Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Computational Statistics & Data Analysis 51 (2007) 2803–2812
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

A note on iterative marginal optimization: a simple algorithm for maximum rank correlation estimation

Hansheng Wang

Guanghua School of Management, Peking University, China

Received 22 February 2006; received in revised form 14 June 2006; accepted 1 October 2006
Available online 2 November 2006

Abstract

The *maximum rank correlation* (MRC) estimator was originally studied by Han [1987. Nonparametric analysis of a generalized regression model. J. Econometrics 35, 303–316] and Sherman [1993. The limiting distribution of the maximum rank correlation estimator. Econometrica 61, 123–137] from the econometrics point of view, and most recently attracted much attention from the classification literature due to its close relationship with the *receiver operating characteristics* (ROC) curve [Baker, 2003. The central role of receiver operating characteristics (ROC) curves in evaluating tests for the early detection of cancer. J. Nat. Cancer Inst. 95, 511–515; Pepe, 2003. The Statistical Evaluation of Medical Tests for Classification and Prediction. Oxford University Press, Oxford; Pepe et al., 2004. Combining predictors for classification using the area under the ROC curve. University of Washington Biostatistics Working Paper Series]. Compared with its nice theoretical properties and successful applications, the MRC estimator's computational aspects are not trivial. This is because the MRC objective function is neither smooth nor continuous. Therefore, the traditional Newton–Raphson type algorithm cannot be used to find the MRC estimator. As an easy solution, we propose in this article a very simple fitting algorithm named *iterative marginal optimization* (IMO), which guarantees a monotone increasing of the MRC objective function at each iteration step in a very efficient manner. We show via extensive simulation that the proposed IMO algorithm is not only computationally stable but also reasonably fast. Moreover, real data about the China stock market are analyzed to further illustrate the usefulness of the proposed.

Keywords: China stock market; Iterative marginal optimization; Maximum rank correlation; Nelder–Mead algorithm; Receiver operating characteristics; Special treatment policy

1. Introduction

The *maximum rank correlation* (MRC) estimator was first proposed by Han (1987) and further studied by Sherman (1993). It is an intuitive yet effective estimation method for a broad class of *generalized regression models* (GRMs) (Han, 1987). As shown by Sherman (1993), the MRC estimator is $\sqrt{n}$-consistent and asymptotically normal under appropriate conditions. Most recently, the MRC estimator also attracted much attention from the classification literature, which is

E-mail address: hansheng@gsm.pku.edu.cn.

0167-9473/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2006.10.004

mainly due to its close relationship with the *receiver operating characteristics* (ROC) curve (Baker, 2003; Pepe, 2003; Pepe et al., 2004). Nevertheless, compared with its nice theoretical properties, the MRC estimator's computational aspects are not trivial. Specifically, the MRC objective function is neither smooth nor continuous. Therefore, the most typically used Newton–Raphson type algorithm cannot be used to find the MRC estimator, which to some extend limited its wide application.

As an easy solution, we propose in this article a very simple yet effective algorithm named *iterative marginal optimization* (IMO). At each iteration step, IMO updates each component of the parameter vector marginally but in a very efficient manner. This makes IMO different from most other optimization algorithms (e.g., Newton–Raphson algorithm, Nelder–Mead (NM) algorithm, etc.), where each component of the parameter vector is modified jointly. By definition, IMO guarantees a monotone increasing of the MRC objective function. Hence, it is not surprising to find that IMO is computationally very stable yet reasonably fast.

The rest of the article is organized as follows. Next section introduces the MRC estimator. Then IMO is formally described in Section 3. Extensive simulation results are reported in Section 4 and a real example is presented in Section 5 for illustration purpose. Finally, the article is concluded by a brief discussion in Section 6.

2. The MRC estimator

Consider a total of n independent and identically distributed random variables (y_i, x_i, e_i), $i=1, \ldots, n$, where $y_i \in \mathscr{R}^1$ is the response of interest, $x_i=(x_{i1}, \ldots, x_{id})^\top \in \mathscr{R}^d$ is the associated d-dimensional explanatory variables, and $e_i \in \mathscr{R}^1$ is the independent random noise. In order to model the relationship between y_i and (x_i, e_i), Han (1987) introduces the following *generalized regression model* (GRM):

$$y_i = D \circ F(x_i^\top \beta, e_i),$$

where $F(\cdot, \cdot)$ is a bivariate function strictly increasing in each of its argument, $D(\cdot)$ is a univariate increasing function, and $\beta=(\beta_1, \ldots, \beta_d)^\top \in \mathscr{R}^d$ is the associated d-dimensional coefficient vector with the true value given by β_0. For the identifiability purpose, it is usually assumed that $\|\beta_0\|=1$, where $\|\cdot\|$ stands for the usual L_2 norm. Note that GRM is indeed a very general model family, which contains most typically encountered parametric models as special cases. For example, the linear regression model, the generalized linear regression model (McCullagh and Nelder, 1989), Cox's regression model (Cox, 1972), and many others. For a detailed discussion, one can refer to Han (1987) and Sherman (1993).

Due to the monotonicity of $D \circ F$ and the independence of e_i and x_i, it can be shown that (Han, 1987; Sherman, 1993)

$$P(y_i \geqslant y_j | x_i, x_j) \geqslant P(y_i \leqslant y_j | x_i, x_j) \quad \text{whenever } x_i^\top \beta_0 \geqslant x_j^\top \beta_0,$$

which motivated the definition of the following MRC

$$Q(\beta) = \frac{1}{n(n-1)} \sum_{i \neq j} I\{y_i > y_j\} \times I\{x_i^\top \beta > x_j^\top \beta\}, \tag{2.1}$$

where $I\{\cdot\}$ stands for the indicator function. Then, the MRC estimator $\hat{\beta}$ can be obtained by maximize the MRC objective function $Q(\beta)$. If y_i is a binary response taking values in $\{0, 1\}$ (as typically encountered in the classification literature), then we can define n_0 and n_1 as the number of subjects with the response value given by 0 and 1, respectively. Then, up to a constant unrelated to β, the MRC objective function is proportional to

$$\widehat{\text{AUC}} = \frac{1}{n_0 n_1} \sum_{\{i: y_i=1\}} \sum_{\{j: y_j=0\}} I\{x_i^\top \beta > x_j^\top \beta\},$$

which happens to be an estimate for the area under the ROC curve (AUC) and plays a critical role for classifier evaluation (Baker, 2003). Hence, it has attracted much attention most recently from the classification literature (Pepe, 2003; Pepe et al., 2004).

As shown by Han (1987) and Sherman (1993), the MRC estimator $\hat{\beta}$ can be $\sqrt{n}$-consistent and asymptotically normal under mild regularity conditions without specifying the form of $F(\cdot,\cdot)$ and $D(\cdot)$. Nevertheless, how to find the MRC estimator $\hat{\beta}$ computationally is indeed not trivial. Note that the MRC objective function $Q(\beta)$ is a stepwise constant function in a d-dimensional Euclidean space, which is neither smooth nor continuous. Therefore, the most typically used Newton–Raphson type algorithm cannot be used to find the estimator $\hat{\beta}$, which makes the computational aspects of the MRC estimation challenging, and to some extend limited its wide application.

In the past literature, Cavanagh and Sherman (1998) suggested the use of the NM simplex algorithm (Nelder and Mead, 1965) for the MRC maximization. Simply speaking, the NM algorithm finds the maximizer of the MRC objective function by iteratively and sequentially update each vertex of a *simplex*, which is a geometric figure consisting a total of $(d+1)$ vertices in a d-dimensional Euclidean space. By comparing the value of the MRC objective function at each vertex, the current *simplex* can be appropriately updated to be the next one, by the action of *reflection*, *expansion*, *contraction*, etc. Then, iterating such an updating process till all the vertices of the simplex converges to the same limiting point. Then, the final NM estimate is obtained. A more detailed discussion about the NM algorithm exceeds the scope of this simple note. We refer to Lagarias et al. (1998) or simply the online numerical recipe at "http://library.lanl.gov/numerical/" for a more detailed discussion.

The major advantage of the NM algorithm lies in its generality. It is directly applicable not only for MRC problem but also for many others, as long as their function values can be easily evaluated. Nevertheless, its major advantage is also its weakness. Specifically, because the NM algorithm was not specifically designed for the MRC estimation. Hence, it is not surprising to find that its numerical performance could be inferior as compared with the proposed IMO algorithm for both computational speed and accuracy.

Last, it is worthwhile to point out that both the NM algorithm and the proposed IMO algorithm share one common limitation. That is there exists no solid theory, which can guarantee the maximizer identified by either of them is indeed the global maximizer. In fact, they are very likely to be just a local maximizer (Lagarias et al., 1998). Nevertheless, our limited numerical experience in together with some from the past literature (Cavanagh and Sherman, 1998) seem to suggest that those local maximizers can still be practically very useful.

3. The IMO algorithm

3.1. The Naïve IMO algorithm

For an easy computation of the MRC estimator, we propose in this article a very simple yet effective algorithm named IMO. Specifically, IMO transfers the original d-dimensional joint optimization problem (2.1) into a number of marginal univariate optimization problems, for which the optimizer can be easily identified in a very efficient manner. Hence, we refer to it as the IMO algorithm.

More precisely, we start with an arbitrary initial value $\beta^{(0)}=(\beta_1^{(0)},\ldots,\beta_d^{(0)})$, which can simply be the naïve least square estimator. Then, let $\beta^{(m)}=(\beta_1^{(m)},\ldots,\beta_d^{(m)})$ be the estimator obtained after the mth iteration. Next, we consider how to obtain $\beta^{(m+1)}$. Instead of updating each component of $\beta^{(m)}$ simultaneously, IMO updates $\beta_{k'}^{(m)}$, $k'=1,\ldots,d$ one by one. For the purpose of convenience, we further define

$$\beta^{(m,k)}=(\beta_1^{(m+1)},\beta_2^{(m+1)},\ldots,\beta_{k-1}^{(m+1)},\beta_k^{(m)},\ldots,\beta_d^{(m)})^\top,$$

where $1\leqslant k\leqslant d$. Note that $\beta^{(m,k)}$'s first $(k-1)$ components have been updated to $\beta_{k'}^{(m+1)}$, $k'\leqslant k-1$ while the rest $(d-k+1)$ are still waiting to be modified. Then, based on $\beta^{(m,k)}$ we consider how to obtain $\beta^{(m,k+1)}$, i.e., how to obtain $\beta_k^{(m+1)}$. For convenience purpose, we refer to such an operation as the mth iteration's kth *grid search step*, or simply *step*.

Note that for a fixed $\beta_{k'}^{(m+1)}$, $k' \leqslant k-1$ and $\beta_{k'}^{(m)}$, $k' \geqslant k+1$, the value of $\beta_k^{(m+1)}$ can be obtained by maximizing the following univariate objective function:

$$
\begin{aligned}
Q^*(\beta_k^{(m+1)}) &= \frac{1}{n(n-1)} \sum_{(i,j)\in\mathscr{S}} I\left\{ x_{ik}\beta_k^{(m+1)} + \sum_{k'=1}^{k-1} x_{ik'}\beta_{k'}^{(m+1)} + \sum_{k'=k+1}^{d} x_{ik'}\beta_{k'}^{(m)} \right. \\
&\qquad \left. > x_{jk}\beta_k^{(m+1)} + \sum_{k'=1}^{k-1} x_{jk'}\beta_{k'}^{(m+1)} + \sum_{k'=k+1}^{d} x_{jk'}\beta_{k'}^{(m)} \right\} \\
&= \frac{1}{n(n-1)} \sum_{(i,j)\in\mathscr{S}} I\{a_{ij,k}\beta_k^{(m+1)} > b_{ij,k}^{(m)}\},
\end{aligned}
\tag{3.1}
$$

where $\mathscr{S} = \{(i,j) : y_i > y_j\}$, $a_{ij,k} = (x_{ik} - x_{jk})$, and

$$
b_{ij,k}^{(m)} = \sum_{k'=1}^{k-1} (x_{jk'} - x_{ik'})\beta_{k'}^{(m+1)} + \sum_{k'=k+1}^{d} (x_{jk'} - x_{ik'})\beta_{k'}^{(m)}.
$$

Note that the definition of Q^* in (3.1) should depend on the value of m and k. With some abuse of the notation, the superscript/subscript associated with (m,k) is omitted. Furthermore, note that the objective function Q^* is very easy to optimize because it is just a piecewise constant function with cut-off values given by

$$
\mathscr{C}_{m,k} = \{c_{ij,k}^{(m)} = b_{ij,k}^{(m)}/a_{ij,k}, (i,j) \in \mathscr{S}\}.
$$

Consequently, the maximizer of (3.1) must be located at one of the cut-off points. Note that theoretically, there may exist infinite many maximizers for (3.1). For simplicity purpose, we only considered the cut-off points as defined by $\mathscr{C}_{m,k}$. Hence, a grid search over $\mathscr{C}_{m,k}$ produces $\beta_k^{(m+1)}$. In the situation where the value of $a_{ij,k}$ is very close to 0, we replace it by a number with the same sign but very small absolute value (e.g., 10^{-12}).

Repeat the above *step* for each k produces $\beta^{(m+1)}$. Iterate the algorithm till $1-\beta^{(m)\top}\beta^{(m+1)}$ converges to a sufficiently small positive number (e.g., 10^{-6}). Note that $\beta^{(m)}$ should be standardized in each iteration so that $\|\beta^{(m)}\| = 1$. By the time of the converge, the MRC estimator $\hat{\beta}$ is obtained. As it can be seen, the value of $Q(\beta^{(m)})$ is guaranteed to increase in both iteration (m) and *grid search step* (k), hence, it is not surprising that IMO algorithm is computationally very stable. We refer to such an algorithm as the naïve IMO algorithm.

3.2. The improved IMO algorithm

Unfortunately, the naïve IMO algorithm is not computationally efficient. For each *grid search step*, a grid search over $\mathscr{C}_{m,k}$ is needed, where number of the grid points $n^* = |\mathscr{C}_{m,k}|$ is of the order $O(n^2)$, where $|\mathscr{C}_{m,k}|$ denotes the number of the elements contained in $\mathscr{C}_{m,k}$. Moreover, for each cut-off value in $\mathscr{C}_{m,k}$ the objective function (3.1) has to be evaluated, which call for another $O(|\mathscr{C}_{m,k}|) = O(n^2)$ operations. Then, each *grid search step* demands a total of $O(n^4)$, which is a computational effort substantially heavier than what can be practically acceptable.

In order to perform an efficient grid search, we need to first sort the value of $\mathscr{C}_{m,k}$ in an ascending order, which is denoted by $c_{(1)}^{(m)} < c_{(2)}^{(m)} < \cdots < c_{(n^*)}^{(m)}$ together with the sorted $\{a_{ij,k}\}$ values given by $a_{(1)}, \ldots, a_{(n^*)}$ (note the subscript associated with k is omitted for simplicity purpose). Such an operation can cost at most $O(n^* \log(n^*)) = O(n^2 \log(n))$ operations (Sedgewick, 1978), which is a complexity slightly larger than $O(n^2)$. We then evaluate the objective function (3.1) at $\beta_k^{(m+1)} = c_{(1)}^{(m)}$, which costs another $O(n^2) = o(n^2\log(n))$ operations. Then, the value of $Q^*(c_{(l+1)}^{m})$ can be obtained easily by an appropriate modification of $Q^*(c_{(l)}^{(m)})$ according to the following relationship:

$$
\begin{aligned}
Q^*(c_{(l+1)}^{(m)}) - Q^*(c_{(l)}^{(m)}) &= +1 \quad \text{if } a_{(l)} > 0, \\
Q^*(c_{(l+1)}^{(m)}) - Q^*(c_{(l)}^{(m)}) &= -1 \quad \text{if } a_{(l)} < 0
\end{aligned}
$$

for $1 \leqslant l \leqslant n^* - 1$, which implies that with the help of $Q^*(c_{(l)}^{(m)})$, the value of $Q^*(c_{(l+1)}^{(m)})$ can be easily obtained at the cost of only O(1). Moreover, note that the total number of cut-off values is of the order $|\mathscr{C}_{m,k}| = \mathrm{O}(n^2)$, then once the ordered cut-off values $\{c_{(l)}^{(m)}\}$ are given, it costs only $\mathrm{O}(n^2)$ operations to accomplish one *grid search step*. Hence, overall speaking, a total of $\mathrm{O}(n^2 \log(n))$ is needed for each *grid search step*.

4. The simulation study

Extensive simulation studies are carried out in this section to evaluate the finite sample performance of the proposed IMO algorithm. Four different regression models are used to simulate the data. They are given by

$$y_i = x_i^\top \beta_0 + \sigma_e \varepsilon_i, \tag{4.1}$$
$$y_i = x_i^\top \beta_0 + \exp(x_i^\top \beta_0) + \sigma_e \varepsilon_i, \tag{4.2}$$
$$y_i = \max\{x_i^\top \beta_0 + \sigma_e \varepsilon_i, 0\}, \tag{4.3}$$
$$y_i = I\{x_i^\top \beta_0 + \sigma_e \varepsilon_i > 0\}, \tag{4.4}$$

where ε_i is a standard normal random variable, and $x_i = (x_{i1}, \ldots, x_{i8})^\top \in \mathscr{R}^8$ is a eight-dimensional multivariate normal random variable with 0 mean and unit variance. Furthermore, x_i is simulated so that the pairwise correlation coefficient between x_{ij_1} and x_{ij_2} is given by $0.5^{|j_1 - j_2|}$ for any $1 \leqslant j_1, j_2 \leqslant 8$. The regression coefficient is given by $\beta_0 = (2.5, 0, \sqrt{3}, \frac{7}{3}, 0, 0, \sqrt{5}, 0)^\top$. Note that (4.1) is just a usual linear regression model and (4.2) is its nonlinear version. On the other hand, model (4.3) stands for a commonly encountered censored regression model (Han, 1987; Sherman, 1993) while (4.4) represents a binary choice model. All those models are special cases of the GRM family (Han, 1987; Sherman, 1993).

The sample size evaluated ranges from small ($n = 25$) to relatively large ($n = 500$), and the standard deviation σ_e is fixed to be 2.0, 1.0, or 0.5, which represents, respectively, a large, moderate, and small noise level. All the simulation studies are carried out in MatLab and a total of 100 simulation iterations are carried out for each model and each parameter setting. For each simulation iteration, the IMO algorithm is used to find the MRC estimator $\hat{\beta}_{imo}$. For comparison purpose, the NM simplex algorithm (Nelder and Mead, 1965; Cavanagh and Sherman, 1998), which is implemented by the function *fminsearch* in MatLab, is used to find the MRC estimator. For the sake of convenience, we denote such an estimator as $\hat{\beta}_{nm}$.

For a given estimator $\hat{\beta}_\tau$ ($\tau \in \{imo, nm\}$), its bias and asymptotic variance are estimated. Specifically, let $\hat{\beta}_\tau^{[k]} = (\hat{\beta}_{\tau,1}^{[k]}, \ldots, \hat{\beta}_{\tau,d}^{[k]})$ be the estimate obtained in the kth simulation iteration based on the method τ. Then, the bias and the variability of $\hat{\beta}_\tau$ are estimated by

$$\widehat{\text{Bias}} = \frac{1}{100} \sum_{k=1}^{100} \left\{ \frac{1}{d} \sum_{j=1}^{d} (\hat{\beta}_{\tau,j}^{[k]} - \beta_{0j}) \right\},$$

$$\widehat{\text{SD}} = \frac{1}{100} \sum_{k=1}^{100} \left\{ \frac{1}{d} \sum_{j=1}^{d} (\hat{\beta}_{\tau,j}^{[k]} - \beta_{0j})^2 \right\}^{1/2},$$

which are reported in Tables 2–5. In order to evaluate the relative computational efficiency, the average CPU time used by each method was also computed.

As one can see, in terms of the accuracy, both $\hat{\beta}_{imo}$ and $\hat{\beta}_{nm}$ are very comparable. The only difference is that the NM algorithm demonstrated a slightly better performance for the linear model (4.1). However, for all the other three nonlinear models, IMO consistently demonstrates to be a better choice for both the bias and the variability. This happens to be the situation where the MRC estimator is most useful. Moreover, for computational efficiency, the CPU time needed by IMO is consistently and substantially smaller than what is demanded by NM. Furthermore, such a computational superiority is most appreciable if the sample size is relatively large (e.g., $n = 500$). Hence, overall speaking it seems that for a general nonlinear model, the IMO algorithm can indeed provide a more accurate estimator than the NM algorithm, however, at a substantially smaller computational cost.

5. A real example

In this section, we use a real data set from China's stock market to illustrate the usefulness of the proposed IMO algorithm together with the MRC estimator itself. Since its re-opening in 1990, the Chinese stock market has experienced a rapid growth from a few stocks in 1990 to more than 1300 today, with a total capitalization of more than 450 billion US dollars. Consequently, it has attracted much attention from both investors and scholars.

In the short history of the stock market, China Security Regulation Commission (CSRC), the government body overseeing the stock market, has developed a very unique special treatment (ST) policy with the intention to protect investors. Specifically, according to the ST policy, if a company reported accounting loss for two consecutive years, its stock will be specially treated. A specially treated stock faces a number of constraints. For example, the stock price's daily movement is forced to be within the limit of $[-5\%, +5\%]$. Furthermore, if a ST firm continues to lose money in its next accounting year, it faces the risk of being delisted from the market, which certainly is a severe punishment to both management and investors. Consequently, it is of great interest to predict the firm's possibility of being specially treated in the future. Due to the fact that two consecutive loss years is the most typical condition for a firm to be specially treated, it is of little interest to predict a firm's ST possibility within two years. Therefore, a common practice is to use the current available information to predict the ST possibility in three years.

The data used in this example is derived from CCER China Stock Database, which is considered to be one of the most authoritative and widely used stock market databases on the Chinese stock market. The data set contains a total of 1522 records with each record corresponding to one yearly observation of one firm. Among them, 726 are from the year 2002, which serve as our training sample. Then, the rest data are from the year 2003, which serve as the testing data to evaluate the prediction accuracy of various estimators. Within the training sample, there are a total of 38 ST stocks, which accounts for about 5.23% of the training data. On the other hand, there are a total of 49 ST firms in the testing sample, which accounts for about 6.16% of the testing data.

For each observation, one binary variable is created to indicate the firm's ST status (1=ST; 0=none ST), and a total of seven accounting variables are collected, which are used to predict the firm's ST status. Those seven predictive variables are, respectively, account receivable to asset ratio (ARA), the logarithm of total asset (ASSET), asset turnover ratio (ATO), sales growth rate (GROWTH), debt to asset ratio or leverage (LEV), return on asset (ROA), and the percentage of outstanding shares owned by the largest shareholder (SHARE). A detailed discussion about the economics meaning of those accounting variables exceeds the scope of this article. Interested readers are referred to Harrison and Horngren (2001) or any other standard accounting textbook. Simply speaking, past empirical research seems to suggest those variables are very likely to be related to a firm's financial health, hence, to its future ST status.

For the purpose of comparison, the following four different estimates are computed. They are, namely, the MRC estimator computed by the IMO algorithm (MRC-IMO), the MRC estimator computed by the NM algorithm (MRC-NM), the logistic regression estimator (LOGIT), and the probit regression estimator (PROBIT). They are all standardized so that unit length is assured. All the estimates are computed using the training data only, and their values are reported

Table 1
Estimation results of the ST data set

Variables	MRC-IMO	MRC-NM	LOGIT	PROBIT
ARA	0.4939	0.8592	0.8455	0.8909
ASSET	0.0106	0.0001	0.0395	0.0434
ATO	−0.0521	−0.0939	−0.1143	−0.1042
GROWTH	−0.0464	−0.0015	−0.0605	−0.0737
LEV	0.2013	0.4541	0.4318	0.4330
ROA	−0.8430	−0.2160	−0.2835	0.0252
SHARE	−0.0004	−0.0012	−0.0025	−0.0022

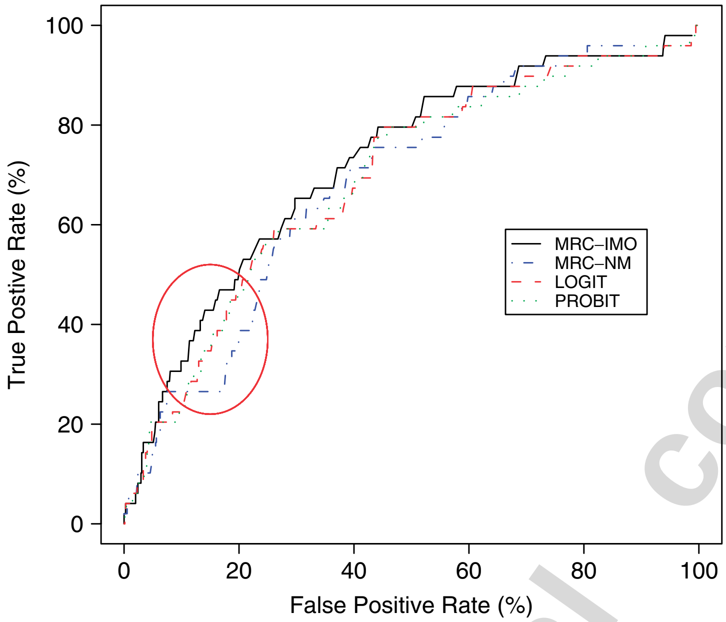

Fig. 1. The out-sample ROC curve of different estimators.

Table 2
Simulation results—model 1

σ_e	n	IMO		NM		CPU time	
		$\widehat{\text{Bias}}$	$\widehat{\text{SD}}$	$\widehat{\text{Bias}}$	$\widehat{\text{SD}}$	IMO	NM
2.0	25	−0.13	0.35	−0.12	0.34	0.02	0.22
	50	−0.08	0.29	−0.08	0.27	0.07	0.69
	100	−0.05	0.21	−0.04	0.20	0.32	3.57
	200	−0.03	0.17	−0.03	0.16	1.35	15.30
	500	−0.01	0.11	−0.01	0.10	9.06	113.20
1.0	25	−0.06	0.25	−0.05	0.23	0.02	0.22
	50	−0.03	0.17	−0.03	0.16	0.07	0.68
	100	−0.01	0.13	−0.01	0.12	0.30	3.55
	200	0.00	0.09	0.00	0.08	1.19	14.96
	500	0.00	0.06	0.00	0.06	8.84	109.18
0.5	25	−0.03	0.16	−0.02	0.14	0.02	0.22
	50	−0.01	0.10	−0.01	0.09	0.07	0.66
	100	−0.01	0.07	0.00	0.06	0.28	3.48
	200	0.00	0.05	0.00	0.04	1.18	14.44
	500	0.00	0.03	0.00	0.03	8.26	104.96

in Table 1. As it can be seen, all those four estimates are very similar, in the sense they share the same signs in most situations. However, in terms of the magnitude of the values, they are all very different. For comparison purpose, we computed the MRC values for both MRC-IMO and MRC-NM estimator, which are given by 0.0393 and 0.0371, respectively. Consequently, in terms of the MRC maximization, IMO finds a better estimator than NM. Furthermore, in terms of the computational efficiency, IMO only demands a total of 2.313 CPU time units while NM needs 188.875, which represents a much slower computational speed.

In order to further compare the relative superiority of the above four estimates, the testing data is used to evaluate their prediction accuracy. Specifically, for a given estimator $\hat{\beta}$ and a testing data $(\tilde{x}_i, \tilde{y}_i)$, we predict the value of $\tilde{y}_i$ by $I\{\tilde{x}_i^\top \hat{\beta} > c^*\}$ for some tuning constant c^*, which controls the balance between the false-positive rate (FPR) and

Table 3
Simulation results—model 2

σ_e	n	IMO		NM		CPU time	
		$\widehat{\text{Bias}}$	$\widehat{\text{SD}}$	$\widehat{\text{Bias}}$	$\widehat{\text{SD}}$	IMO	NM
2.0	25	−0.06	0.24	−0.07	0.26	0.02	0.22
	50	−0.03	0.18	−0.03	0.19	0.08	0.70
	100	−0.01	0.12	−0.01	0.13	0.35	3.80
	200	0.00	0.08	−0.01	0.09	1.47	16.53
	500	0.00	0.05	0.00	0.06	11.43	127.42
1.0	25	−0.04	0.19	−0.05	0.23	0.02	0.22
	50	−0.02	0.12	−0.03	0.16	0.08	0.74
	100	0.00	0.08	−0.01	0.10	0.37	4.14
	200	0.00	0.05	−0.01	0.06	1.57	17.68
	500	0.00	0.03	0.00	0.04	11.48	137.13
0.5	25	−0.02	0.16	−0.05	0.20	0.02	0.23
	50	−0.01	0.09	−0.02	0.15	0.09	0.78
	100	0.00	0.05	0.00	0.08	0.39	4.66
	200	0.00	0.03	0.00	0.04	1.68	20.77
	500	0.00	0.02	0.00	0.02	12.13	153.19

Table 4
Simulation results—model 3

σ_e	n	IMO		NM		CPU time	
		$\widehat{\text{Bias}}$	$\widehat{\text{SD}}$	$\widehat{\text{Bias}}$	$\widehat{\text{SD}}$	IMO	NM
2.0	25	−0.13	0.37	−0.16	0.40	0.01	0.23
	50	−0.09	0.31	−0.12	0.34	0.06	0.71
	100	−0.05	0.24	−0.07	0.27	0.28	3.72
	200	−0.02	0.18	−0.03	0.19	1.16	16.16
	500	−0.01	0.12	−0.01	0.13	8.70	128.33
1.0	25	−0.09	0.28	−0.12	0.32	0.01	0.22
	50	−0.04	0.21	−0.05	0.24	0.06	0.70
	100	−0.02	0.16	−0.03	0.19	0.27	3.64
	200	−0.01	0.11	−0.01	0.12	1.13	16.63
	500	0.00	0.07	−0.01	0.07	8.56	126.19
0.5	25	−0.05	0.25	−0.08	0.30	0.01	0.22
	50	−0.01	0.16	−0.03	0.20	0.06	0.71
	100	−0.01	0.10	−0.02	0.13	0.28	3.90
	200	0.00	0.07	−0.01	0.08	1.18	17.53
	500	0.00	0.04	0.00	0.05	8.85	132.61

true-positive rate (TPR). For a given c^*, the FPR and TPR can be estimated by

$$\widehat{\text{FPR}} = \left\{\sum (1-\tilde{y}_i) \times I\{\tilde{x}_i^\top \hat{\beta} > c^*\}\right\} \times \left\{\sum (1-\tilde{y}_i)\right\}^{-1},$$

$$\widehat{\text{TPR}} = \left\{\sum \tilde{y}_i \times I\{\tilde{x}_i^\top \hat{\beta} > c^*\}\right\} \times \left\{\sum \tilde{y}_i\right\}^{-1}.$$

By using different c^* values, different $(\widehat{\text{FPR}}, \widehat{\text{TPR}})$ rates can be obtained, which forms the ROC curve and is displayed in Fig. 1.

As it can be seen, the MRC-IMO estimator clearly demonstrates to be the best choice compared with the other three choices. This is because the ROC curve associated with MRC-IMO is almost always on the top of the other three

Table 5
Simulation results—model 4

σ_e	n	IMO		NM		CPU time	
		$\widehat{\text{Bias}}$	$\widehat{\text{SD}}$	$\widehat{\text{Bias}}$	$\widehat{\text{SD}}$	IMO	NM
2.0	25	−0.14	0.40	−0.17	0.43	0.01	0.22
	50	−0.10	0.33	−0.11	0.35	0.04	0.68
	100	−0.06	0.28	−0.09	0.31	0.19	3.81
	200	−0.04	0.21	−0.05	0.23	0.86	17.20
	500	−0.02	0.14	−0.02	0.15	5.59	134.22
1.0	25	−0.11	0.34	−0.14	0.37	0.01	0.18
	50	−0.05	0.26	−0.08	0.31	0.04	0.68
	100	−0.02	0.19	−0.04	0.24	0.19	3.66
	200	−0.01	0.14	−0.02	0.17	0.83	16.48
	500	0.00	0.08	−0.01	0.09	5.73	130.31
0.5	25	−0.08	0.30	−0.10	0.33	0.01	0.25
	50	−0.04	0.20	−0.07	0.26	0.04	0.69
	100	−0.02	0.13	−0.03	0.18	0.19	3.74
	200	−0.01	0.10	−0.01	0.11	0.84	17.30
	500	0.00	0.06	0.00	0.06	5.74	132.22

curves, which simply indicates for the same $\widehat{\text{FPR}}$ rate, MRC-IMO usually achieves a larger $\widehat{\text{TPR}}$ value than the other three estimators. Such a superiority is most appreciable for the situation $15\% < \widehat{\text{FPR}} < 25\%$ (i.e., the circled region in Fig. 1), with a margin could be as large as about 20%. It is worthwhile to note that in such a situation the MRC-NM is clearly the worst estimator, which is mainly due to the computation inefficiency of the NM algorithm. On the other hand, the performance of LOGIT and PROBIT are always very comparable. Therefore, overall speaking MRC-IMO seems to be the best choice (see also Tables 2–5).

6. Discussion

We show in our numerical study that the MRC estimator could be quite a useful alternative for the many typically used parametric regression models. Hence, it is of great interest to further extend its application to various nonparametric and semiparametric models (Fan and Gijbels, 1996; Horowitz, 1998). For the future research along this line, we believe the proposed IMO algorithm can be a useful computational help.

Acknowledgments

We are grateful to the editor and the two referees for their valuable comments and constructive suggestions, which led to a substantial improvement of this manuscript.

References

Baker, S.G., 2003. The central role of receiver operating characteristics (roc) curves in evaluating tests for the early detection of cancer. J. Nat. Cancer Inst. 95, 511–515.
Cavanagh, C., Sherman, R.P., 1998. Rank estimators for monotonic index models. J. Econometrics 84, 351–381.
Cox, D.R., 1972. Regression models and life tables. J. Roy. Statist. Soc. Ser. B 34, 187–220.
Fan, J., Gijbels, I., 1996. Local Polynomial Modelling and Its Applications. Chapman & Hall, New York, NY.
Han, A.K., 1987. Nonparametric analysis of a generalized regression model. J. Econometrics 35, 303–316.
Harrison, W.T., Horngren, C.T., 2001. Financial Accounting. Prentice-Hall, New York, NY.
Horowitz, J.L., 1998. Semiparametric Methods in Econometrics. Springer, New York.
Lagarias, et al., 1998. Convergence properties of the Nelder–Mead simplex method in low dimensions. SIAM J. Optim. 9, 112–147.
McCullagh, P., Nelder, J.A., 1989. Generalized Linear Models. Chapman & Hall, New York, NY.
Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. Comput. J. 7, 308–313.

Pepe, M.S., 2003. The Statistical Evaluation of Medical Tests for Classification and Prediction. Oxford University Press, Oxford.
Pepe, M.S., et al., 2004. Combining predictors for classification using the area under the ROC curve. University of Washington Biostatistics Working Paper Series.
Sedgewick, R., 1978. Implementing quicksort programs. Comm. ACM 21, 847–857.
Sherman, R.P., 1993. The limiting distribution of the maximum rank correlation estimator. Econometrica 61, 123–137.