Probability, Uncertainty and Quantitative Risk (2019) 4:8
https://doi.org/10.1186/s41546-019-0042-6
Probability, Uncertainty and Quantitative Risk

RESEARCH Open Access

Nonlinear regression without i.i.d. assumption

Qing Xu · Xiaohua (Michael) Xuan

Received: 30 July 2018 / Accepted: 23 September 2019 / Published online: 05 November 2019

Abstract In this paper, we consider a class of nonlinear regression problems without the assumption of being independent and identically distributed. We propose a correspondent mini-max problem for nonlinear regression and give a numerical algorithm. Such an algorithm can be applied in regression and machine learning problems, and yields better results than traditional least squares and machine learning methods.

Keywords Nonlinear regression · Minimax · Independent · Identically distributed · Least squares · Machine learning · Quadratic programming

Abbreviations
i.i.d.: Independent and identically distributed
MAE: Mean absolute error
MSE: Mean squared error

1 Introduction

In statistics, linear regression is a linear approach for modelling the relationship between a response variable y and one or more explanatory variables denoted by x:

$$y = w^T x + b + \varepsilon. \tag{1}$$

Here, ε is a random noise. The associated noise terms $\{\varepsilon_i\}_{i=1}^m$ are assumed to be i.i.d. (independent and identically distributed) with mean 0 and variance σ^2. The parameters w, b are estimated via the method of least squares as follows.

Q. Xu and X. (Michael) Xuan (✉)
UniDT, Shanghai, China
e-mail: morrisxq@126.com

Lemma 1 *Suppose* $\{(x_i, y_i)\}_{i=1}^m$ *are drawn from the linear model* (1)*. Then the result of least squares is*

$$(w_1, w_2, \cdots, w_d, b)^T = A^+ c.$$

Here,

$$A = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} & 1 \\ x_{21} & x_{22} & \cdots & x_{2d} & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{md} & 1 \end{pmatrix}, \quad c = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{pmatrix}.$$

A^+ *is the Moore−Penrose inverse*[1] *of* A*.*

In the above lemma, $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m$ are assumed to be i.i.d. Therefore, $y_1, y_2, \cdots, y_m$ are also i.i.d.

When the i.i.d. assumption is not satisfied, the usual method of least squares does not work well. This is illustrated by the following example.

Example 1 *Denote by* $\mathcal{N}(\mu, \sigma^2)$ *the normal distribution with mean* μ *and variance* σ^2 *and denote by* δ_c *the Dirac distribution, i.e.,*

$$\delta_c(A) = \begin{cases} 1 & c \in A, \\ 0 & c \notin A. \end{cases}$$

Suppose the sample data are generated by

$$y_i = 1.75 * x_i + 1.25 + \varepsilon_i, \quad i = 1, 2, \cdots, 1517,$$

where

$$\varepsilon_1, \cdots, \varepsilon_{500} \sim \delta_{0.0325}, \quad \varepsilon_{501}, \cdots, \varepsilon_{1000} \sim \delta_{0.5525},$$

$$\varepsilon_{1001}, \cdots, \varepsilon_{1500} \sim \delta_{-0.27}, \quad \varepsilon_{1501}, \cdots, \varepsilon_{1517} \sim \mathcal{N}(0, 0.2).$$

The result of the usual least squares is

$$y = 0.4711 * x + 1.4258,$$

which is displayed in Fig. 1.

We see from Fig. 1 that most of the sample data deviates from the regression line. The main reason is that the i.i.d. condition is violated.

For overcoming the above difficulty, Lin et al. (2016) studied the linear regression without i.i.d. condition by using the nonlinear expectation framework laid out by Peng (2005). They split the training set into several groups and in each group the i.i.d. condition can be satisfied. The average loss is used for each group and the maximum of average loss among groups is used as the final loss function. They show that the

[1]For the definition and property of Moore−Penrose inverse, see (Ben-Israel and Greville 2003).

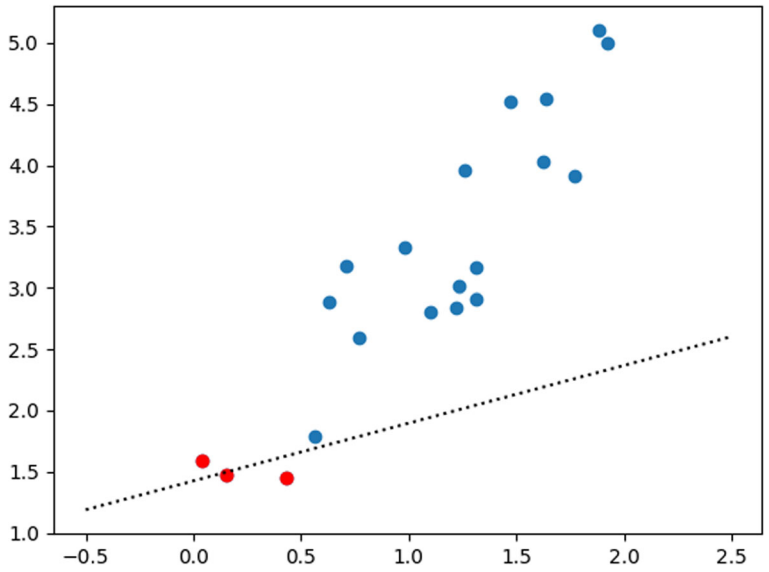

Fig. 1 Result of least squares

linear regression problem under the nonlinear expectation framework is reduced to the following mini-max problem.

$$\min_{w,b} \max_{1\le j\le N} \frac{1}{M} \sum_{l=1}^{M} \left(w^T x_{jl} + b - y_{jl} \right)^2 . \tag{2}$$

They suggest a genetic algorithm to solve this problem. However, such a genetic algorithm does not work well generally.

Motivated by the work of Lin et al. (2016) and Peng (2005), we consider nonlinear regression problems without the assumption of i.i.d. in this paper. We propose a correspondent mini-max problems and give a numerical algorithm for solving this problem. Meanwhile, problem (2) in Lin's paper can also be well solved by such an algorithm. We also have done some experiments in least squares and machine learning problems.

2 Nonlinear regression without i.i.d. assumption

Nonlinear regression is a form of regression analysis in which observational data are modeled by a nonlinear function which depends on one or more explanatory variables (see, e.g., Seber and Wild (1989)).

Suppose the sample data (training set) is

$$S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\},$$

where $x_i \in X$ and $y_i \in Y$. X is called the input space and Y is called the output (label) space. The goal of nonlinear regression is to find (learn) a function $g^\theta : X \to Y$

from the hypothesis space $\{g^\lambda : X \to Y | \lambda \in \Lambda\}$ such that $g^\theta(x_i)$ is as close to y_i as possible.

The closeness is usually characterized by a loss function φ such that $\varphi\left(g^\theta(x_1), y_1, \cdots, g^\theta(x_m), y_m\right)$ attains its minimum if and only if

$$g^\theta(x_i) - y_i = 0, \quad 1 \le i \le m.$$

Then the nonlinear regression problem (learning problem) is reduced to an optimization problem of minimizing φ.

Following are two kinds of loss functions, namely, the average loss and the maximal loss.

$$\varphi_2 = \frac{1}{m} \sum_{j=1}^{m} \left(g^\theta(x_j) - y_j\right)^2.$$

$$\varphi_\infty = \max_{1 \le j \le m} \left(g^\theta(x_j) - y_j\right)^2.$$

The average loss is popular, particularly in machine learning, since it can be conveniently minimized using online algorithms, which process fewer instances during each iteration. The idea behinds the average loss is to learn a function that performs equally well for each training point. However, when the i.i.d. assumption is not satisfied, the average loss function method may become a problem.

To overcome this difficulty, we use the max-mean as the loss function. First, we split the training set into several groups and in each group the i.i.d. condition can be satisfied. Then, the average loss is used for each group and the maximum of average loss among groups is used as the final loss function. We propose the following minimax problem for nonlinear regression problems.

$$\min_\theta \max_{1 \le j \le N} \frac{1}{n_j} \sum_{l=1}^{n_j} \left(g^\theta(x_{jl}) - y_{jl}\right)^2. \tag{3}$$

Here, n_j is the number of samples in group j.

Problem (3) is a generalization of problem (2). Next, we will give a numerical algorithm which solves problem (3).

Remark 1 *Jin and Peng (2016) put forward a max-mean method to give the parameter estimation when the usual i.i.d. condition is not satisfied. They show that if $Z_1, Z_2, \cdots, Z_k$ are drawn from the maximal distribution $M_{[\underline{\mu}, \overline{\mu}]}$ and are nonlinearly independent, then the optimal unbiased estimation for $\overline{\mu}$ is*

$$\max\{Z_1, Z_2, \cdots, Z_k\}.$$

This fact, combined with the Law of Large Numbers (Theorem 19 in Jin and Peng (2016)*) leads to the max-mean estimation of μ. We borrow this idea and use the max-mean as the loss function for the nonlinear regression problem.*

3 Algorithm

Problem (3) is a mini-max problem. The mini-max problems arise in different kinds of mathematical fields, such as game theory and the worst-case optimization. The general mini-max problem is described as

$$\min_{u\in\mathbb{R}^n}\max_{v\in V} h(u,v). \tag{4}$$

Here, h is continuous on $\mathbb{R}^n \times V$ and differentiable with respect to u.

Problem (4) was considered theoretically by Klessig and Polak (1973) in 1973 and Panin (1981) in 1981. Later in 1987, Kiwiel (1987) gave a concrete algorithm for problem (4). Kiwiel's algorithm dealt with the general case in which V is a compact subset of $\mathbb{R}^d$ and the convergence could be slow when the number of parameters is large.

In our case, $V = \{1, 2, \cdots, N\}$ is a finite set and we give a simplified and faster algorithm.

Denote

$$f_j(u) = h(u,j) = \frac{1}{n_j}\sum_{l=1}^{n_j}\left(g^u(x_{jl}) - y_{jl}\right)^2, \quad \Phi(u) = \max_{1\le j\le N} f_j(u).$$

Suppose each f_j is differentiable. Now, we outline the iterative algorithm for the following discrete mini-max problem

$$\min_{u\in\mathbb{R}^n}\max_{1\le j\le N} f_j(u).$$

The main difficulty is to find the descent direction at each iteration point $u_k (k = 0, 1, \cdots)$ since Φ is nonsmooth in general. In light of this, we linearize f_j at u_k and obtain the convex approximation of Φ as

$$\hat{\Phi}(u) = \max_{1\le j\le N}\{f_j(u_k) + \langle \nabla f_j(u_k), u - u_k\rangle\}.$$

Next, we find u_{k+1}, which minimizes $\hat{\Phi}(u)$. In general, $\hat{\Phi}$ is not strictly convex with respect to u, and thus it may not admit a minimum. Motivated by the alternating direction method of multipliers (ADMM, see, e.g., Boyd et al. (2010) and Kellogg (1969)), we add a regularization term and the minimization problem becomes

$$\min_{u\in\mathbb{R}^n}\left\{\hat{\Phi}(u) + \frac{1}{2}\|u - u_k\|^2\right\}.$$

By setting $d = u - u_k$, the above is converted to the following form

$$\min_{d\in\mathbb{R}^n}\left\{\max_{1\le j\le N}\{f_j(u_k) + \langle \nabla f_j(u_k), d\rangle\} + \frac{1}{2}\|d\|^2\right\}, \tag{5}$$

which is equivalent to

$$\min_{d,a} \quad \left(\frac{1}{2}\|d\|^2 + a\right) \tag{6}$$

$$\text{s.t. } f_j(u_k) + \langle \nabla f_j(u_k), d\rangle \le a, \ \forall\, 1 \le j \le N. \tag{7}$$

Problem (6)−(7) is a semi-definite QP (quadratic programming) problem. When n is large, the popular QP algorithms (such as the active-set method) are time-consuming. So we turn to the dual problem.

Theorem 1 *Denote $G = \nabla f \in \mathbb{R}^{N\times n}$, $f = (f_1, \cdots, f_N)^T$. If λ is the solution of the following QP problem*

$$\min_{\lambda} \left(\frac{1}{2}\lambda^T G G^T \lambda - f^T \lambda\right) \tag{8}$$

$$\text{s.t. } \sum_{i=1}^{N} \lambda_i = 1, \lambda_i \ge 0. \tag{9}$$

Then $d = -G^T\lambda$ is the solution of problem (6)−(7).

Proof See Appendix. □

Remark 2 *Problem (8)−(9) can be solved by many standard methods, such as active-set method (see, e.g.,* (Nocedal and Wright 2006)*). The dimension of the dual problem (8)−(9) is N (number of groups), which is independent of n (number of parameters). Hence, the algorithm is fast and stable, especially in deep neural networks.*

Set $d_k = -G^T\lambda$. The next theorem shows that d_k is a descent direction.

Theorem 2 *If $d_k \neq 0$, then there exists $t_0 > 0$ such that*

$$\Phi(u_k + t d_k) < \Phi(u_k), \quad \forall\, t \in (0, t_0).$$

Proof See Appendix. □

For a function F, the directional derivative of F at x in a direction d is defined as

$$F'(x; d) := \lim_{t\to 0+} \frac{F(x + td) - F(x)}{t}.$$

The necessary optimality condition for a function F to attain its minimum (see Demyanov and Malozemov (1977)) is

$$F'(x; d) \geq 0, \ \forall d \in \mathbb{R}^n.$$

x is called a stationary point of F.

Theorem 2 shows that when $d_k \neq 0$, we can always find a descent direction. The next theorem reveals that when $d_k = 0$, u_k is a stationary point.

Theorem 3 *If $d_k = 0$, then u_k is a stationary point of Φ, i.e.,*

$$\Phi'(u_k; d) \geq 0, \ \forall d \in \mathbb{R}^n.$$

Proof See Appendix. □

Remark 3 *When each f_j is a convex function, Φ is also a convex function. Then, the stationary point of Φ becomes the global minimum point.*

With d_k being the descent direction, we use line search to find the appropriate step size and update the iteration point.

Now, let us conclude the above discussion by giving the concrete steps of the algorithm for the following mini-max problem.

$$\min_{u \in \mathbb{R}^n} \max_{1 \leq j \leq N} f_j(u). \tag{10}$$

Algorithm.

Step 1. Initialization

Select arbitrary $u_0 \in \mathbb{R}^n$. Set $k = 0$, termination accuracy $\xi = 10^{-8}$, gap tolerance $\delta = 10^{-7}$, and step size factor $\sigma = 0.5$.

Step 2. Finding Descent Direction

Assume that we have chosen u_k. Compute the Jacobian matrix

$$G = \nabla f(u_k) \in \mathbb{R}^{N \times n},$$

where

$$f(u) = (f_1(u), \cdots, f_N(u))^T.$$

Solve the following quadratic programming problem with gap tolerance δ (see, e.g., Nocedal and Wright (2006)).

$$\min_{\lambda} \left(\frac{1}{2} \lambda^T G G^T \lambda - f^T \lambda \right)$$

$$\text{s.t.} \sum_{i=1}^{N} \lambda_i = 1, \lambda_i \geq 0.$$

Take $d_k = -G^T \lambda$. If $\|d_k\| < \xi$, stop. Otherwise, goto Step 3.

Step 3. Line Search
Find the smallest natural number j such that

$$\Phi\left(u_k + \sigma^j d_k\right) < \Phi(u_k).$$

Take $\alpha_k = \sigma^j$ and set $u_{k+1} = u_k + \alpha_k d_k,\ k = k + 1$. Go to Step 2.

4 Experiments

4.1 The linear regression case

Example 1 can be numerically well solved by the above algorithm with

$$f_j(w, b) = (wx_j + b - y_j)^2, \quad j = 1, 2, \cdots, 1517.$$

The corresponding optimization problem is

$$\min_{w,b} \max_{1 \le j \le 1517} (wx_j + b - y_j)^2.$$

The numerical result using the algorithm in Section 3 is

$$y = 1.7589 * x + 1.2591.$$

The result is summarized in Fig. 2. Note that the mini-max method (black line) performs better than the traditional least squares method (pink line).

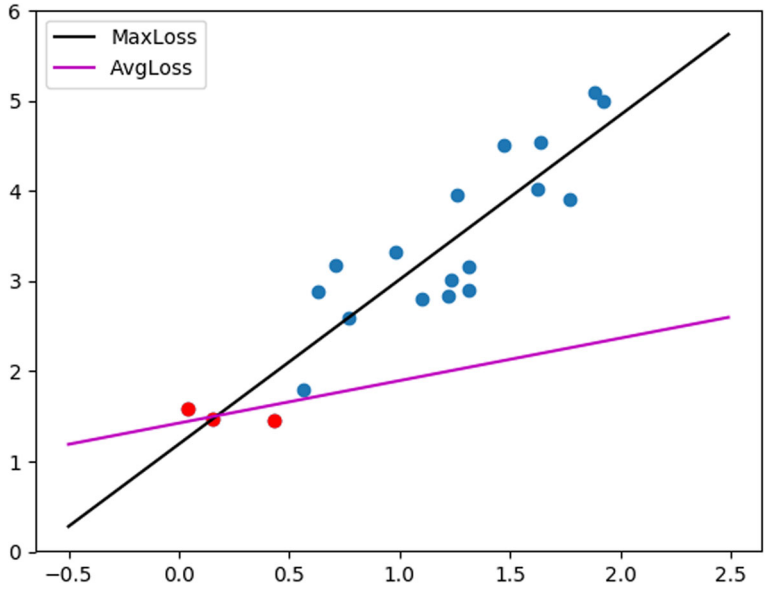

Fig. 2 Results of the two methods

Table 1 Comparisons of the two methods

Method	D_2	D_1
Traditional method	1.2789	1.2878
Mini-max method	0.1755	0.1848

Next, we compare the two methods. Both l^2 distance and l^1 distance are used as measurements.

$$D_2 := \sqrt{(w - \hat{w})^2 + (b - \hat{b})^2}.$$

$$D_1 := |w - \hat{w}| + |b - \hat{b}|.$$

We see from table 1 that mini-max method outperforms the traditional method in both l^2 and l^1 distances.

Lin et al. (2016) have mentioned that the above problem can be solved by genetic algorithms. However, the genetic algorithm is heuristic and unstable especially when the number of groups is large. In contrast, our algorithm is fast and stable and the convergence is proved.

4.2 The machine learning case

We further test the proposed method by using the CelebFaces Attributes Dataset (CelebA)[2] and implement the mini-max algorithm with a deep learning approach. The dataset CelebA has 202599 face images among which 13193 (6.5%) have eyeglass. The objective is eyeglass detection. We use a single hidden layer neural network to compare the two different methods.

We randomly choose 20000 pictures as the training set among which 5% have eyeglass labels. For the traditional method, the 20000 pictures are used as a whole. For the mini-max method, we separate the 20000 pictures into 20 groups. Only 1 group contains eyeglass pictures while the other 19 groups do not contain eyeglass pictures. In this way, the whole mini-batch is not i.i.d. while each subgroup is expected to be i.i.d.

The traditional method uses the following loss

$$\text{loss} = \frac{1}{20000} \sum_{i=1}^{20} \sum_{j=1}^{1000} (\sigma(W x_{ij} + b) - y_{ij})^2.$$

The mini-max method uses the maximal group loss

$$\text{loss} = \max_{1 \le i \le 20} \frac{1}{1000} \sum_{j=1}^{1000} (\sigma(W x_{ij} + b) - y_{ij})^2.$$

[2] see http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Here, σ is an activation function in deep learning such as the sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

We perform the two methods for 100 iterations. We see from Fig. 3 that the mini-max method converges much faster than the traditional method. Figure 4 also shows that the mini-max method performs better than the traditional method in accuracy. (Suppose the total number of the test set is n, and m of them are classified correctly. Then the accuracy is defined to be m/n.)

The average accuracy for the mini-max method is 74.52% while the traditional method is 41.78%. Thus, in the deep learning approach with a single layer, the mini-max method helps to speed up convergence on unbalanced training data and improves accuracy as well. We also expect improvement with the multi-layer deep learning approach.

5 Conclusion

In this paper, we consider a class of nonlinear regression problems without the assumption of being independent and identically distributed. We propose a correspondent mini-max problem for nonlinear regression and give a numerical algorithm. Such an algorithm can be applied in regression and machine learning problems, and yields better results than least squares and machine learning methods.

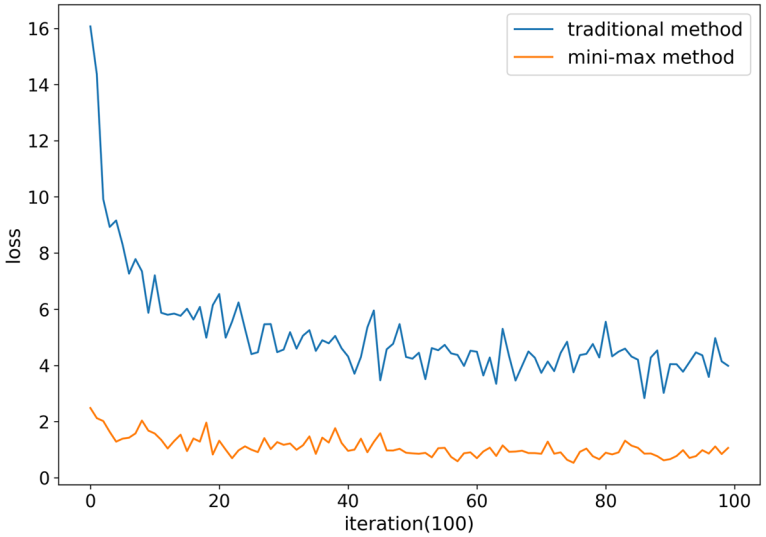

Fig. 3 Loss of the two methods

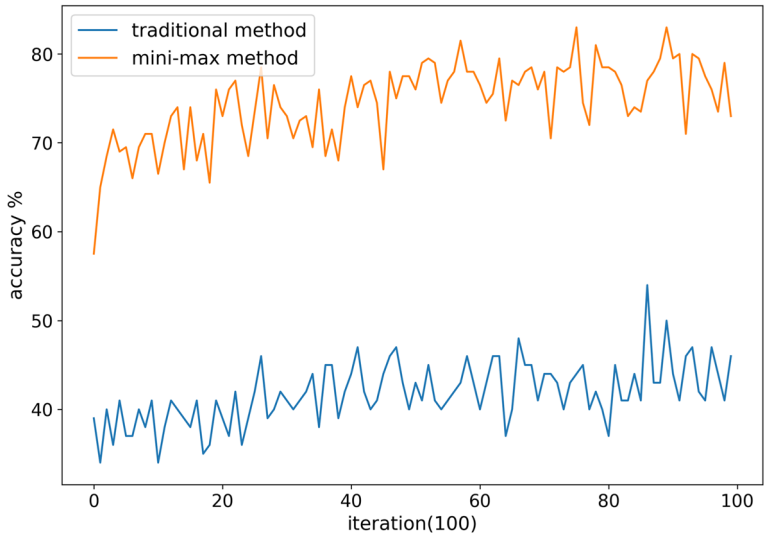

Fig. 4 Accuracy of the two methods

Appendix

Proof of Theorem 1

Consider the Lagrange function

$$L(d, a; \lambda) = \frac{1}{2}\|d\|^2 + a + \sum_{j=1}^{N} \lambda_j (f_j(u_k) + \langle \nabla f_j(u_k), d \rangle - a).$$

It is easy to verify that problem (6)−(7) is equivalent to the following minimax problem.

$$\min_{d,a} \max_{\lambda \geq 0} L(d, a; \lambda).$$

By the strong duality theorem (see, e.g., (Boyd and Vandenberghe 2004)),

$$\min_{d,a} \max_{\lambda \geq 0} L(d, a; \lambda) = \max_{\lambda \geq 0} \min_{d,a} L(d, a; \lambda).$$

Set $e = (1, 1, \cdots, 1)^T$, the above problem is equivalent to

$$\max_{\lambda \geq 0} \min_{d,a} \left(\frac{1}{2}\|d\|^2 + a + \lambda^T (f + Gd - ae) \right).$$

Note that

$$\frac{1}{2}\|d\|^2 + a + \lambda^T (f + Gd - ae) = \frac{1}{2}\|d\|^2 + \lambda^T (f + Gd) + a(1 - \lambda^T e).$$

If $1-\lambda^T e \neq 0$, then the above is $-\infty$. Thus, we must have $1-\lambda^T e = 0$ when the maximum is attained. The problem is converted to

$$\max_{\lambda_i \geq 0, \sum_{i=1}^N \lambda_i = 1} \min_d \left(\frac{1}{2}\|d\|^2 + \lambda^T G d + \lambda^T f \right).$$

The inner minimization problem has the solution $d = -G^T\lambda$ and the above problem is reduced to

$$\min_\lambda \left(\frac{1}{2}\lambda^T G G^T \lambda - f^T \lambda \right)$$

$$\text{s.t.} \sum_{i=1}^N \lambda_i = 1, \lambda_i \geq 0.$$

Proof of Theorem 2

Denote $u = u_k, d = d_k$. For $0 < t < 1$,

$$\begin{aligned}
&\Phi(u+td) - \Phi(u) \\
=& \max_{1\leq j\leq N} \{f_j(u+td) - \Phi(u)\} \\
=& \max_{1\leq j\leq N} \{f_j(u) + t\langle \nabla f_j(u), d\rangle - \Phi(u) + o(t)\} \\
\leq& \max_{1\leq j\leq N} \{f_j(u) + t\langle \nabla f_j(u), d\rangle - \Phi(u)\} + o(t) \\
=& \max_{1\leq j\leq N} \{t(f_j(u) + \langle \nabla f_j(u), d\rangle - \Phi(u)) + (1-t)(f_j(u) - \Phi(u))\} + o(t) \\
& \left(\text{Note that} f_j(u) \leq \Phi(u) = \max_{1\leq k\leq N} f_k(u) \right) \\
\leq& t \max_{1\leq j\leq N} \{f_j(u) + \langle \nabla f_j(u), d\rangle - \Phi(u)\} + o(t).
\end{aligned}$$

Since d is the solution of problem (5), we have that

$$\begin{aligned}
& \max_{1\leq j\leq N} \left\{ f_j(u) + \langle \nabla f_j(u), d\rangle + \frac{1}{2}\|d\|^2 \right\} \\
\leq& \max_{1\leq j\leq N} \left\{ f_j(u) + \langle \nabla f_j(u), 0\rangle + \frac{1}{2}\|0\|^2 \right\} \\
=& \max_{1\leq j\leq N} \{f_j(u)\} \\
=& \Phi(u).
\end{aligned}$$

Therefore,

$$\max_{1\le j\le N}\{f_j(u)+\langle\nabla f_j(u),d\rangle-\Phi(u)\}\le-\frac{1}{2}\|d\|^2.$$

$$\Rightarrow\ \Phi(u+td)-\Phi(u)\le-\frac{1}{2}t\|d\|^2+o(t).$$

$$\Rightarrow\ \frac{\Phi(u+td)-\Phi(u)}{t}\le-\frac{1}{2}\|d\|^2+o(1).$$

$$\Rightarrow\ \limsup_{t\to0+}\frac{\Phi(u+td)-\Phi(u)}{t}\le-\frac{1}{2}\|d\|^2<0.$$

For $t>0$ small enough, we have that

$$\Phi(u+td)<\Phi(u).$$

Proof of Theorem 3

Denote $u=u_k$. Then, $d_k=0$ means that $\forall d$,

$$\max_{1\le j\le N}\{f_j(u)+\langle\nabla f_j(u),d\rangle\}+\frac{1}{2}\|d\|^2\ge\max_{1\le j\le N}f_j(u).\tag{11}$$

Denote

$$M=\max_{1\le j\le N}f_j(u).$$

Define

$$\Theta=\Big\{j|f_j(u)=M,\ j=1,2,\cdots,N\Big\}.$$

Then (see Demyanov and Malozemov (1977))

$$\Phi'(u;d)=\max_{j\in\Theta}\langle\nabla f_j(u),d\rangle.\tag{12}$$

When $\|d\|$ is small enough, we have that

$$\begin{aligned}&\max_{1\le j\le N}\{f_j(u)+\langle\nabla f_j(u),d\rangle\}\\=&\max_{j\in\Theta}\{f_j(u)+\langle\nabla f_j(u),d\rangle\}\\=&M+\max_{j\in\Theta}\langle\nabla f_j(u),d\rangle.\end{aligned}$$

In view of (11), we have that for $\|d\|$ small enough,

$$\max_{j\in\Theta}\langle\nabla f_j(u),d\rangle+\frac{1}{2}\|d\|^2\ge0.$$

For any $d_1\in\mathbb{R}^n$, by taking $d=rd_1$ with sufficient small $r>0$, we have that

$$\max_{j\in\Theta}\langle\nabla f_j(u),rd_1\rangle+\frac{r^2}{2}\|d_1\|^2\ge0.$$

$$\max_{j\in\Theta}\langle\nabla f_j(u),d_1\rangle+\frac{r}{2}\|d_1\|^2\ge0.$$

Let $r \to 0+$,

$$\max_{j\in\Theta}\langle \nabla f_j(u), d_1\rangle \geq 0.$$

Thus, we fulfill the proof by combining with (12).

Acknowledgements The authors would like to thank Professor Shige Peng for useful discussions. We especially thank Xuli Shen for performing the experiment in the machine learning case.

Authors' contributions

MX puts forward the main idea and the algorithm. QX proves the convergence of the algorithm and collects the results. Both authors read and approved the final manuscript.

Funding

This paper is partially supported by Smale Institute.

Availability of data and materials

Please contact author for data requests.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

References

Ben-Israel, A. and T.N.E. Greville. (2003). *Generalized inverses: Theory and applications (2nd ed.)*, Springer, New York.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein. (2010). *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, Found. Trends Mach. Learn. **3**, 1–122.

Boyd, S. and L. Vandenberghe. (2004). *Convex Optimization*, Cambridge University Press. https://doi.org/10.1017/cbo9780511804441.005.

Demyanov, V.F. and V.N. Malozemov. (1977). *Introduction to Minimax*, Wiley, New York.

Jin, H. and S. Peng. (2016). *Optimal Unbiased Estimation for Maximal Distribution*. https://arxiv.org/abs/1611.07994.

Kellogg, R.B. (1969). *Nonlinear alternating direction algorithm*, Math. Comp. **23**, 23–38.

Kendall, M.G. and A. Stuart. (1968). *The Advanced Theory of Statistics, Volume 3: Design and Analysis, and Time-Series (2nd ed.)*, Griffin, London.

Kiwiel, K.C. (1987). *A Direct Method of Linearization for Continuous Minimax Problems*, J. Optim. Theory Appl. **55**, 271–287.

Klessig, R. and E. Polak. (1973). *An Adaptive Precision Gradient Method for Optimal Control*, SIAM J. Control **11**, 80–93.

Legendre, A.-M. (1805). *Nouvelles methodes pour la determination des orbites des cometes*, F. Didot, Paris.

Lin, L., Y. Shi, X. Wang, and S. Yang. (2016). *k-sample upper expectation linear regression-Modeling, identifiability, estimation and prediction*, J. Stat. Plan. Infer. **170**, 15–26.

Lin, L., P. Dong, Y. Song, and L. Zhu. (2017a). *Upper Expectation Parametric Regression*, Stat. Sin. **27**, 1265–1280.

Lin, L., Y.X. Liu, and C. Lin. (2017b). *Mini-max-risk and mini-mean-risk inferences for a partially piecewise regression*, Statistics **51**, 745–765.

Nocedal, J. and S.J. Wright. (2006). *Numerical Optimization*, Second Edition, Springer, New York.

Panin, V.M. (1981). *Linearization Method for Continuous Min-max Problems*, Kibernetika **2**, 75–78.

Peng, S. (2005). *Nonlinear expectations and nonlinear Markov chains*, Chin. Ann. Math. **26B**, no. 2, 159–184.

Seber, G.A.F. and C.J. Wild. (1989). *Nonlinear Regression*, Wiley, New York.