DATA FLOW ANALYSIS AND THE LINEAR PROGRAMMING MODEL[1]

Levon Aslanyan

Abstract: The general discussion of the data flow algorithmic models, and the linear programming problem with the variating by data flow criterion function coefficients are presented. The general problem is widely known in different names - data streams, incremental and online algorithms, etc. The more studied algorithmic models include mathematical statistics and clustering, histograms and wavelets, sorting, set cover, and others. Linear programming model is an addition to this list. Large theoretical knowledge exists in this as the simplex algorithm and as interior point methods but the flow analysis requires another interpretation of optimal plans and plan transition with variate coefficients. An approximate model is devised which predicts the boundary stability point for the current optimal plan. This is valuable preparatory information of applications, moreover when a parallel computational facility is supposed.

Keywords: data flow algorithm, linear programming, approximation

ACM Classification Keywords: G.1.6 Numerical analysis: Optimization

1. Introduction

Data flow is a concept, traditionally appearing in the sensor based monitoring systems. Advanced global networks brought a number of novel applied disciplines intensively dealing with data flows. The network monitoring itself and optimal management of telecommunication systems, search engines with consequent data analysis, the network measuring instruments and network monitoring for security, etc. are the novel examples of data flow models. These deal with continuous data flows and unusual, non-finite and non-stored data set. In this case, the queries (the data analysis requests) are long-term and continuous processes in contrast to usual one-time queries. The traditional databases and data processing algorithms are poorly adjusted for the hard and continuous queries in data flows. This generates the necessity of new studies for serving continuous, multilayered, depending on time and subjected to indefinite behaviour of data flows [MM 2003]. Concerning the mentioned problem area, systems and algorithms are devised for different needs: real time systems, automation control systems, modelling processes, etc., but they are episodes in point of view of the formulated general problem. Traditional trade offs of such systems include one-pass and multi-pass algorithms, deterministic and randomized algorithms, and exact and approximate algorithms. Off-line algorithms solve a problem with full knowledge of the complete problem data. Online algorithms construct partial solutions with partial knowledge of the problem data, and update their solutions every time some new information is provided. In other words, they must handle a sequence of closely related and interleaved sub-problems, satisfying each sub-problem without knowledge of the future sub-problems. Standard examples of online problems include scheduling the motion of elevators, finding routes in networks and allocating cache memory. The usual way of measuring the quality of an online algorithm is to compare it to the optimal solution of the corresponding off-line problem where all information is available at the beginning. An online algorithm that always delivers results that are only a constant factor away from the corresponding optimum off-line solution, is called a competitive algorithm.

The “incremental update” algorithmic model of data analysis [AJ 2001] modifies the solution of a problem that has been changed, rather than re-solving the entire problem. For example, partial change of conditions of a time-table problem must be force to only partial reconstruction of the table. It is obvious that it is possible to construct a

[1] The research is supported partly by INTAS: 04-77-7173 project, http://www.intas.be

theoretical problem, where any particular change brings to the full reconstruction of the problem. It is also clear that there are numerous problems, which are not so critical to the local transformations. It is an option to try to solve the given data flow problem by the mentioned incremental algorithms, moreover, in the specific conditions it is the only possible way for solving the problem including the data flows analysis.

Measuring the "variability", "sortedness" and similar properties of data streams could be useful in some applications; for example, in determining the choice of a compression or sort algorithm for the underlying data streams. [MM 2003] have studied the bit level changes in video sequences and [AJ 2001] - the problem of estimating the number of inversions (the key element of the Shell type sorting algorithms) in a permutation of length n to within a factor $1 \pm \varepsilon$, where the permutation is presented in a data stream model. [MM 2003] proves the decreasing bit level changes of image pixels in video sequences and in [AJ 2001] - an algorithm obtained requiring the space $O(\log n \log \log n)$.

Sketching tools are usual for many data oriented applications. These include approximations of statistical parameters, histograms, wavelets, and other similar general descriptors. The simplest calculations for data streams serve the base statistical means like the averages and variations [AC 2003]. Other data flow descriptors also appear in publications: frequency moments [AM 1996], histograms [GG 2002], etc.

The paper below discusses an important applied model for the flow environments. We consider the linear programming mathematical problem, parameters of which are formed by data flows. In a moment it is assumed that the optimal plan is found and the coordinates of the target function are variable by the flow. In this case, there is an emerging question: which mechanisms are well suited to follow the coefficients variations in creating the configuration of the next resulting optimal plan. It is clear that the small changes of coefficients lead to simple changes of the current optimal plan, probably not requiring the total analysis of the problem by the complete flow information.

2. Linear Programming in Data Flows

Let's consider the linear programming problem in its canonical form:

$$\min\ c'x$$
$$Ax = b, \quad x \geq 0,$$

where $c \in R^n$, $b \in R^m$, A is an $m \times n$ full rank real matrix, and $m < n$. Without a loss of generality we may also suppose that $b_i \geq 0$, $i = \overline{1,m}$. Particular examples of linear programming problem are given through the definition of coefficient values: a_{ij}, c_j, b_i, for $i = \overline{1,m};\ j = \overline{1,n}$. Let's imagine a specific situation arising from application that the mentioned coefficients are changing in time. Such problems appear, for example, in data flow analysis.

Le us consider a data flow $B(t,n)$, which is finite but a very large-sized sequence of values $b_1, b_2, \cdots, b_n$, where $b_t, t = \overline{1,n}$ are certain structures. The data flows processing algorithms may use comparatively small storages than the input size. A limitation window is given in certain cases for processing separate data fragments. The time-dependent values of parameters, forming the applied problem model are formed as a result of algorithmic analysis. Unlike other natural and similar definitions, the variation of parameters is unpredictable here, as it has not probabilistic distribution and is not described by one or another property. Instead, it is considered that the variation takes place very slowly, because of the accumulation concept. In its turn, the applied problem demands to have ready answers to the certain questions in each time stamp.

There are two strategies: (1) solving a problem for each stage by all actual information which is practically impossible because of the large data sizes; and (2) structuring hereditary systems when the new data analysis is relatively easily integrated with the results of the previous data analysis.

We are going to consider the linear programming model in the mentioned conditions. The arbitrary variation of the coefficients is not allowed, instead, slow variations are considered so that the variation is fully monitored and it

changes the solutions very slowly. Of course, it is possible to formalize this fully. At the same time, it is possible to consider partial behaviour of parameters variations, providing simple scenes of the algorithmic developments.

Let's suppose that the coefficients c_j of linear criterion function $Z = \sum_{j=1}^{n} c_j x_j$ of the linear programming problem are varying by flow $B(t,n)$. Assume that t_0 is the moment where the complete analysis exists, i.e. we know about the existence of optimization at that moment and the optimal vertex and plan, if the latter exists. This vertex obeys the property of stability of optimality for certain variations of coefficients c_j [GL 1980]. The stability area is described by a set of simple inequalities and it is clear that it is an issue to consider the border of this area. The theoretical analysis of optimality of vertex set elements of the area of base restrictions is well known as the simplex method [V 1998]. The simplex method looks for sequence chains of vertex transitions, which converge to an optimal plan. Complementary, in our case, we study all the possible ways of optimality transitions driven by the changes of coefficients c_j.

The novelty is that we device the concept of equivalency of vertices groups of the feasible polyhedron vertices set and prove that the transition from one optimal vertex to another takes place through these groups. So the continuous change of target function coefficients generates the continuous change of optimality vertices.

From practical point of view – a path prediction process is possible to apply to the situation with varying coefficients. Prediction of intersection of the trajectory extrapolation of coefficient changes to the boundary of stability area of the current optimal plan helps to determine the vertex equivalency cluster and so - the further possible transitions and by these – the most likely arriving optimums when coefficients keep the track of their current modifications.

Going through the transitions some vertices might neighbour with comparatively large equivalency groups of vertices and then the total number of those vertices can become large. Theoretically, in terms of flows, having the current optimization vertex, it is necessary to prepare neighbouring equivalent vertices by calculating them by the current and predicted coefficients c_j. The weakness of the direct application of the given approach is in drastic increase in the number of calculations for the vertex sets involving the predictions and equivalencies. The considered below natural approach gives primary significance to the vertices which correspond to the linear approximations of the given variations.

Let's denote the optimal vertex at the moment t_0 by $\tilde{x}^{t_0}$ and let $\tilde{c}^{t_0}$ is the corresponding vector of coefficients. Let's follow the transition of $\tilde{c}^{t_0}$ to the $\tilde{c}^{t}$. It is clear that this transition is more or less arbitrary and it is controlled by the flow. It is important if during the transition the vector $\tilde{c}^{t}$ of coefficients approaches to the boarder of stability of current optimal plan $\tilde{x}^{t_0}$, - or not. To see this we have to monitor the changes of $\tilde{c}^{t}$. Alternatively, it is possible to approximate the transition, activating the possible future “optimal plans”. For example, spline approximations or a more simple approximation by the main values and standard deviations might be applied. The most simple is the linear approximation model, which we consider below. As an extrapolation, it leads to the intersection with the stability boundary (shell) of the vertex $\tilde{x}^{t_0}$ at the most probability point. In case of sufficient computational resources, it is also possible to consider some neighbourhood of that point, and it is important that in contrast to the above mentioned theoretical model, this applied approach gives an opportunity to work with the limited quantity of the possible candidate vertices. Depending on the considered problem, an algorithmic system is able to choose a corresponding extrapolation scheme, which deals with different numbers of neighbouring vertices. The approximation of $\tilde{c}^{t}$ by the flow averages and dispersions requires their calculation, which is a simple flow problem (it is shown in [MM 2003]). Supposing that this question is clarified, let's consider the problem behaviour in the case of linear approximations.

3. Linear Approximation

In the case mentioned, variation of the optimization criteria function coefficients is supposed to be changed by an expression $c_j(\lambda) = c_j^{t_0} + \lambda(c_j^t - c_j^{t_0})$, where λ varies in certain limits. The interval $[0,1]$ for λ is internal and characterizes the variation from $\tilde{c}^{t_0}$ to $\tilde{c}^t$, and the values $\lambda > 1$ are extrapolating the further behaviour of coefficients in a linear model. Let's denote $c_j^{\Delta} = c_j^t - c_j^{t_0}$.

So we are given the linear function

$$Z = \sum_{j=1}^{n} (c_j^{t_0} + \lambda c_j^{\Delta}) x_j \tag{1}$$

and the system of linear requirements, given by

$$\begin{aligned} &\sum_{i=1}^{n} a_{ij} x_j = b_i, \ i = 1,2,\cdots,m, \\ &x_j \geq 0, \qquad j = 1,2,\cdots,n. \end{aligned} \tag{2}$$

It is necessary to accompany the changes of λ, finding out in the interval $1 < \lambda$ the minimal value at which the change of the optimal plan takes place for the first time. Assume that the vector $\tilde{x}^t = \left(x_1^t, x_2^t, \cdots, x_n^t\right)$ which satisfies the system (2) introduces the corresponding new optimization basis.

According to the assumptions, we have optimal solution when $\lambda = 0$. Assume that the solution basis consists of the first m vectors of $\bar{a}_1, \cdots, \bar{a}_n$. In accord to the simplex algorithm and its optimization condition, all the "estimations" in this case must obey to the following condition: $z_j - c_j^{t_0} \leq 0, j = 1,2,\cdots,n$.

As $c_j(\lambda) = c_j^{t_0} + \lambda c_j^{\Delta}$ then that general optimization condition becomes:

$$z_j - c_j(\lambda) = (c_j^{t_0} + \lambda c_j^{\Delta}) x_j^0 - (c_j^{t_0} + \lambda c_j^{\Delta}) \leq 0, \ j = 1,2,\cdots,n.$$

Let's group the expression in the following way:

$$c_j^{t_0}(x_j^0 - 1) + \lambda c_j^{\Delta}(x_j^0 - 1) \leq 0, \ j = 1,2,\cdots,n,$$

and let introduce the notations: $\alpha_j = c_j^{t_0}(x_j^0 - 1)$ and $\beta_j = c_j^{\Delta}(x_j^0 - 1)$. The constants α_j and β_j are defined by the initial configuration: optimization criterion function coefficients and the corresponding solution basis, criterion current coefficients with the supposition that optimization did not change during that period.

For $\lambda = 0$ we have the optimization vertex $\tilde{x}^0$, and therefore, we get the following limitations: $\alpha_j = c_j^{t_0}(x_j^0 - 1) \leq 0$. The optimization vertex change does not take place when $0 \leq \lambda \leq 1$, so we get also: $\alpha_j + \lambda\beta_j = c_j^{t_0}(x_j^0 - 1) + \lambda c_j^{\Delta}(x_j^0 - 1) \leq 0$.

In particular, when $\lambda = 1$ we get $\alpha_j + \beta_j \leq 0$. The extreme condition will be written in the following general form: $\alpha_j + \lambda\beta_j \leq 0, j = 1,2,\cdots,n$. Let's find the minimal value of λ at which at least one of this inequalities violates for the first time.

Let's separate negative and positive cases of β_j. The restrictions on λ will accept the following forms:

$\lambda \geq -\alpha_j/\beta_j$ for all $\beta_j < 0$, and

$\lambda \leq -\alpha_j/\beta_j$ for all $\beta_j > 0$.

Let's introduce one more notation:

$$\bar{\lambda} = \left\{ \min\left(-\alpha_j/\beta_j\right) \ if \ a\,\beta_j > 0, and + \infty, \ when all \ \beta_j \leq 0 \right.$$

The optimal solution for $\lambda = 0$ coincides with optimal solutions for all λ which obeys the condition $0 \leq \lambda \leq \bar{\lambda}$. It is ensued that $\bar{\lambda}$ is the possible transition configuration. If $\bar{\lambda} = +\infty$ then there is no change of optimal plan. If $\bar{\lambda}$

is limited then it is necessary to consider two cases: the first one (a/) is the point $\bar{\lambda}$ with the possible equivalent optimal plans and possible continuations in this case, and the second one (b/): if there is a new optimal plan and if the problem has no solution at $\lambda > \bar{\lambda}$.

a/ Assume that $\bar{\lambda}$ is finite, i.e. $\bar{\lambda} = -\alpha_k / \beta_k$ for the corresponding value of parameter k. It means that $z_k - c_k(\lambda) = 0$ from which follows that the optimization plan is not single. Actually, let's insert the k-th vector into the basis and according to the simplex method let's exclude one of the vectors from the previous basis. We will get a new optimal plan the criterion value of which will stay unchanged. It follows even more – that, by all null estimations and by all basis modifications we can get many optimization equivalent vertexes and all elements of their linear closure also have the same discussing optimization value.

b/ In this case, we consider the values $\lambda > \bar{\lambda}$ and the $\bar{\lambda}$ is finite. If the coefficients of above mentioned k-th vector all not positive, i.e. $\tau_{ik} \le 0$, by optimization basis, then according to the simplex method, the criterion function becomes unlimited. This takes place any time when according to the increasing character of the criterion function we get the vector which is going to be involved into the basis $z_k - c_k(\lambda) > 0$, but it becomes clear that the vector has no positive $\tau_{ik} > 0$ coordinate because of we could exclude it from the basis. In this case, it is impossible to choose such a coefficient $\theta > 0$ that any $x_i - \theta\tau_{ik} = 0$ when $i = \{1, \cdots, m\}$. Therefore, we get the optimization plan with $m+1$ positive components; the set of $\bar{a}_1, \bar{a}_2, \cdots, \bar{a}_m, \bar{a}_k$ vectors are linearly depending and this corresponds to the non-angle vertex. Therefore, linear criterion function could not get to its minimal value. This means that hyper-plane which is defined by linear function could not become supporting hyper-plane of permissible polyhedron at any shift in the direction of gradient.

If a $\tau_{ik} > 0$ then the vector $\bar{a}_k$ is included into the basis and another vector $\bar{a}_l$ is excluded from it. As the new basis is constructed by the simplex method then it corresponds to a new optimal solution, and at those inequalities

(3) $$\alpha'_j + \lambda\beta'_j \le 0, \; j = 1,2,\cdots,n$$

are compatible.

Let's show that any $\lambda < \bar{\lambda}$ does not satisfy the system (3) of inequalities. Really, for the vector $\bar{a}_l$, excluded from the basis we will get the following:

(4) $$\alpha'_l = -\alpha_k / \tau_{lk}; \qquad \beta'_l = -\beta_k / \tau_{lk},$$

where $\tau_{lk} > 0$. Suppose that (3) takes place for any $\lambda < \bar{\lambda}$ then $\alpha'_j + \lambda\beta'_j \le 0$, or according to (4) $-\alpha_k - \lambda\beta_k \le 0$. As $\beta_k > 0$ then, from the latter inequality follows that $\lambda \ge -\alpha_k / \beta_k = \bar{\lambda}$.

4. Conclusion

The paper is devoted to the discussion of applied algorithms for data flows. The linear programming problems and the simplex algorithm of their solution were considered. This research is not about the simplex algorithm developments, but is about the approaches processed in this sphere that also help when according to the problem assumption the coefficients of criterion function variate in the result of the data flows analysis. We got that it is possible to introduce and develop the concepts and tools related to the simplex algorithm by approaches, which solve flow linear optimization problems. The core result is the construction of the extrapolation mechanism that applies linear extrapolation by predicting the stationary data. The concept of equivalency of optimal vertices is introduced, which helps to accompany the variation process preparing the possible optimization vertexes in advance.

This is important from the viewpoint of linear programming systems and optimization in applied data flow systems.

Bibliography

[AJ 2001], M. Ajtai, T. Jayram, R. Kumar, and D. Sivakumar. Counting inversions in a data stream. manuscript, 2001.

[AC 2003], Aslanyan L., Castellanos J., Mingo F., Sahakyan H., Ryazanov V., Algorithms for Data Flows, International Journal "Information Theories and Applications", ISSN 1310-0513, 2003, Volume 10, Number 3, pp. 279-282.

[AM 1996], N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In Proc. of the 1996 Annual ACM Symp. on Theory of Computing, pages 20-29, 1996.

[GL 1980], E.N. Gordeev and V.K. Leontiev, Stability in problems of narrow places, JCM&MP, No. 4, Moscow, 1980.

[MM 2003], Manoukyan T and Mirzoyan V., Image Sequences and Bit-Plane Differences. "Computer Science & Information Technologies Conference", Yerevan, September 22-26, pp. 284-286 (2003).

[GG 2002], A. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. Strauss. Fast, small-space algorithms for approximate histogram maintenance. In Proc. of the 2002 Annual ACM Symp. on Theory of Computing, 2002.

[V 1998], R.J. Vanderbei, Linear Programming; Foundations and Extensions, Kluwer Academic Publishers, Boston/London/Dordrecht, 1998.

Author's Information

Levon Aslanyan – Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia; e-mail: lasl@sci.am

MATRICIAL MODEL FOR THE STUDY OF LOWER BOUNDS

Jose Joaquin Erviti, Adriana Toni

Abstract: Let V be an array. The range query problem concerns the design of data structures for implementing the following operations. The operation update(j,x) has the effect $v_j \leftarrow v_j + x$, *and the query operation retrieve(i,j) returns the partial sum* $v_i + \ldots + v_j$. *These tasks are to be performed on-line. We define an algebraic model – based on the use of matrices – for the study of the problem. In this paper we establish as well a lower bound for the sum of the average complexity of both kinds of operations, and demonstrate that this lower bound is near optimal – in terms of asymptotic complexity.*

Keywords: zero-one matrices, lower bounds, matrix equations

ACM Classification Keywords: F.2.1 Numerical Algorithms and Problems

1 Introduction

Let V=(v_1 .. v_n) be an array of length n storing values from an arbitrary commutative semigroup S. We define the operations:

- retrieve(j,k): returns v_j+..+v_k $\quad \forall 1 \le j \le k \le n$
- update(j,x) : v_j := vj +x $\quad \forall 1 \le j \le n, \quad x \in S$ (1)

We refer to n as the size of the range query problem. We see that the complexity of executing an update(j,x) operation is constant meanwhile the worst complexity of a retrieve(i,j) operation is linear on n.