Greedy Algorithm for Sparse Monotone Regression*

Alexey R. Faizliev[0000−0001−6442−4361],
Alexander A. Gudkov[0000−0002−6472−4855],
Sergei V. Mironov[1][0000−0003−3699−5006], and Mikhail A. Levshunov

Saratov State University, Saratov, Russia
[1] sidorovsp@info.sgu.ru

Abstract. The problem of constructing the best fitted monotone regression is NP-hard problem and can be formulated in the form of a convex programming problem with linear constraints. The paper proposes a simple greedy algorithm for finding a sparse monotone regression using Frank–Wolfe-type approach. A software package for this problem is developed and implemented in R and C++. The proposed method is compared with the well-known pool-adjacent-violators algorithm (PAVA) using simulated data.

Keywords: greedy algorithms, pool–adjacent–violators algorithm, isotonic regression, monotone regression

1 Introduction

The recent years have seen an increasing interest in shape-constrained estimation in statistics [10]. One of such problems is the problem of constructing monotone regression. The problem is to find best fitted non-decreasing points to a given set of points on the plane. The survey of results on monotone regression can be found in the book by Robertson and Dykstra [25]. The papers of Barlow and Brunk [3], Dykstra [16], Best and Chakravarti [4], Best [5] consider the problem of finding monotone regression in quadratic and convex programming frameworks.

Using mathematical programming approach the works [1,21,31] have recently provided some new results on the topic. The papers [7,17] extend the problem to particular orders defined by the variables of a multiple regression. The paper [8] investigates a dual active-set algorithm for regularized monotonic regression.

Monotone regression is widely used in mathematical statistics [2, 10]; in smoothing of empirical data [15]; in shape-preserving approximation [19], [26], [30], [6], [27], [13]; in shape-preserving dynamic programming [9].

Constructing monotone regression we assume a relationship between a predictor $x = (x_1, \ldots, x_n)$ and a response $y = (y_1, \ldots, y_n)$. In the general case it is expected that $x_{i+1} - x_i \neq \text{const}$, $x_i < x_{i+1}$, $i = 1, \ldots, n-1$.

* The work was supported by RFBR (grant 16-01-00507).

A sequence $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$ is called monotone if

$$z_i - z_{i-1} \geq 0,\ i = 2, \ldots, n.$$

Denote Δ_1^n the set of all vectors from $\mathbb{R}^n$, which are monotone.

The problem of constructing monotone regression can be formulated in the form of a convex programming problem with linear constraints as follows: it is necessary to find a vector $z \in \mathbb{R}^n$ with the lowest mean square error of approximation to the given vector $y \in \mathbb{R}^n$ under condition of monotonicity of z:

$$f(z) = \frac{1}{n}\sum_{i=1}^{n}(z_i - y_i)^2 \to \min_{z \in \Delta_1^n}, \tag{1}$$

In many situations researchers have no information regarding the mathematical specification of the true regression function. Typically, this involves non-decreasing of y_i's with the ordered x_i's. Such a situation is called isotonic regression. Isotonic regression (monotone regression) is a special case to the k-monotone regression [24].

It is well-known that the problem (1) is NP-hard problem [24]. In this paper we present a simple greedy algorithm which employs Frank–Wolfe-type approach for finding sparse monotone regression. A software package for this problem is developed and implemented in R and C++.

For the convenience of solving the problem (1), we move from points z_i to its increments ζ_i, where $\zeta_i = z_{i+1} - z_i$, $i = 1, \ldots, n-1$, $\zeta_0 = z_1$. Then monotonicity of z corresponds non-negativity of ζ_i's (exept ζ_0). The proposed method is compared with the well-known pool-adjacent-violators algorithm (PAVA) using simulated data.

2 Algorithms for monotone regression

2.1 PAVA

Simple iterative algorithm for solving the problem (1) is called Pool-Adjacent-Violators Algorithm (PAVA) [11,24]. The work [4] examined the generalization of this algorithm. The paper [32] studied this problem as the problem of identifying the active set and proposed a direct algorithm of the same complexity as the PAVA (the dual algorithm).

PAVA computes a non-decreasing sequence of values $z = (z_i)_{i=1}^n$ such that the problem (1) is optimized. In the simple monotone regression case we have the measurement pairs (x_i, y_i). Let us assume that these pairs are ordered with respect to the predictors. The following (Algorithm 1) is a pseudocode of PAVA for the problem. The generalized pool-adjacent-violators algorithm (GPAVA), which is a strict generalization of PAVA, was developed in the article [33].

The block values are expanded with respect to the observations $i = 1, \ldots, n$ such that the final result is the vector z of length n with elements $\widetilde{z}_i$ of increasing order [24].

Algorithm 1: Pool-Adjacent-Violators Algorithm (PAVA)

begin

· Let $z_j^{(0)} := y_i$ be the start point, $l = 0$;

· The index for the blocks is $r = 1, \ldots, B$ where at step $l = 0$ we set $B := n$, i.e. each observation $z_r^{(0)}$ forms a block;

repeat

· (*Adjacent pooling*) Merge values of $z^{(l)}$ into blocks if $z_{r+1}^{(l)} < z_r^{(l)}$;

· Solve $f(z)$ for each block r, i.e., compute the update based on the solver which gives $z_r^{(l+1)} := s(z_r^{(l)})$, the solver s is conditional (weighted) mean and (weighted) quantities;

· If $z_{r+1}^{(l)} \leq z_r^{(l)}$ then $l := l + 1$;

until *the z-blocks are increasing, i.e.* $z_{r+1}^{(l)} \geq z_r^{(l)}$ *for all* r;

· Return z;

end

2.2 Frank–Wolfe type greedy algorithm

Frank–Wolfe method (or conditional gradient method) solves conditional convex optimization problems in vector finite-dimensional space. The method was introduced in 1956. The original algorithm did not use a fixed step size, and has the complexity of the linear programming. Frank–Wolfe method was developed by Levitin and Polyak in 1966, and V.F. Demianov and A.M. Rubinov generalized it to the case of arbitrary Banach spaces in 1970 [14]. Recently Frank–Wolfe type methods have caused an increased interest related to the possibility of obtaining sparse solutions, as well as a good scaling [12, 23]. In particular, [22, 34] researched algorithms for solving problems with penalty functions (instead of considering the conditional optimization problems). Besides, the paper [34] uses interlacing boosting with fixed-rank local optimization.

As it was mentioned above, for computational convenience of the problem (1), we moved from points z_i to increments $\zeta_i = z_{i+1} - z_i$, $i = 1, \ldots, n-1$, $\zeta_0 = z_1$. Then the problem (1) can be rewritten as follows:

$$g(\zeta) := \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=0}^{i-1} \zeta_j - y_i \right)^2 \to \min_{\zeta \in S}, \tag{2}$$

where S denotes the set of all $\zeta = (\zeta_0, \zeta_1, \ldots, \zeta_{n-1}) \in \mathbb{R}^n$ such that $\zeta_0 \in \mathbb{R}$, $(\zeta_1, \ldots, \zeta_{n-1}) \in \mathbb{R}_+^{n-1}$ and $\sum_{j=0}^{n-1} \zeta_j \leq \max_i y_i$.

Let $\nabla g(\zeta)$ denote the gradient of function g at point ζ.

It should be noted that for larger-scale problems the solution can appear computationally quite challenging. In this regard, the present study proposes to use a greedy algorithm of Frank–Wolfe type for solving this problem.

The following (Algorithm 2) is a pseudocode of Frank–Wolfe-type algorithm for the problem (2).

The rate of convergence is estimated according to the following theorem.

Algorithm 2: GREEDY ALGORITHM FOR SPARSE MONOTONE REGRESSION

begin

· Let N be the number of iteration;

· Our function $g(\zeta)$ and the feasible set S were defined above.

Let $\nabla g(\zeta) = \Big(\frac{\partial g}{\partial \zeta_0}, \frac{\partial g}{\partial \zeta_1}, \dots, \frac{\partial g}{\partial \zeta_{n-1}}\Big)$ be the gradient of function g at a point ζ,

$\frac{\partial g}{\partial \zeta_k} = \frac{2}{n}\sum\limits_{i=k}^{n}\Big(\sum\limits_{j=0}^{i-1}\zeta_j - y_i\Big),\ k = 0, \dots, n-1;$

· Let zero vector $\zeta^0 = (0, \dots, 0)$ be the start point, and let the counter $t = 0$;

· **while** $t < N$ **do**

  · Calculate $\nabla g(\zeta^t)$, the gradient of the function g at the point ζ^t;

  · Let $\widetilde{\zeta^t}$ be the solution of the linear optimization problem $\langle \nabla g(\zeta^t)^T, \zeta\rangle \to \min\limits_{\zeta \in S}$, where $\langle \nabla g(\zeta^t)^T, \zeta\rangle$ is a scalar product of vectors;

  · (*Update step*) Let $\zeta^{t+1} = \zeta^t + \alpha_t(\widetilde{\zeta^t} - \zeta^t)$, $\alpha_t = \frac{2}{t+2}$ and than $t := t+1$;

· Recover the monotone sequence $z = (z_1, \dots, z_n)$ from the vector of increments ζ^N;

end

Theorem 1. *Let $\{\zeta^t\}$ be generated according to the Frank–Wolfe method (Algorithm 2) using the step-size rule $\alpha_t = \frac{2}{t+2}$. Then for all $t \geq 2$*

$$g(\zeta^t) - g^* \leq 4\sqrt{\frac{n(n+1)(2n+1)}{6n^2}}\,\frac{(\max_i y_i - \min_i y_i)^2}{t+2}, \tag{3}$$

where g^* is the optimal solution of (2).

Proof. It it is know [18] that for all $t \geq 2$:

$$g(\zeta^t) - g^* \leq \frac{2L(\mathrm{Diam}(S))^2}{t+2},$$

where L is the Lipschitz constant and and $\mathrm{Diam}(S)$ is the diameter of S.

Let

$$\nabla^2 g(\zeta) := \left(\frac{\partial^2 g}{\partial \zeta_0^2}, \frac{\partial^2 g}{\partial \zeta_1^2}, \dots, \frac{\partial^2 g}{\partial \zeta_{n-1}^2}\right).$$

It is well-known that if ∇g is differentiable then its Lipschitz constant L satisfies the inequality

$$L \leq \sup_{\zeta} \|\nabla^2 g(\zeta)\|_2.$$

Then

$$L \leq \sup_{\zeta} \sqrt{\sum_{k=0}^{n-1} \left(\frac{\partial^2 g}{\partial \zeta_k^2}\right)^2} =$$

$$= \frac{1}{n} \sqrt{\sum_{k=1}^{n} (2(n-k+1))^2} = \frac{2}{n} \sqrt{\sum_{k=1}^{n} k^2} = 2\sqrt{\frac{n(n+1)(2n+1)}{6n^2}}. \quad (4)$$

It is easy to prove and $\mathrm{Diam}(S) := \sqrt{2}(\max_i y_i - \min_i y_i)$.

The disadvantage of this method is the dependence of the theoretical degree of convergence on the dimensionality of the problem. The papers [28], [20], [29] suggest to use the values of duality gap as the stopping criterion for Frank-Wolfe type algorithms.

3 Empirical Result

The algorithms have been implemented both in R and C++. We compared the performance of the greedy algorithm (Algorithm 2) with the performance of PAVA (Algorithm 1) using simulated data sets.

It should also be noted that the PAVA's speed is significantly higher for small-scale tasks in R. But if the number of points is greater than at least 2000, the greedy algorithm spends less time searching for a solution (Fig. 1).

Tables 1, 2 present empirical results for PAVA and greedy algorithms for a simulated set of points. The simulated points are obtained as the values of logarithm function with added normally distributed noise: $A = \{(x_i, y_i), y_i = \ln(x_0 + i\triangle x) + \varphi_i, \varphi_i \sim N(0,1), x_0 = 1, \triangle x = 1, i = 1, \ldots, 10000$. The dimension of the problem is 10000 points. The tables contain information on errors $\frac{1}{n}\sum_{i=1}^{n}(z_i - y_i)^2$, elapsed time, cardinality and greedy algorithm's iteration number.

The results show that error of greedy algorithm are getting closer to the error of PAVA with increase of number of iterations for greedy algorithm. While PAVA is better than greedy algorithm in terms of errors, the solutions of greedy algorithm have a better sparsity. Greedy algorithm's output solution is more sparse. It should be noted that the elapsed time for PAVA implemented in C++ is smaller than for greedy algorithm. However, greedy algorithm has a better rate of convergence if number of iterations is less than 700 for the algorithms implemented in R,. Both algorithms obtain a sparse solutions, but we can control the number of nonzero elements (cardinality) in the greedy algorithm as opposed to PAVA. Generally, the greedy algorithm's cardinality increases by one at each iteration. Consequently, we should limit the number of iterations to obtain more sparse solution.

Figure 2 shows simulated points ($N = 100$) with logarithm structure and isotonic regressions, where green line represents the greedy algorithm's isotonic

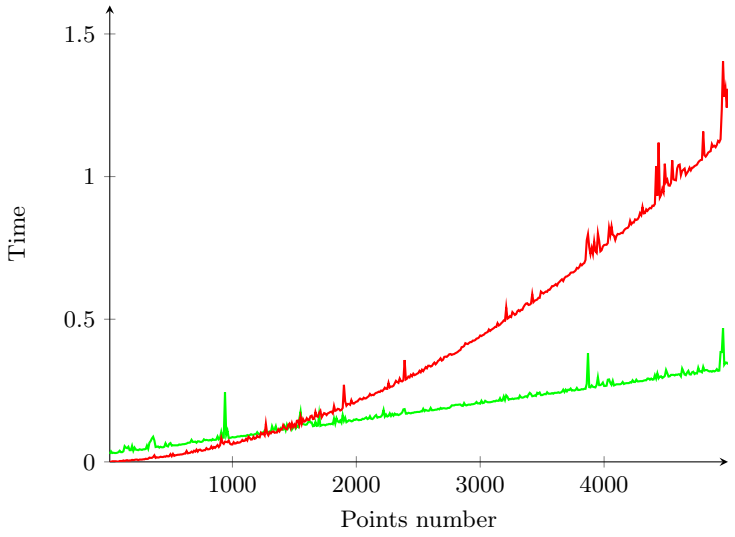

Fig. 1. The dependence of the CPU time on dimension of the problem of the greedy algorithm (green line, 100 iterations) and PAVA (red line) implemented in R.

Table 1. Comparison of algorithms PAVA and greedy algorithm (Greedy) on an example of the simulated data (implementation in language C++): $A = \{(x_i, y_i), y_i = \ln(x_0 + i\triangle x) + \varphi_i, \varphi_i \sim N(0,1), x_0 = 1, \triangle x = 1, i = 1, \ldots, 10000$

Algorithm (the number of iterations)	Error	Cardinality	Time
PAVA	0.994	82	0.00
Greedy (10)	1.254	7	0.00
Greedy (50)	1.101	31	0.00
Greedy (100)	1.003	43	0.01
Greedy (200)	1.001	55	0.01
Greedy (500)	0.995	74	0.03
Greedy (1000)	0.995	78	0.07
Greedy (2000)	0.994	82	0.09
Greedy (5000)	0.994	82	0.17
Greedy (10000)	0.994	82	0.33

regression and red line presents PAVA's isotonic regression. Greedy algorithm gives a solution with 14 jumps, and PAVA with 16 jumps. Since the solutions of the greedy algorithm are more sparse, the greedy algorithm error (ε) is slightly higher than the PAVA.

The obtained empirical results for the greedy algorithm show that the degree of convergence for the considered examples is much higher than its theoretical estimates obtained in Theorem 1.

Table 2. Comparison of algorithms PAVA and greedy algorithm (Greedy) on an example of the simulated data (implementation in language R): $A = (x_i, y_i)$

Algorithm (the number of iterations)	Error	Cardinality	Time
PAVA	0.994	82	4.28
Greedy (10)	1.169	6	0.09
Greedy (50)	1.011	30	0.33
Greedy (100)	0.999	41	0.63
Greedy (200)	0.996	57	1.23
Greedy (500)	0.995	76	3.08
Greedy (1000)	0.994	79	6.28
Greedy (2000)	0.994	82	12.67
Greedy (5000)	0.994	82	31.58
Greedy (10000)	0.994	82	60.91

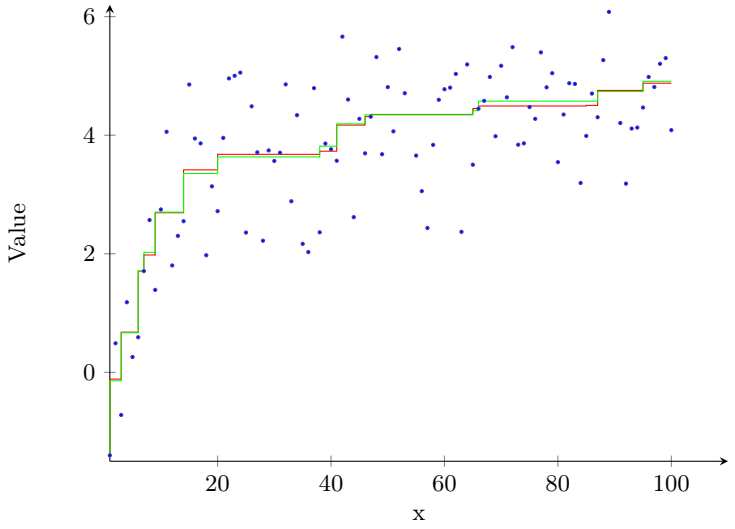

Fig. 2. Step functions obtained by the greedy algorithm ($\varepsilon = 0.753$) and PAVA ($\varepsilon = 0.751$)

4 Conclusion

Our research proposes an algorithm for solving the problem of constructing the best fitted monotone regression by using the Frank–Wolfe method. The software was implemented in R and C++. We compared the performance of the greedy algorithm with the performance of PAVA using simulated data sets. While PAVA gives a slightly smaller errors than greedy algorithm, greedy algorithm obtains significantly sparser solutions. The advantages of greedy algorithm are the simplicity of implementation, the potential for controlling cardinality and the

elapsed time is lower for the implementation in R in the case of problem with large dimension.

References

1. Ahuja, R., Orlin, J.: A fast scaling algorithm for minimizing separable convex functions subject to chain constraints. Operations Research 49(1), 784–789 (2001)
2. Bach, F.: Efficient algorithms for non-convex isotonic regression through submodular optimization. Tech. Rep. hal-01569934 (Jul 2017), https://hal.archives-ouvertes.fr/hal-01569934, working paper or preprint
3. Barlow, R., Brunk, H.: The isotonic regression problem and its dual. Journal of the American Statistical Association 67(1), 140–147 (1972)
4. Best, M.J., Chakravarti, N.: Active set algorithms for isotonic regression: a unifying framework. Mathematical Programming: Series A and B 47(3), 425–439 (1990)
5. Best, M., Chakravarti, N., Ubhaya, V.: Minimizing separable convex functions subject to simple chain constraints. SIAM Journal on Optimization 10(1), 658–672 (2000)
6. Boytsov, D.I., Sidorov, S.P.: Linear approximation method preserving k-monotonicity. Siberian electronic mathematical reports 12, 21–27 (2015)
7. Burdakov, O., Grimvall, A., Hussian, M.: A generalised PAV algorithm for monotonic regression in several variables. In J Antoch (ed.), COMPSTAT, Proceedings of the 16th Symposium in Computational Statistics 10(1), 761–767 (2004)
8. Burdakov, O., Sysoev, O.: A dual active-set algorithm for regularized monotonic regression. Journal of Optimization Theory and Applications 172(3), 929–949 (2017)
9. Cai, Y., Judd, K.L.: Shape-preserving dynamic programming. Math. Meth. Oper. Res. 77, 407–421 (2013)
10. Chen, Y.: Aspects of Shape-constrained Estimation in Statistics. Ph.D. thesis (2013)
11. Chepoi, V., Cogneau, D., Fichet, B.: Polynomial algorithms for isotonic regression. Lecture Notes–Monograph Series 31(1), 147–160 (1997)
12. Clarkson, K.L.: Sparse greedy approximation, and the Frank-Wolfe algorithm. ACM Transactions on Algorithms 6(4), 1–30 (2010)
13. Cullinan, M.P.: Piecewise convex-concave approximation in the minimax norm. In: Demetriou, I., Pardalos, P. (eds.) Abstracts of Conference on Approximation and Optimization: Algorithms, Complexity, and Applications, June 29Ц30, 2017, Athens, Greece. p. 4. National and Kapodistrian University of Athens (2017)
14. Demyanov, V.F., Rubinov, A.M.: Approximate Methods in Optimization Problems. (Modern Analytic and Computational Methods in Science and Mathematics). American Elsevier Publishing Company. New York (1970)
15. Diggle Peter, M.S., Tony, M.J.: Case-control isotonic regression for investigation of elevation in risk around a point source. Statistics in medicine 18(1), 1605–1613 (1999)
16. Dykstra, R.: An isotonic regression algorithm. Journal of Statistical Planning and Inference 5(1), 355–363 (1981)
17. Dykstra, R., Robertson, T.: An algorithm for isotonic regression for two or more independent variables. The Annals of Statistics 10(1), 708–719 (1982)
18. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quarterly 3(1-2), 95–110 (1956)

19. Gal, S.G.: Shape-Preserving Approximation by Real and Complex Polynomials. Springer (2008)
20. Gudkov, A.A., Mironov, S.V., Faizliev, A.R.: On the convergence of a greedy algorithm for the solution of the problem for the construction of monotone regression. Izv. Saratov Univ. (N. S.), Ser. Math. Mech. Inform. 17, 431–440 (2017)
21. Hansohm, J.: Algorithms and error estimations for monotone regression on partially preordered sets. Journal of Multivariate Analysis 98(1), 1043–1050 (2007)
22. Harchaoui, Z., Juditsky, A., Nemirovski, A.: Conditional gradient algorithms for norm-regularized smooth convex optimization. Mathematical Programming: Series A and B 152(1-2), 75–112 (2015)
23. Jaggi, M.: Sparse Convex Optimization Methods for Machine Learning. Ph.D. thesis, ETH Zürich (2001)
24. Leeuw, J., Hornik, K., P., M.: Isotone optimization in R: Pool-adjacent-violators algorithm (PAVA) and active set methods. Journal of Statistical Software 32(5), 1–24 (2009)
25. Robertson, T., Wright, F., Dykstra, R.: Order Restricted Statistical Inference. John Wiley & Sons, New York (1988)
26. Shevaldin, V.T.: Local approximation by splines. UrO RAN, Ekaterinburg (2014)
27. Sidorov, S.P.: Linear k-monotonicity preserving algorithms and their approximation properties. LNCS 9582, 93–106 (2016)
28. Sidorov, S.P., Mironov, S.V.: Duality gap analysis of weak relaxed greedy algorithms. LNCS 10556, 251–262 (2017)
29. Sidorov, S.P., Mironov, S.V., Pleshakov, M.G.: Dual convergence estimates for a family of greedy algorithms in banach spaces. LNCS 10710 (2018), in press
30. Sidorov, S.: On the saturation effect for linear shape-preserving approximation in Sobolev spaces. Miskolc Mathematical Notes 16(2), 1191–1197 (2015)
31. Stromberg, U.: An algorithm for isotonic regression with arbitrary convex distance function. Computational Statistics & Data Analysis 11(1), 205–219 (1991)
32. Wu, W.B., Woodroofe, M., Mentz, G.: Isotonic regression: Another look at the changepoint problem. Biometrika 88(3), 793–804 (2001)
33. Yu, Y.L., Xing, E.P.: Exact algorithms for isotonic regression and related. Journal of Physics: Conference Series 699(1), 1–9 (2016)
34. Zhang, X., Yu, Y., Schuurmans, D.: Accelerated training for matrix-norm regularization: A boosting approach. NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems 1(1), 2906–2914 (2012)