McMaster University

Advanced Optimization Laboratory

Title:

AdvOL

The use of low-rank updates in interior-point methods

Author:

E.D. Andersen, C. Roos, T. Terlaky, T. Trafalis and J.P. Warners

AdvOl-Report No. 2000/9
February 2000, Hamilton, Ontario, Canada

The use of low-rank updates in interior-point methods

E.D. Andersen, C. Roos,
Faculty of Information Technology and Systems
Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands
T. Terlaky[1],
Department Computing and Software
McMaster University,
1280 Main Street West, Hamilton, ON, Canada, L8S 4L7
T. Trafalis[2],
Department Industrial Engineering
Oklahoma State University, Oklahoma, USA
J.P. Warners[3]
Faculty of Information Technology and Systems
Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands

February 28, 2001

[1]The research was done while the third author worked at the the Department SSOR of the TU Delft.
[2]The major part of the research was done while the fourth author visited the the Department SSOR of the TU Delft. He gratefully acknowledges the financial support of the department.
[3]Supported by the Dutch Organization for Scientific Research (NWO) under grant SION 612–33–001.

Abstract

The most computational intensive part of interior point methods for solving various kinds of optimization problems is the solution of large linear systems. This note discusses several applications of linear systems with low-rank updates which arise in interior point methods and can efficiently be solved. It is also briefly discussed how this method is related to the widely used Sherman-Morrison formula. The relation with the normal equation and augmented system approach, used in interior point methods for linear optimization to calculate the search directions, is discussed. Further, applications in potential reduction methods, neural network training and conic quadratic optimization are discussed as well.

Key words. Linear programming, interior point methods, linear equation systems, positive definite matrix, rank-one update, Sherman-Morrison formula.

0.1 Introduction

When applying an interior point method to solve optimization problems, large linear equation systems like

$$Qx = q,$$

where Q is an $n \times n$ matrix and $q \in \mathbb{R}^n$ turn up regularly. The coefficient matrix Q in such linear systems is frequently sparse (less than 1% of the coefficients is nonzero). By exploiting the sparsity structure of Q one can solve such linear systems with hundreds of thousands or even millions of variables. As the density of the coefficient matrix increases, i.e. the percentage of nonzero coefficients increases, the efficiency of the solution procedures deteriorates. One possible way of getting a dense coefficient matrix is that the original matrix Q is modified by a rank one (or more generally by a low-rank update) [9, 8, 19]. Even a rank-one update by the all-one vector e results in the completely dense matrix $Q + ee^T$; without utilizing this special structure, large linear systems with such coefficient matrices would not be solvable. Dense matrices which are a sum of a sparse matrix plus some low-rank terms arise frequently in practice. In fact as has been observed by Fiacco and McCormick [4, p. 185], then the Hessian of many functions can easily be written as a sum of a diagonal matrix and some low-rank terms and they call such functions factorable. This implies if an optimization problem containing only factorable functions is solved using a Newton based method, then the coefficient matrix will contain low-rank terms. Moreover, the coefficient matrix can frequently be written as a sum of a sparse matrix plus a dense low-rank update. This for example happens when solving large linear optimization problems having dense columns using interior point methods [2, 3, 6, 13, 16]. Contrary to our approach, dense linear systems obtained by low rank updates are solved frequently by using preconditioned conjugate gradient methods [8].

This paper is organized as follows. In Section 0.2 the approach to solve linear systems with low-rank updates is reviewed; see also [7]. Section 0.2.3 discusses how the method relates to a clever use of the Sherman-Morrison [5] formula. In Section 0.3.1, the use of low rank updates in interior point algorithms to solve linear optimization problems is discussed. In particular, efficient handling of dense columns in the coefficient matrix of the linear optimization problem is considered. Finally, in Section 0.3 further applications are discussed.

0.2 Solving Linear Systems with Low-rank Updates

In this section we first review a method to solve linear systems with low rank updates. It can also be found in [7]. Subsequently, for completeness we verify its correctness. Finally, the relation with the well-known Sherman Morrison formula is made explicit.

0.2.1 The solution procedure

Large, typically sparse linear equation systems have to be solved in many applications of mathematics. Frequently the coefficient matrix of a linear system is a low-rank update of a non-singular (in many cases positive definite) matrix. Here we discuss the solution of such linear systems. To be concrete, let us consider the following system of linear equations with the unknown vector $x \in \mathbb{R}^n$,

$$(Q + RS^T)x = q, \tag{1}$$

where Q is an $n \times n$ matrix, R and S are $n \times k$ matrices and $q \in \mathbb{R}^n$ are given. We make the following assumptions.

Assumptions:

1. The matrix Q is nonsingular.
2. The rank of both R and S are equal to $k \leq n$.

3. The matrix $Q + RS^T$ is nonsingular.

4. The matrix Q is sparse, while RS^T is dense.

We will see later that Assumptions 1–4 guarantee that the system (1) is solvable efficiently.

As it is intuitively clear, solving a linear system

$$Qx = q, \tag{2}$$

where the sparse matrix Q is the coefficient matrix is computationally cheap, while directly solving equation system (1) with the dense coefficient matrix is computationally expensive. Our primary goal is to utilize the sparsity of the matrix Q. As the first step we reformulate (1) as follows

$$Qx = q - RS^T x = q - Ry.$$

For a moment neglecting the fact that the vector $y = S^T x$ depends on the unknown vector x we can decompose the solution process into the following steps.

Step 1. Determine x_0, the solution of the sparse system $Qx_0 = q$.

Step 2. Determine the solution of the sparse system $QU = R$, where U is an $n \times k$ matrix matrix of unknowns.

Step 3. Now we consider how to find an appropriate vector y. By the definition of y we have

$$y = S^T x = S^T(x_0 - Uy),$$

or equivalently, y is the solution of the system

$$(I + S^T U)y = S^T x_0. \tag{3}$$

Step 4. Let U be the $n \times k$ matrix with column vectors u_j. Then we have the solution

$$x = x_0 - Uy.$$

The above procedure involves the solution of $k + 1$ sparse linear systems (in Steps 1 and 2), all with the same sparse coefficient matrix Q, hence these can be solved easily.

In the next section we will verify that the k-dimensional linear equation system (3) has a unique solution and that in the way we just indicated, the linear equation system (1) can indeed be solved efficiently.

Before doing that, let us make a simple estimation of the computational complexity under Assumption 4. By a direct approach the linear system (1) with a dense coefficient matrix can be solved in $\mathcal{O}(n^3)$ arithmetic operations. Assume that the matrix Q is sparse and so the equation system $Qx_0 = q$ can be solved in $\rho < \mathcal{O}(n^3)$ arithmetic operations. We have to solve $k+1$ such systems and a small dense system with k unknowns. Thus the total complexity becomes $(k+1)\rho + \mathcal{O}(k^3)$. Note that, in many applications matrix Q is either (multi)diagonal, or block-diagonal with small diagonal blocks. As a consequence one has $\rho = \mathcal{O}(n)$. Indeed, in many of these cases it holds that $k \leq \mathcal{O}(\sqrt{n})$, so the total computational complexity becomes $\mathcal{O}(n\sqrt{n})$, which is a factor $n\sqrt{n}$ better than the direct approach.

0.2.2 Verification

As it is proposed above the linear system (1) is solved by solving $k + 1$ linear equation systems with the same coefficient matrix Q and a small $k \times k$ linear system as follows.

Let $x_0 \in \mathbb{R}^n$ be such that $Qx_0 = q$ and $u_j \in \mathbb{R}^n$ such that $Qu_j = r_j$ for $j = 1, \ldots, k$, where r_j denotes the jth column of matrix R and finally, let $y \in \mathbb{R}^k$ be such that $(I + S^T U)y = S^T x_0$. where I denotes the k-dimensional identity matrix and $U = [u_1, \ldots, u_k]$. We prove the following theorem.

Theorem 0.2.1 *If Assumptions 1–3 hold then the unique solution of the linear system (1) is given by*

$$x = x_0 - Uy.$$

Proof: First note that by Assumption 3 the solution x exists and it is unique. By Assumption 1, x_0 and the r_j are unique as well. Further, by Assumption 2 the vectors $r_j, \quad j = 1, \ldots, k$ are linearly independent and then, by Assumption 1, the solution vectors $u_j, \quad j = 1, \ldots, k$ are linearly independent as well, i.e. $\text{rank}(U) = k$. Now by proving that the coefficient matrix $I + S^T U$ is nonsingular we verify that y is unique as well. Assume to the contrary that there is a nonzero vector $w \in \mathbb{R}^k$ such that $(I + S^T U)w = 0$, or equivalently $S^T U w = -w$. Then, by multiplying the nonsingular matrix $Q + RS^T$ by the nonzero vector Uw and using $Qu_j = r_j$ we have

$$(Q + RS^T)Uw = (QU)w + R(S^T Uw) = Rw - Rw$$

yielding a contradiction.

Finally by simple calculations the reader may verify that $(Q + RS^T)(x_0 - Uy) = q$ which completes the proof.

0.2.3 The Sherman-Morrison Formula

There is an intimate relation between the solution procedure we reviewed and the well known Sherman-Morrison formula [5], given below. For a survey of the Sherman-Morrison formula we refer the reader to Hager [7].

Sherman-Morrison formula:

$$(Q + RS^T)^{-1} = Q^{-1} - Q^{-1}R(I + S^T Q^{-1} R)^{-1} S^T Q^{-1} \tag{4}$$

We can make two important observations.

- Having proved the correctness of our procedure the Sherman-Morrison formula (4) can be derived from our procedure. This can be done as follows. By definition – we have $x_0 = Q^{-1}q$, $U = Q^{-1}R$ and $y = (I + S^T U)^{-1} S^T x_0$ – the solution x is given by

  $$x = x_0 - Uy = Q^{-1}q - Q^{-1}R(I + S^T U)^{-1} S^T x_0 = (Q^{-1} - Q^{-1}R(I + S^T Q^{-1} R)^{-1} S^T Q^{-1})q.$$

  By Assumption 3 we can write $x = (Q + RS^T)^{-1} q$. Comparing this with the above expression and observing that Theorem 0.2.1 holds for all right-hand-side vector $q \in \mathbb{R}^n$ the Sherman-Morrison formula (4) follows.

- On the other hand, one can derive our procedure by carefully analyzing the Sherman-Morrison formula (4). Using again that the solution of (1) can be written as $x = (Q + RS^T)^{-1} q$ we have

  $$x = (Q^{-1} - Q^{-1}R(I + S^T Q^{-1} R)^{-1} S^T Q^{-1})q.$$

  Here all the "inverse matrix – vector" products have to be replaced by the solution of a linear equation system. Thus we have the expression $Q^{-1}q$ which is equivalent to x_0; the expression $Q^{-1}R$ which is equivalent to U; and finally, having these done, the expression

  $$(I + S^T Q^{-1} R)^{-1} S^T Q^{-1} q = (I + S^T U)^{-1} S^T x_0$$

  which is equivalent to z.

0.3 Applications in Interior Point Methods

0.3.1 Normal Equations Versus Augmented System in Linear Optimization

When solving linear programming problems with interior point methods one has to solve at each iteration [2, 3, 6], either the so-called *normal equation*

$$(AD^{-2}A^T)x = q \tag{5}$$

or the so-called *augmented system*

$$\begin{pmatrix} 0 & A \\ A^T & D^2 \end{pmatrix} \begin{pmatrix} x \\ x^0 \end{pmatrix} = \begin{pmatrix} -q \\ 0 \end{pmatrix}, \tag{6}$$

where matrix A is the $m \times n$ constraints coefficient matrix of the linear programming problem with $\mathrm{rank}(A) = m$, the matrix D is an appropriate n dimensional positive definite diagonal matrix and $q \in \mathbb{R}^n$. The exact values and definitions of D and q are specific for the different algorithmic variants. The reader easily verifies that the solutions x of both systems (5) and (6) are identical.

Early implementations of interior point methods were based on the normal equation approach [12]. People recognized that dense columns of the matrix A result in unacceptable fill-in in the coefficient matrix $AD^{-2}A^T$ of the normal equation. Let us have a closer look at this problem. For simplicity let us assume that the problem is scaled in such a way that $D = I$, and that all the dense columns are the last columns, thus matrix A can be partitioned as $A = [A_s, A_d]$ where A_s contains the sparse and A_d contains the dense columns. Now (5) gives

$$(A_s A_s^T + A_d A_d^T)x = q. \tag{7}$$

In the case $A_s A_s^T$ is nonsingular, this system is in the form (1) with $Q = A_s A_s^T$, hence can be solved efficiently. Thus our procedure described above can be used efficiently to handle dense columns in linear programming when solved by interior point methods. When $A_s A_s^T$ is singular a further trick will be needed.

Having a closer look at the augmented system (6) an interesting observation can be made. By substituting $A = [A_s, A_d]$ we have

$$\begin{pmatrix} 0 & A_s & A_d \\ A_s^T & I & 0 \\ A_d^T & 0 & I \end{pmatrix} \begin{pmatrix} x \\ x^1 \\ x^2 \end{pmatrix} = \begin{pmatrix} -q \\ 0 \\ 0 \end{pmatrix}. \tag{8}$$

By performing the pivots at the central unit matrix block we have

$$\begin{pmatrix} -A_s A_s^T & 0 & A_d \\ A_s^T & I & 0 \\ A_d^T & 0 & I \end{pmatrix} \begin{pmatrix} x \\ x^1 \\ x^2 \end{pmatrix} = \begin{pmatrix} -q \\ 0 \\ 0 \end{pmatrix}, \tag{9}$$

i.e. one has to solve the system

$$\begin{pmatrix} -A_s A_s^T & A_d \\ A_d^T & I \end{pmatrix} \begin{pmatrix} x \\ x^2 \end{pmatrix} = \begin{pmatrix} -q \\ 0 \end{pmatrix}. \tag{10}$$

The sparse part is nonsingular. If in (10) we would pivot on the unit matrix block we would get the dense normal equation (7). Instead, assuming (again, as above) that the matrix $Q = A_s A_s^T$ is nonsingular, we pivot on the left upper block which results in

$$\begin{pmatrix} I & -Q^{-1}A_d \\ 0 & I + A_d^T Q^{-1} A_d \end{pmatrix} \begin{pmatrix} x \\ x^2 \end{pmatrix} = \begin{pmatrix} Q^{-1}q \\ -A_d^T Q^{-1} q \end{pmatrix}. \tag{11}$$

Now the reader readily recognizes that the right hand side of (11) can be calculated by solving $Qx = q$ and one obtains $x_0 = Q^{-1}q$. Similarly, the matrix $U = Q^{-1}A_d$ can easily be calculated as well, and then clearly the equation $(I + A_d^T Q^{-1} A_d)x^2 = -A_d^T x_0$ is equivalent to solving $(I + A_d^T U)x^2 = -A_d^T x_0$. Hence our procedure can be interpreted as a specific implementation of the augmented system approach.[1]

The sparse part is singular. In the case $A_s A_s^T$ is singular, we cannot apply the above procedure. To handle this case efficiently a clever trick of K.D. Andersen [3] is needed. The solution process up to formula (10) can be performed independently of the (non)singularity of $A_s A_s^T$, but then, due to the singularity of $A_s A_s^T$ we cannot pivot on this block. To ensure the nonsingularity of this sparse block a nonsingular, possibly diagonal subsidiary matrix H is used. The reader easily verifies that the system

$$\begin{pmatrix} -A_s A_s^T - HH^T & A_d & H \\ A_d^T & I & 0 \\ H^T & 0 & I \end{pmatrix} \begin{pmatrix} x \\ x^2 \\ x^0 \end{pmatrix} = \begin{pmatrix} -q \\ 0 \\ 0 \end{pmatrix}, \tag{12}$$

is equivalent to the linear system (10), but here the sparse left-upper block is nonsingular, thus our procedure is applicable again.[2]

0.3.2 Karmarkar's Potential Function in Combinatorial Optimization

Let us consider the binary feasibility problem (BFP).

$$\text{(BFP)} \quad \text{find } x \in \{-1, 1\}^n \text{ such that } Ax \leq b.$$

Here $A \in \mathbb{R}^{t \times n}$, $b \in \mathbb{R}^t$. Problems in this format are also considered in [8, 9]. Karmarkar [8, 9] presented a potential reduction algorithm to solve problems of the form (BFP). After relaxing the integrality constraints to linear constraints $-e \leq x \leq e$ (here e denotes the all-one vector of length n), he proposed to optimize a nonconvex potential function in order to get integer solutions. Let us assume that an interior solution x (i.e. $Ax < b$, $-e < x < e$) is given. Then Karmarkar's potential function is given by

$$\phi(x) = \log \sqrt{n - x^T x} - \frac{1}{m} \sum_{j=1}^{m} s_j, \tag{13}$$

where the variables s_j are the slack variables associated with all the inequality constraints involved in the relaxed problem (note that $m := t + 2n$ and the slack variables are $s_j = b_j - a_j^T x, \quad 1 \leq j \leq t$; $s_j = 1 - x_{j-t}, \quad t+1 \leq j \leq t+n$; and $s_j = 1 + x_{j-(t+n)}, \quad t+n+1 \leq j \leq t+2n$). To minimize Karmarkar's potential function (13) a trust region algorithm is implemented. In this algorithm linear systems involving the Hessian of (13) as the coefficient matrix, are solved repeatedly. This Hessian matrix is given by

$$H_\phi(\mu) = -\frac{1}{f_0} I - \frac{2}{f_0^2} xx^T + (\frac{1}{m} + \mu) \overline{A}^T S^{-2} \overline{A},$$

where $f_0 = n - x^T x$, $\overline{A} = (A^T, I, -I)^T$ and S is the diagonal matrix of the slack variables (for more details see [8, 9, 19]). Due to the term xx^T the matrix H_ϕ is completely dense. Assuming that matrix A is sparse, the matrix $H_\phi(\mu)$ is a rank-one update of the sparse matrix $-\frac{1}{f_0} I + (\frac{1}{m} + \mu) \overline{A}^T S^{-2} \overline{A}$, hence the theory discussed in this paper is applicable to minimize Karmarkar's potential function efficiently.

[1] Observe that in actual implementations of the augmented system approach the last mentioned pivots on the block $-A_s A_s^T$ are not performed explicitly; instead, a Bunch-Parlett factorization is made. The augmented system approach with Bunch-Parlett factorization automatically does the job of efficiently handling dense columns in interior point methods.

[2] Note that if the linear optimization problem is preprocessed so that the problem is transformed to the canonical (inequalities and nonnegative variables) form, then this trick is not needed. In this case the left upper sparse block of the linear system has the form $-A_s A_s^T + \overline{D}$, where $\overline{D}$ is a positive diagonal matrix. Because this matrix is nonsingular, our procedure is applicable without further adjustment.

However, an additional requirement has to be satisfied; namely the parameter μ must be such that $H_\phi(\mu)$ is positive semidefinite, while at the same time the norm of the solution $x = (H_\phi(\mu))^{-1}h$ (here h denotes the right hand side of the linear system) must be sufficiently large. Note that this norm decreases when μ increases. Thus, it is not possible just to choose a large value for μ (to make sure that $H_\phi(\mu)$ is positive semidefinite). Now we show that we can use the rank-one update scheme and still check whether $H_\phi(\mu)$ is positive definite.

We have the following theorem.

Theorem 0.3.1 *Assume that $R \in \mathbb{R}^{n\times n}$ is symmetric. Then $R - zz^T$ is positive definite if and only if R is positive definite and $z^T R^{-1} z < 1$.*

Proof: Using the well known Schur complement reformulation we know that $R - zz^T$ is positive definite if and only if the matrix

$$\begin{bmatrix} R & z \\ z^T & 1 \end{bmatrix}$$

is positive definite. Thus, R must be positive definite and it easily follows that $z^T R z < 1$, or equivalently, $z^T R^{-1} z < 1$ if and only if $R - zz^T$ is positive definite.

Hence we obtain the following scheme to solve the system

$$(R - zz^T)x = q, \tag{14}$$

with the additional condition that $R - zz^T$ be positive semidefinite.

1. Determine the Cholesky factor U of R. If during this process R is discovered to be non positive definite, stop.
2. With the help of U, solve $Rx_0 = z$. If $z^T R^{-1} z = z^T x_0 \geq 1$, $R - zz^T$ is not positive semidefinite, so stop.
3. Solve $Rx_1 = q$, again using U.
4. Let $\alpha := \frac{x_1^T z}{1 - x_0^T z}$. Then $x = x_1 + \alpha x_0$ solves system (14).

This technique is used in the computational results reported in [19].

0.3.3 An application in unsupervised neural network training

In unsupervised neural network training, the objective is to categorize or discover features or regularities in a given set of training data. An approach proposed by Linsker [10] is to maximize the output variance. This gives rise to a nonconvex quadratic optimization problem of the following form.

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}(C + ee^T)x + c^T x \\ \text{subject to} & 0 \leq x \leq e. \end{array}$$

Here the matrix C is related to the *auto correlation* matrix which is (in this case) negative definite. In [17] a potential reduction approach to solve the above problem is developed. This involves solving linear systems with the matrix $C + ee^T$ as a coefficient matrix. Obviously, if C is sparse, the technique described in this paper can be used to speedup the computations.

0.3.4 Markowitz portfolio model

The well-known Markowitz portfolio problem can be modelled by the following convex quadratic optimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}||Fx||^2 + c^T x \\ \text{subject to} & Ax = b, \\ & x \geq 0, \end{array} \tag{15}$$

where A is either just the all-one vector ($A = e^T$) or a matrix with significantly less rows than columns. F is a $k \times n$ dense matrix, where in practice k quite frequently is much smaller than n. Now if (15) is solved using one of the popular interior point methods [15], then in each iteration a linear equation system having the coefficient matrix

$$A(D + F^T F)^{-1} A^T$$

should be solved. In this case D is a positive definite diagonal matrix. One possible way of solving (0.3.4) is of course to form the inverse of $(D + F^T F)$ explicitly. However, the technique presented in Section 0.2 is directly applicable which in this case leads to computing the inverse (or a Cholesky factorization) of a $k \times k$ matrix instead of a $n \times n$ matrix. Given k is much smaller than n a large computational saving is realized. It should be noted that Fiacco and McCormick [4, p. 186] and later Vanderbei and Carpenter [18, p. 22] have noted that this particular structure in the portfolio optimization problem can be exploited to speed up the computations.

0.3.5 Conic quadratic optimization

Recently, so-called conic quadratic optimization or second order cone optimization has received much attention due to the wide applicability of the model, see [11]. A conic quadratic optimization problem can be stated as follows

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{p} (c^i)^T x^i \\ \text{subject to} & \sum_{i=1}^{p} A^i x^i = b, \\ & x^i \in K^i, \quad \text{for} \quad i = 1, \ldots, p, \end{array} \tag{16}$$

where

$$x^i = \begin{bmatrix} x_0^i \\ x_1^i \end{bmatrix} \in R^{n_i} \quad \text{and} \quad K^i = \{(x_0; x_1) \in R^{n_i} : \quad x_0 \geq ||x_1||\} \tag{17}$$

is known as the Lorentz cone, the second order cone, or simply the ice-cream cone. These problems can be solved efficiently using either a primal, a dual, or a primal-dual interior point method, see [14, 20] for details. These three algorithms have in common that in each iteration a system of linear equations having the coefficient matrix

$$\sum_{i=1}^{p} A^i H^i A^i \tag{18}$$

should be solved where

$$H^i = \alpha^i I + (v^i)(v^i)^T - 2\alpha^i e_1^i (e_1^i)^T. \tag{19}$$

Clearly H^i is symmetric and it is known H^i is positive definite. α^i is a positive scalar, v^i is an arbitrary possible dense vector and finally e_1^i is the first unit vector. For simplicity we assume I, v^i, and e_1^i are all of an appropriate dimension. H^i is the inverse of the Hessian of the barrier related to the ith cone K^i.

Clearly, H^i is essentially identical to the identity matrix plus a rank 2 term. Therefore, (18) is identical to

$$\sum_{i=1}^{p} A^i H^i (A^i)^T = \sum_{i=1}^{p} \alpha^i A^i (A^i)^T + \sum_{i=1}^{p} (A^i v^i)(A^i v^i)^T - \sum_{i=1}^{p} 2\alpha^i (A^i e_1^i)(A^i e_1^i)^T.$$

It can be assumed without loss of generality, that

$$A = \begin{bmatrix} A^1 & A^2 & \dots & A^p \end{bmatrix}$$

is of full row rank. This implies that

$$\sum_{i=1}^{p} \alpha^i A^i (A^i)^T \tag{20}$$

is a positive definite matrix. Note that the matrix (20) has the same sparsity pattern as AA^T which implies that if A is a sparse matrix without dense columns, then AA^T is likely to be a sparse matrix.

In the case the matrices H^i is of a fairly small dimension say less than 10 or A contains few rows or is dense, then the most efficient solution method is based on computing a Cholesky factorization of (18) directly. However, in the case H^i has a fairly large dimension and the matrix A contains many rows, then it is likely to be more efficient to compute an inverse (or a Cholesky factorization) of (20) and then use the low-rank technique to handle the low-rank terms $\sum_{i=1}^{p}((A^i v^i)(A^i v^i)^T + (A^i e_1^i)(A^i e_1^i)^T)$. In particular this is advantageous if p is small compared to the total number of variables in the problem under consideration. This may for example be the case in robust linear least squares problems [11, p. 13].

Obviously it also possible to use a hybrid strategy based on handling the small dimensional blocks directly and the large dimensional blocks by the low-rank update technique.

We conclude this section by mentioning that similar observations were recently made by Alizadeh and Schmieta [1].

Bibliography

[1] F. Alizadeh and S.H. Schmieta. Optimization with semidefinite, quadratic and linear constraints. Technical Report RRR 23–97, 1997.

[2] E. D. Andersen, J. Gondzio, C. Mészáros, and X. Xu. Implementation of interior point methods for large scale linear programming. In T. Terlaky, editor, *Interior-point methods of mathematical programming*, pages 189–252. Kluwer Academic Publishers, 1996.

[3] K. D. Andersen. A Modified Schur Complement Method for Handling Dense Columns in Interior-Point Methods for Linear Programming. *ACM Trans. Math. Software*, 22(3):348–356, 1996.

[4] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley and Sons, New York, 1968.

[5] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore and London, 2 edition, 1989.

[6] J. Gondzio and T. Terlaky. A computational view of interior point methods for linear programming. In J. Beasley, editor, *Advances in linear and integer programming*. Oxford University Press, Oxford, England, 1995.

[7] W. W. Hager. Updating the inverse of a matrix. *SIAM Rev.*, 31(2):221–239, June 1989.

[8] A.P. Kamath, N.K. Karmarkar, K.G. Ramakrishnan, and M.G.C. Resende. Computational experience with an interior point algorithm on the satisfiability problem. *Annals of Operations Research*, 25:43–58, 1990.

[9] N. Karmarkar, M.G.C. Resende, and K.G. Ramakrishnan. An interior point algorithm to solve computationally difficult set covering problems. *Mathematical Programming*, 52:597–618, 1991.

[10] R. Linsker. Self-organization in a perceptual network. *Computer IEEE*, pages 105–117, 1988.

[11] M. S. Lobo, L. Vanderberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. Technical report, Information Systems Laboratory, Electrical Engineering Department, Stanford University, Standford, California 94305-4022, USA, January 1998. To appear in Linear Algebra and its Applications.

[12] R. E. Marsten, M. J. Saltzman, D. F. Shanno, G. S. Pierce, and J. F. Ballintijn. Implementation of a dual affine interior point algorithm for linear programming. *ORSA J. on Comput.*, 1(4):287–297, 1989.

[13] K. A. McShane, C. L. Monma, and D. F. Shanno. An implementation of a primal-dual method for linear programming. *ORSA J. on Comput.*, 1(2):70–83, 1989.

[14] Yu. Nesterov and J.-Ph. Vial. Homogeneous analytic center cutting plane methods for convex problems and variational inequalities. Technical Report 1997.4, Logilab, HEC Geneva, Section of Management Studies, University of Geneva, jul 1997.

[15] C. Roos, T. Terlaky, and J. -Ph. Vial. *Theory and algorithms for linear optimization: an interior point approach*. John Wiley and Sons, New York, 1997.

[16] M. A. Saunders and J. A. Tomlin. Solving regularized linear programs using barrier methods and KKT systems. Technical Report SOL 96-4, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Standford, California 94305-4022, USA, December 1996.

[17] T. Trafalis, T. Terlaky, J.P. Warners, A.J. Quist, and C. Roos. Unsupervised neural network training via a potential reduction approach. Technical Report 96–172, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, The Netherlands, 1996.

[18] R. J. Vanderbei and T. J. Carpenter. Symmetric indefinite systems for interior point methods. *Math. Programming*, 58:1–32, 1993.

[19] J.P. Warners, T. Terlaky, C. Roos, and B. Jansen. Potential reduction algorithms for structured combinatorial optimization problems. *Operations Research Letters*, 21:55–64, 1997.

[20] G. Xue and Y. Ye. An efficient algorithm for minimizing a sum of euclidean norms with apllications. *SIAM J. on Optim.*, 7(4):1017–1036, 1997.