p_2 = \cdots = p_{i_0-1}$. Thus by (3.16), we have:

$$p_1 = \frac{d_1+d_2+\cdots+d_{i_0-1}}{n\,(i_0-1)} = v(1, i_0 - 1). \quad (3.17)$$

Since $a_{i_0} > 0$, we have $p_1 < p_{i_0}$, hence $\sum_{i=1}^{k} p_i > kp_1$ by (3.12). Thus by (3.17) and (3.15), we have:

$$kv(1, i_0 - 1)$$
$$= kp_1 < \sum_{i=1}^{k} p_i = kv(1,k)$$
$$\Rightarrow v(1, i_0 - 1) < v(1,k).$$

This contradicts the fact that $j = k$ is the largest index that $v(1,j)$ reaches it minimum for all $1 \leq j \leq k$. □

IV. ALGORITHMS FOR MONOTONIC ESTIMATION BY MINIMUM GENERALISED CROSS-ENTROPY

In this section we propose algorithms for finding monotonic estimates. First, we propose a non-parametric algorithm with time complexity $O(k^2)$ for the partition integers, hence the exact solution for the monotonic estimates.

Algorithm 4.1 (Non-parametric). Set $k_0 = 0$. Assume that partition integers $\{k_j\}, 0 \leq j \leq i-1$, have been found for an integer $i > 0$, and that $\{p_j\}$, $1 \leq j \leq k_{i-1}$, have been calculated by (3.10) or (3.13). Scan into the remaining indexes range $k_{i-1} + 1 \leq j \leq k$ for a value $j = k_i$ such that

$$v(k_{i-1} + 1, j) = \frac{r_{k_{i-1}+1}+r_{k_{i-1}+2}+\cdots+r_j}{(j-k_{i-1})}$$

reaches its minimum for all $k_{i-1} + 1 \leq j \leq k$, and $j = k_i$ is the largest index for this minimum. Calculate $\{p_j\}$, $k_{i-1} + 1 \leq j \leq k_i$, by (3.10) or (3.13) as $v(k_{i-1} + 1, k_i)$. Repeat this process until $k_i = k$. □

Next, we propose a parametric algorithm as below, which can be implemented by using SAS procedure PROC NLMIXED ([13]), for an approximation of the estimates minimizing (1.6) subject to (1.4) and (1.7) (or strictly monotonic constraints: $0 \leq p_1 < p_2 < \cdots < p_k$). Recall that $R = \sum_{i=1}^{k} r_i = D/n$.

Algorithm 4.2 (Parametric). Assume $R > 0$. Parameterize p_j by:

$$p_j = \frac{Rw_j}{w_1+w_2+\cdots+w_k}, \quad (4.1)$$

where:

$$w_j = \exp(\,b_1 + b_2 + \cdots + b_j), \quad (4.2)$$

and $b_i = a_i^2 + \epsilon,\ 1 \leq i,j \leq k$. Then $\sum_{i=1}^{k} p_i = R$ and $\frac{p_i}{p_{i-1}} \geq \exp(\epsilon)$. Here $\epsilon \geq 0$ is an appropriately selected constant for the desired monotonicity. Plug (4.1) into (1.6) and perform a non-constrained optimization to obtain the estimates $\{a_i\}_{i=1}^{k}$, hence $\{p_i\}_{i=1}^{k}$ by (4.1) and (4.2). □

V. APPLICATIONS

A. 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 ([14]) that the results for isotonic regression coincide with the maximum likelihood estimates subject to (1.4) for the Bernoulli 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 obtained by a non-parametric algorithm called Pool Adjacent Violators (PAV) ([1]). The basic idea, as described in [4], 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 [1], [2], [4], [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, [15]) conditions for linearly constrained optimization ([2], [8]).

For a given sample $S = \{(y_{i1}, y_{i2}, \dots, y_{ik})\}_{i=1}^{n}$, where y_{ij} are real numbers, the sum-squares-error SSE in (3.11) can be rewritten as:

$$\begin{aligned} SSE &= \sum_{j=1}^{k}\sum_{i=1}^{n}\left(y_{ij} - p_j\right)^2 \\ &= \sum_{j=1}^{k}\sum_{i=1}^{n}\left(y_{ij} - r_j\right)^2 + \\ &\quad \sum_{j=1}^{k} n\left(r_j - p_j\right)^2 = SSE_1 + SSE_2 \end{aligned}$$

where $SSE_1 = \sum_{j=1}^{k}\sum_{i=1}^{n}\left(y_{ij} - r_j\right)^2$, $SSE_2 = \sum_{j=1}^{k} n\left(r_j - p_j\right)^2$, $r_j = \frac{d_j}{n}$, and $d_j = \sum_{i=1}^{n} y_{ij}$. Because SSE_1 does not depend on parameters $\{p_j\}_{j=1}^{k}$, the estimates that minimize SSE subject to (3.12) are the same as the estimates that minimize SSE_2 subject to (3.12). Hence, the least squares estimates $\{p_j\}_{j=1}^{k}$of (3.11) subject to (3.12) are the solution to the isotonic regression problem where weights w_i are equal to n.

The algorithm PAV repeatedly searches both backward and forward for violators and takes average whenever a violator is found. In contrast, Algorithm 4.1 determines explicitly the groups of consecutive indexes by a forward search for partition integers. Average is then to be taken over each of these groups. For Algorithm 4.2, the constrained optimization is transformed into a non-constrained mathematical programming, through a re-parameterization. No KKT conditions and active set method are used.

B. Monotonic Estimation of Risk Scales for Multivariate Outcomes

In this section, we show two examples on how the proposed algorithms can be used for monotonic estimation for a loss rate time series and a long-run migration matrix. Parametric methods for monotonic estimation of long-run migration matrices was discussed in ([5]).

In the first example, a loan portfolio is observed for loss for each loan since the account is opened. The 1st row in Table 1 below shows the yearly (since account open date) loss rate for the portfolio for 25000 accounts (i.e. $n = 25000$). The rate is calculated as the ratio of the total loss amount in a year divided by the total initial balance at open date for the portfolio. It is assumed that the loss rate is decreasing as loans survive through time.

The non-parametric algorithm (Algorithm 4.1, labelled as "NPSM") is used, by reversing the time index, to obtain the monotonic least squares estimates for 10 yearly rates. As a result, simple average is taken for cell groups $\{1, 2\}$ and $\{8, 9, 10\}$ respectively. For other cells the rate is kept unchanged. Strictly monotonic least squares estimates are obtained by using algorithm 4.2 (labelled as "PSM", where ϵ in the algorithm is chosen to satisfy $\exp(\epsilon) = 1.05$).

A benchmark model of the form $p_i = a + be^{\lambda t_i}$ is calibrated, where t_i denotes the time since account opening, with parameters being estimated by least squares regression. This is a simplified model for monotonic continuous yield curve used by Nelson and Siegel ([16], pp.483). We label this approach by "NSSM".

As shown in the table, the non-parametric algorithm gets the lowest sum squared error (labelled as "SSE").

TABLE 1. SMOOTHING LOSS RATE SERIES

	Loss rate by year since open									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**SSE**
	5.000%	6.500%	5.000%	4.000%	5.000%	3.500%	3.000%	2.500%	3.000%	3.000%	0.00000
NPSM	5.750%	5.750%	5.000%	4.500%	4.500%	3.500%	3.000%	2.830%	2.830%	2.830%	4.47917
PSM	5.890%	5.610%	5.000%	4.610%	4.390%	3.500%	3.089%	2.942%	2.802%	2.668%	6.70271
NSSM	5.723%	5.332%	4.948%	4.572%	4.203%	3.841%	3.486%	3.138%	2.796%	2.462%	9.85846

In the second example, the non-parametric algorithm is used to "smooth" the long-run average rating migration matrix for a portfolio with six non-default ratings. It is expected that an entity will migrate to the closer non-default rating than a faraway non-default rating, i.e. the following conditions are required for each i^{th}row in the long-run average migration matrix:

$$p_{i,i+1} \geq p_{i,i+2} \geq \cdots \geq p_{i,k}, \qquad (5.1)$$
$$p_{i,1} \leq p_{i,2} \leq \cdots \leq p_{i,i-1} \qquad (5.2)$$

where $p_{i,j}$ denotes the probability of migrating from non-default rating i to non-default rating j, conditional on that it migrates to a non-default rating. Smoothing of a given migration matrix means the action of modifying the migration matrix subject to (5.1) and (5.2) with minimum loss (cross-entropy).

Table 2 below shows the sample long-run average rating migration matrix before smoothing, conditional on migrating to a non-default rating, calculated from the historical sample generated synthetically between 2007Q1 and 2017Q1 for a commercial portfolio. There are six non-default ratings. Three highlighted blocks violate (5.1) or (5.2).

For the i^{th} row of the migration matrix, we let n_{ij} and r_{ij} denote respectively the observed frequency and rate migrating from i^{th} rating to j^{th} rating, conditional on migrating to a non-default rating. Let $n_1 = n_{i1} + n_{i2} + \cdots + n_{ii-1}$, and $n_2 = n_{i\,i+1} + n_{i\,i+2} + \cdots + n_{i6}$. For $j < i$, let $p_{ij}^0 = p_{ij}/p_1$, where $p_1 = p_{i1} + p_{i2} + \cdots + p_{i\,i-1}$. For $j > i$, let $p_{ij}^0 = p_{ij}/p_2$, where $p_2 = p_{i\,i+1} + p_{i\,i+2} + \cdots + p_{i6}$.The log-likelihood for a specific i^{th} row of the migration matrix is:

$$\begin{aligned} LL = \textstyle\sum_{j=1}^{6} n_{ij}\log(p_{ij}) &= n_{ii}\log(p_{ii}) + \\ &\textstyle\sum_{j=1}^{i-1} n_{ij}\log(p_{ij}) + \sum_{j=i+1}^{6} n_{ij}\log(p_{ij}) \\ = n_{ii}\log(p_{ii}) + n_1\log(p_1) + n_2\log(p_2) + \\ \textstyle\sum_{j=1}^{i-1} n_{ij}\log(p_{ij}^0) + \sum_{j=i+1}^{6} n_{ij}\log(p_{ij}^0) \\ = LL_1 + \quad \textstyle\sum_{j=1}^{i-1} n_{ij}\log(p_{ij}^0) + \sum_{j=i+1}^{6} n_{ij}\log(p_{ij}^0) \end{aligned}$$

where $LL_1 = n_{ii}\log(p_{ii}) + n_1\log(p_1) + n_2\log(p_2)$. By Lemma 3.1, LL_1 is maximized at $p_{ii} = r_{ii}, p_1 = r_{i1} + r_{i2} + \cdots + r_{i\,i-1}$, and $p_2 = r_{i\,i+1} + r_{i\,i+2} + \cdots + r_{i\,6}$. Applying Algorithm 4.1 respectively to $\sum_{j=1}^{i-1} n_{ij}\log(p_{ij}^0)$ for the left-hand-side off the diagonal in the row, and to $\sum_{j=i+1}^{6} n_{ij}\log(p_{ij}^0)$ for the right-hand-side off the diagonal, we get the maximum likelihood estimates for p_{ij}^0 subject to (5.2) or (5.1), hence the maximum likelihood estimates p_{ij} for all j for a fixed i, subject to (5.1) or (5.2).

Take, for example, the right-hand-side of the diagonal for the first row in the matrix, before smoothing, these numbers are:

$$0.0183, 0.0031, 0.0055, 0.0010, 0.0002. \qquad (5.3)$$

We can think these numbers are the sample multinomial percentages by dividing into each the sum of these 5 numbers, then applying Algorithm 4.1 to obtain the smoothed rates, and finally times back the sum of the above 5 numbers. Or equivalently, apply Algorithm 4.1 directly without normalization to (5.3). This means, the smoothed results are given by replacing the values for the 2nd and the 3rd numbers by their average on 0.0031 and 0.0055, while keeping others unchanged. Table 3 shows the migration matrix after smoothing.

TABLE 2. LONG-RUN TRANSITION MATRIX BEFORE SMOOTHING

	Transition probability before smoothing					
Rating	1	2	3	4	5	6
1	0.9716	0.0183	0.0031	0.0055	0.0010	0.0002
2	0.0062	0.9453	0.0307	0.0128	0.0021	0.0026
3	0.0007	0.0103	0.9380	0.0409	0.0066	0.0028
4	0.0002	0.0007	0.0126	0.9673	0.0126	0.0054
5	0.0004	0.0012	0.0079	0.0800	0.8272	0.0705
6	0.0002	0.0013	0.0027	0.0450	0.0120	0.8994

TABLE 3. LONG-RUN TRANSITION MATRIX AFTER SMOOTHING

	Transition probability after smoothing					
Rating	1	2	3	4	5	6
1	0.9718	0.0183	0.0043	0.0043	0.0010	0.0002
2	0.0062	0.9455	0.0307	0.0128	0.0024	0.0024
3	0.0007	0.0103	0.9387	0.0409	0.0066	0.0028
4	0.0002	0.0007	0.0126	0.9684	0.0126	0.0054
5	0.0004	0.0012	0.0080	0.0810	0.8380	0.0714
6	0.0002	0.0014	0.0028	0.0296	0.0296	0.9363

VI. CONCLUSIONS AND FUTURE WORKS

With the proposed non-parametric algorithm, the exact solution to the monotonic estimation of the risk scales for multivariate outcomes becomes easier. No calculation for the optimization gradients and Hessian matrices, only a machine learning data driven process is required.

One of the interesting future research subjects is the monotonic estimation for the survival probability of a loan over a risk rated portfolio: a loan with lower risk rating is expected to survive more likely. We will propose models and algorithms for the monotonic estimation of these survival probabilities.

ACKNOWLEDGMENT

The author thanks Carlos Lopez for his consistent inputs, insights, and supports for this research.

Special thanks go to Clovis Sukam, Biao Wu, Wallace Law, and Glenn Fei for many valuable conversations and their critical reading for this manuscript.

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 at: h_y02@yahoo.ca.

APPENDIX

Given two discrete probability distributions $p = \{p_i\}_{i=1}^{k}$ and $q = \{q_i\}_{i=1}^{k}$, the Kullback-Leibler (KL) divergence (also called relative entropy) between q and p is defined as

$$D_{KL}(q||p) = \sum_{i=1}^{k} q_i \log(q_i/p_i) \qquad \text{(A-1)}$$

For a fixed q, $D_{KL}(q||p)$ measures the dissimilarity between q and p ([7]). The cross-entropy $H(q,p)$ is defined as

$$H(q,p) = -\sum_{i=1}^{k} q_i \log(p_i) \qquad \text{(A-2)}$$

Hence, we have

$$H(q,p) = H(q) + D_{KL}(q||p)$$

where $H(q) = -\sum_{i=1}^{k} q_i \log(q_i)$, the entropy for distribution q. When q is fixed and given, cross-entropy $H(q,p)$ is the same as $D_{KL}(q||p)(= \sum_{i=1}^{k} q_i \log(q_i) - \sum_{i=1}^{k} q_i \log(p_i)\,)$ as a function of p, up to an additive constant (because q is fixed). Both take on minimal values when $p = q$, which is 0 for KL divergence and $H(q)$ for the cross-entropy. Thus cross-entropy measures the dissimilarity between the given distribution q and the distribution p ([5], [9]).

REFERENCES

[1] Barlow, R. E.; Bartholomew, D. J.; Bremner, J. M.; Brunk, H. D. (1972). Statistical inference under order restrictions; the theory and application of isotonic regression. New York: Wiley. ISBN 0-471-04970-0.

[2] Best, M.J.; Chakravarti N. (1990). Active set algorithms for isotonic regression; a unifying framework. Mathematical Programming. 47: 425–439. doi:10.1007/BF0158087

[3] Friedman, J. and Tibshirani, R. (1984). The Monotone Smoothing of Scatterplots. Technometrics, Vol. 26, No. 3, pp. 243-250. DOI: 10.2307/1267550

[4] Leeuw, J. D, Hornik, K. and Mair, P. (2009). Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods, Journal of statistical software 32(5). DOI: 10.18637/jssw.v032.i05

[5] Yang, B. H. (2018). Smoothing Algorithms by Constrained Maximum Likelihood, Journal of Risk Model Validation, Volume 12 (2), pp. 89-102.

[6] Potharst, R. and Feelders, A. J. (2002). Classification Trees for Problems with Monotonicity Constraints, SIGKDD Explorations, Vol. 14 (1): 1-10, 2002

[7] Kotlowski, W. and Slowinski, R. (2009). Rule Learning with Monotonicity Constraints, Proceedings of the 26th Annual International Conference on Machine Learning, pp. 537-544, 2009

[8] Eichenberg, T. (2018). Supervised Weight of Evidence Binning of Numeric Variables and Factors, R-Package Woebinning.

[9] You, S.; Ding, D.; Canini, K.; Pfeifer, J. and Gupta, M. (2017). Deep Lattice Networks and Partial Monotonic Functions, 31st Conference on Neural Information Processing System (NIPS), 2017

[10] Kullback, S.; Leibler, R.A. (1951). On information and sufficiency. Annals of Mathematical Statistics. 22 (1): 79–86. doi:10.1214/aoms/1177729694.

[11] Goodfellow, I..; Bengio, Y.; Courville, A. (2016). Deep Learning. MIT Press.

[12] Murphy, K. (2012). Machine Learning: A Probabilistic Perspective. MIT. ISBN 978-0262018029.

[13] SAS Institute Inc (2014). SAS/STAT(R) 13.2 User's Guide

[14] Robertson, T.; Wright, F. T.; Dykstra, R. L. (1998). Order Restricted Statistical Inference, John Wiley & Son.

[15] Nocedal, J., and Wright, S. J. (2006). Numerical Optimization, 2nd edn. Springer

[16] Nelson, C. R, Siegel, A. F. (1987), Parsimonious Modeling of Yield Curves, Journal of Business, Vol. 60, N0. 4, 473-489