An Approach to Parallelizing Isotonic Regression

Anthony Kearsley
Richard Tapia
Michael Trosset

CRPC-TR96640
January 1996

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

An Approach to Parallelizing Isotonic Regression

Anthony J. Kearsley* Richard A. Tapia† Michael W. Trosset‡

January 19, 1996

Abstract

Isotonic regression is the problem of fitting data to order constraints. We demonstrate that the isotonic regression of a finite set of numbers Y can be obtained by decomposing Y into subsets, performing parallel isotonic regressions on each subset, then performing a trivial isotonic regression on the resulting combined set. Numerical experiments confirm the efficacy of this approach.

*Department of Mathematics, University of Massachusetts at Dartmouth.

†Department of Computational & Applied Mathematics and Center for Research in Parallel Computation, Rice University. This author was supported in part by NSF Cooperative Agreement No. CCR9120008 and DOE DEFG05-86ER25017.

‡Department of Computational & Applied Mathematics (adjunct), Rice University; Department of Psychology (adjunct), University of Arizona; and Consultant, P. O. Box 40993, Tucson, AZ 85717-0993. This author was supported in part by NSF Cooperative Agreement No. CCR9120008, as a visiting member of the Center for Research in Parallel Computation, Rice University, August 1993 and 1994.

1 Introduction

Given a finite set of real numbers, $Y = \{y_1, \ldots, y_n\}$, the problem of isotonic regression with respect to a complete order is the following quadratic programming problem:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} w_i (x_i - y_i)^2 \\ \text{subject to} & x_1 \leq \cdots \leq x_n, \end{array}$$

where the w_i are strictly positive weights. Many important problems in statistics and other disciplines can be posed as isotonic regression problems. Comprehensive surveys of this subject were made by Barlow, Bartholomew, Bremner, and Brunk [1] and by Robertson, Wright, and Dykstra [5] in their respective monographs.

The fundamental concern of the present report is revealed by considering a simple example. Suppose that $\{1, 3, 2, 4, 5, 7, 6, 8\}$ is the given set of real numbers, and that the weights are all identically one. This set is almost isotonic; however, {3,2} and {7,6} violate the requirement that the numbers are nondecreasing. The antidote to this difficulty is very simple: replacing each "block" of "violators" with the average of the numbers in the block produces $\{1, 2.5, 2.5, 4, 5, 6.5, 6.5, 8\}$, which turns out to be the unique solution of the isotonic regression problem. This is an example of the well-known "Pool Adjacent Violators" algorithm.

What is intriguing about this very simple example is that the two computations required to produce the isotonic regression do not depend on each other and could be performed simultaneously. Furthermore, this property appears to be a characteristic of the isotonic regression problem itself, not of the algorithm used to solve it. Whatever the computational algorithm that is employed, it is obvious that the isotonic regressions of the subsets $\{1, 3, 2, 4\}$ and $\{5, 7, 6, 8\}$ are easily combined to produce the isotonic regression of the entire set. It is this observation that motivates the rigorous derivation of a foundation for using parallel computation to solve isotonic regression problems.

2 A Decomposition Theorem

To attain the necessary rigor, we exploit a famous and very elegant characterization of the solution to the isotonic regression problem. Let $W_j = \sum_{i=1}^{j} w_j$, let P_0 denote the point $(0, 0)$, and let P_j denote the point $(W_j, \sum_{i=1}^{j} w_i y_i)$, for $j = 1, \ldots, n$. We interpret $P_0, \ldots, P_n$ as points in the graph of a function, which we extend to the interval $[0, W_n]$ by linear interpolation. Both the function and its graph are called the *cumulative sum diagram* (CSD) of the isotonic regression problem.

The *greatest convex minorant* (GCM) of a function f is the convex function defined by

$$GCM[f] = \sup \{\phi : \phi \text{ convex}, \phi \leq f\}.$$

It is a well-known and beautiful result that the isotonic regression problem is solved by taking x_j^* to be the left derivative of $GCM[CSD]$ at W_j. Thus, theorems about isotonic regressions can be stated and proved as theorems about greatest convex minorants.

The particular theorem on which the ideas in this report are based is quite elementary, yet it has profound implications for parallel computation. Suppose that we decompose the set Y into $Y_1 \oplus Y_2$, where $Y_1 = \{y_1, \ldots, y_k\}$ and $Y_2 = \{y_{k+1}, \ldots, y_n\}$. Analogously, we can decompose a function f with domain $[0, W_n]$ into $f_1 \oplus f_2$, where f_1 is the restriction of f to $[0, W_k]$ and f_2 is the restriction of f to $(W_k, W_n]$. Then the following result is easily demonstrated.

Theorem 1 $GCM[\ GCM[f_1] \oplus GCM[f_2]\ ] \ = \ GCM[f]$

Proof: Because this result is of fundamental importance to this report, we provide a detailed proof.

Since $GCM[f_1] \leq f_1$ and $GCM[f_2] \leq f_2$,

$$GCM[f_1] \oplus GCM[f_2] \leq f_1 \oplus f_2 = f.$$

It follows that, if $\phi \leq GCM[f_1] \oplus GCM[f_2]$, then $\phi \leq f$, and hence that

$$GCM[\ GCM[f_1] \oplus GCM[f_2]\ ] \ \leq \ GCM[f]. \tag{1}$$

Conversely, suppose that $\phi \leq f$ is convex and write $\phi = \phi_1 \oplus \phi_2$. Then $\phi_1 \leq f_1$ and $\phi_2 \leq f_2$, so $\phi_1 \leq GCM[f_1]$ and $\phi_2 \leq GCM[f_2]$. It follows that $\phi \leq GCM[f_1] \oplus GCM[f_2]$, and hence that

$$GCM[f] \ \leq \ GCM[\ GCM[f_1] \oplus GCM[f_2]\ ]. \tag{2}$$

Combining inequalities (1) and (2) gives the desired result. $\square$

3 Implications for Parallel Computation

If one takes the function f to be the CSD for the isotonic regression problem, then Theorem 1 states the following: decomposing Y into $Y_1 \oplus Y_2$, performing separate isotonic regressions on Y_1 and Y_2, and then performing a final isotonic regression on the combined result, produces the isotonic regression on Y. Because the separate isotonic regressions on Y_1 and Y_2 can be performed simultaneously, parallel computations of isotonic regressions will be desirable if the final isotonic regression on the combined result is easy to compute. In point of fact, this is the case.

Suppose that Y_1 satisfies $y_1 \leq \cdots \leq y_k$ and Y_2 satisfies $y_{k+1} \leq \cdots \leq y_n$. If $y_k \leq y_{k+1}$, then Y is isotonic. If Y is not isotonic, then it must be because some of the largest numbers in Y_1 exceed some of the smallest numbers in Y_2. The antidote to this difficulty is to identify this central block of offending numbers and to replace each of these numbers with the weighted average of the block. (This is just the Pool Adjacent Violators algorithm again.) To accomplish this, let

$$m \ = \ \min\ \{i : y_i > y_{k+1}\},$$
$$M \ = \ \max\ \{i : y_i < y_k\},$$

and

$$\bar{y} \ = \ \sum_{i=m}^{M} w_i y_i \ / \ \sum_{i=m}^{M} w_i.$$

Then, replacing y_i with $\bar{y}$ for $i = m, \ldots, M$ gives the isotonic regression of Y. Thus, if one decomposes the isotonic regression problem and performs two smaller, separate isotonic regressions, it becomes fairly simple to obtain the solution to the original problem.

By now it should be apparent that what is being proposed in this report is *not* a new, parallel algorithm for isotonic regression that will compete with existing algorithms. Rather, it is the isotonic regression problem itself that has been parallelized. (An instructive analogy is the familiar exercise of sorting a list of numbers by subdividing the list, sorting each sublist, then interweaving the sorted sublists.) Because the problem itself has been parallelized, *any* isotonic regression algorithm can be used to compute the separate isotonic regressions assigned to separate processors. The efficiency of various isotonic regression algorithms has been discussed by Best and Chakravarti [2]. A very fast formulation of the Pool Adjacent Violators algorithm was provided by Grotzinger and Witzgall [3].

In light of the preceding arguments, we are virtually assured that a parallel approach to isotonic regression will speed up computation when n is sufficiently large. This phenomenon is demonstrated in Section 4. Notice, however, that we should not expect that the most efficient strategy will necessarily be the one that uses the largest number of processors, since the more that the original problem is decomposed, the more difficult it becomes to obtain the final solution from the separate isotonic regressions. As an extreme example of this limitation, one might decompose Y into n subsets of singleton values, in which case nothing whatsoever has been accomplished. Furthermore, the more that the original problem is decomposed, the greater the communication costs of parallelization. Hence, it is impossible to anticipate the most efficient decomposition strategy.

4 Numerical Experiments

To obtain a suite of isotonic regression problems, we imagined the problem of measuring the viscosity of a fluid at different temperatures. This problem motivates the models that we describe, although ultimately we are more concerned with varying conditions that might affect computational performance than with faithfully modelling physical reality.

Viscosity is a nonincreasing function of temperature; however, due to measurement error, the observed viscosities may not be nonincreasing when ordered by temperature. In this case, one might want to replace the vector of observed viscosities with the nearest vector that is nonincreasing when ordered by temperature. This can be posed as an isotonic regression problem with unit weights.

As have Kearsley [4] and others, we assumed that viscosity (η) is exponentially dependent on temperature (T):

$$\eta = \eta_0 \exp(-\alpha T).$$

For our experiments, we set $\eta_0 = 1$ and $\alpha = 10^{-4}$. Then, in order to obtain an increasing function, we set

$$y = f(t) = 100 - \eta_0 \exp(-\alpha t)$$

and computed $y_k = f(t_k)$ at $n = 10^6$ equally-spaced grid points in the interval $[0, 100]$. The resulting set Y of n increasing numbers was perturbed in various ways to obtain the data sets that we subjected to isotonic regression. Each of ten strategies for perturbing the original set of numbers was replicated $R = 5$ times, resulting in a total of fifty data sets.

Let $\sigma = \log(2)/1.95996$ be fixed. In what follows, whenever we perturb a value y_k, we do so by replacing y_k with $y_k \exp(\sigma z)$, where z is a standard normal deviate. This multiplicative model of measurement error was constructed so that approximately 95 percent of the perturbed values would be at least one half and no more than twice the replaced value.

The following loops describe our perturbation strategies. In each case, the intent was to perturb P values in the form of B blocks of length L.

For $R = 1$ to 5 repetitions:

1. Perturb *each* of the n values in the original data set Y to obtain data set Y1000.R.
2. For $P = .49n, .25n, .01n$ and $L = 1, \sqrt{P}, P$:
   (a) Randomly select $B = P/L$ numbers from $\{1, \ldots, n/L\}$ without replacement. Call these numbers $s_1, \ldots, s_B$.
   (b) Let $\pi = n/P$. For $i = 1, \ldots, B$ and $j = 0, \ldots, L-1$, let $k = \pi s_i + j$ and perturb each original value y_k.
   (c) Denote the resulting data set by Y0ppl.R, where $pp = 100P/n$ and

   $$l = 2\log(L)/\log(P).$$

   For example, the data set produced on the fourth repetition of the case for which $P = .49n$ and $L = \sqrt{P}$ is denoted by Y0491.4.

Thus, we generated five data sets (Y1000) in which all values were perturbed, fifteen data sets (Y049l) in which 49 percent of the values were perturbed, fifteen data sets (Y025l) in which 25 percent of the values were perturbed, and fifteen data sets (Y001l) in which 1 percent of the values were perturbed. Furthermore, in each of the cases that $P = .49n, .25n, .01n$ of the values were perturbed, we generated five data sets (Y0pp0) in which we perturbed P isolated values, five data sets (Y0pp1) in which we perturbed $\sqrt{P}$ blocks of $\sqrt{P}$ consecutive values, and five data sets (Y0pp2) in which we perturbed one block of P consecutive values. This allowed us to investigate the effect of different data structures on the efficacy of parallel computation.

Each of the fifty data sets was submitted to six isotonic regressions on the Intel Touchstone Delta parallel computing system at the California Institute of Technology. These regressions used respectively $A = 1, 2, 4, 8, 16, 32$ of the Delta's processors. For each regression, the data set was decomposed into A subsets of (approximately) equal size. Each subset was simultaneously sent to a separate processor, where its isotonic regression was computed using Grotzinger's and Witzgall's [3] formulation of the Pool Adjacent Violators algorithm. As soon as the isotonic regressions of two consecutive subsets were computed, the combined result was sent to one of the available processors, which then computed the combined isotonic regression by means of the device described in Section 3. This process was continued until the isotonic regression of the entire data set was obtained. The elapsed time from job submission to completion was measured by the Delta's intrinsic timer. The results are summarized in Table 1.

Table 1 exhibits several striking features. First, the variations in times produced by $R = 5$ replications are extremely small relative to the magnitudes of the times. In retrospect

Table 1: Sample means and standard deviations ($\bar{y} \pm s_y$) of elapsed times in milliseconds for five repetitions of ten isotonic regression experiments.

Data	Number of Processors					
Sets	1	2	4	8	16	32
Y1000	2278 ± 93	1376 ± 22	1158 ± 16	930 ± 7	1062 ± 25	1058 ± 15
Y0490	2406 ± 142	1416 ± 5	1182 ± 4	938 ± 8	1062 ± 4	1068 ± 8
Y0491	2376 ± 180	1436 ± 29	1208 ± 50	958 ± 19	1060 ± 14	1080 ± 27
Y0492	2370 ± 171	1378 ± 8	1152 ± 8	922 ± 4	1032 ± 4	1036 ± 15
Y0250	2396 ± 54	1386 ± 9	1144 ± 11	944 ± 48	1040 ± 14	1040 ± 12
Y0251	2298 ± 128	1410 ± 7	1174 ± 5	936 ± 5	1058 ± 4	1052 ± 4
Y0252	2330 ± 95	1406 ± 9	1172 ± 4	942 ± 4	1066 ± 9	1058 ± 4
Y0010	2232 ± 30	1378 ± 4	1150 ± 7	922 ± 4	1034 ± 5	1032 ± 8
Y0011	2368 ± 156	1410 ± 10	1188 ± 24	940 ± 0	1062 ± 4	1062 ± 4
Y0012	2380 ± 152	1418 ± 8	1184 ± 22	944 ± 5	1068 ± 18	1070 ± 12

this is not surprising: each data set contains a very large number of independent errors, so that one should expect that most data sets constructed in accordance with a specific perturbation strategy will be quite similar.

Second, there is very little variation in mean timing profiles between the ten perturbation strategies. This suggests that the phenomena described below are not unique to a particular data structure.

As anticipated, it is apparent that some degree of parallelization decreases the time required to perform an isotonic regression. For the 50 data sets that we considered, the time required by $A = 2$ processors divided by the time required by $A = 1$ processor ranged from a minimum of 53.2% to a maximum of 65.9%, with a median of 60.3%. The time required by $A = 4$ processors divided by the time required by $A = 1$ processor ranged from a minimum of 44.9% to a maximum of 54.8%, with a median of 50.1%. The time required by $A = 8$ processors divided by the time required by $A = 1$ processor ranged from a minimum of 35.7% to a maximum of 43.5%, with a median of 40.5%. Thus, there is compelling evidence that, for $n = 10^6$ and these types of data sets, using $A = 8$ processors is more efficient than using $A = 4, 2, 1$ processors.

For $A = 16, 32$ processors, the communication costs of the parallelization strategy begin to dominate and the times are actually slower than for $A = 8$ processors. This phenomenon was also anticipated. With larger data sets, we know that we can take advantage of additional processors, but the tradeoff between n and the optimal A must be empirically determined for the data structures and parallel computing system of interest.

Finally we note that, although the proportional improvements in efficiency produced by parallel processing are impressive, the absolute times for serial processing are small. At present, it is difficult to forsee applications involving isotonic regressions on data sets so large that the absolute savings in time will warrant parallel computation. Perhaps that day will come; for now, our primary interest in parallelizing isotonic regression is for the pedagogical value of so doing. In our view, isotonic regression is a remarkably simple and elegant example of a problem for which mathematical theory virtually guarantees that parallelization will be beneficial.

Acknowledgements

The authors thank Christoph Witzgall and Andrea Reiff for sharing their thoughts about isotonic regression algorithms. This research was performed in part using the Intel Touchstone Delta System operated by the California Institute of Technology on behalf of the Concurrent Supercomputing Consortium. Access to this facility was provided by the Center for Research on Parallel Computing under NSF Cooperative Agreement No. CCR9120008.

References

[1] R. E. Barlow, J. M. Bartholomew, J. M. Bremner, and H. D. Brunk. *Statistical Inference Under Order Restrictions.* John Wiley & Sons, New York, 1972.

[2] M. J. Best and N. Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47:425–439, 1990.

[3] S. J. Grotzinger and C. Witzgall. Projections onto order simplexes. *Applied Mathematics and Optimization*, 12:247–270, 1984.

[4] A. J. Kearsley. A steady state model of Couette flow with viscous heating. *International Journal of Engineering Science*, 32:179–186, 1994.

[5] T. Robertson, F. T. Wright, and R. L. Dykstra. *Order Restricted Statistical Inference.* John Wiley & Sons, New York, 1988.