CORRELATIVE SPARSITY IN SOLVING OPTIMIZATION PROBLEMS

Sunyoung Kim [1]

1) *Department of Mathematics, Ewha University, Seoul 120-750, KOREA*

Corresponding Author : Sunyoung Kim, skim@ewha.ac.kr

ABSTRACT

Exploiting sparsity has been a key issue in solving large-scale optimization pro blems. The most time-consuming part of primal-dual interior-point methods for linear programs, second-order cone programs, and semidefinite programs is solving the Schur complement equation at each iteration, usually by the Cholesky factorization. The computational efficiency is greatly affected by the sparsity of the coefficient matrix of the equation that is determined by the sparsity of an optimization problem (linear program, semidefinite program or second-order program). We show if an optimization problem is *correlatively sparse*, then the coefficient matrix of the Schur complement equation inherits the sparsity, and a sparse Cholesky factorization applied to the matrix results in no fill-in.

INTRODUCTION

Primal-dual interior-point methods are shown to be numerically robust for solving linear programs (LPs), semidefinite programs (SDPs) and second-order cone programs (SOCPs). For large-scale LPs, many free and commercial software packages implementing primal-dual interior-point methods have been established as very efficient and powerful solvers. Challenges still remain in solving large-scale SDPs and SOCPs, although several successful software packages have been developed.

Efficient handling of large-scale LPs, SDPs and SOCPs in the implementation of primal-dual interior-point methods have taken two significant approaches: solving the Schur complement equation (or larger systems that induce the Schur complement equation [14,15]) efficiently and exploiting the sparsity of the problems. In the case of LP, these two issues have been studied widely and various practical techniques have been implemented in free and commercial software packages. For LPs and SOCPs, the sparsity of the Schur complement matrix (the coefficient matrix of the Schur complement equation) was exploited by splitting the matrix into sparse and dense parts, factorizing the sparse part, and applying low-rank update to the dense part [2,5,12]. For the case of SDP, the importance of exploiting the sparsity of the data matrices was recognized in [3], which proposed three types of methods for computing the elements of the Schur complement matrix depending on their sparsity. For large-scale SDPs, solving the Schur complement equation using iterative methods was proposed in [13,14]. The aggregated sparsity pattern of all data matrices of SDPs was exploited for the primal-dual interior-point methods [4,10]. The current paper adopts some of basic ideas such as a chordal graph structure of the sparsity pattern used there. The aggregated sparsity, however, does not necessarily imply the sparsity of the Schur complement matrix. The issue of an efficient solution to the Schur complement equation was not addressed there. Instead, the focus was on an efficient handling of the primal matrix variable that becomes dense in general even when the aggregated sparsity pattern is sparse.

Sparsity can be used in various ways depending on optimization problems. The sparsity from partially separable functions was used in connection with efficient implementation of quasi-Newton methods for solving large-scale unconstrained optimization [6]. The correlative sparsity was introduced to handle the sparsity of polynomial optimization problems (POPs) [16], as a special case of sparsity described in [7,9]. The relationship between the partial separability and the correlative sparsity was discussed in the recent paper [8]. In the primal-dual interior-point methods, exploiting the sparsity of the Schur complement matrix becomes important for efficiency because the Cholesky factorization is commonly used for the solution of the Schur complement equation. We note that many fill-ins may occur after applying a sparse Cholesky factorization to a general non-structured sparse matrix. The sparsity of the Schur complement matrix is determined by the sparsity of the data matrices of an optimization problem (LP, SDP or SOCP). Our motivation is based on finding a sparsity condition on SDPs and SOCPs that can lead to a sparse Cholesky factorization of the Schur complement matrix with no fill-in. We show that the correlative sparsity is indeed such a sparsity condition that provides a sparse Cholesky factorization of the Schur complement matrix with no fill-in. We also propose a correlatively-sparse linear optimization problem (LOP) for a unified representation of correlatively-sparse LPs, SDPs and SOCPs.

We introduce the correlative sparsity, which was originally proposed for a POP [16], for an LOP with a linear objective function in an n-dimensional variable vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and inequality (linear matrix inequality, second-order cone inequality) constraints in $\boldsymbol{x}$. This LOP is so called the dual inequality standard form LOP unifying LPs, SDPs and SOCPs. The correlative sparsity of the LOP is defined by an $n \times n$ symmetric matrix $\boldsymbol{R}$, called the correlative sparsity pattern (csp) matrix, as follows. Each element R_{ij} of the csp matrix $\boldsymbol{R}$ is either 0 or $\star$ for a nonzero value. The symbol $\star$ was assigned to all diagonal elements of $\boldsymbol{R}$ and also to each off-diagonal element $R_{ij} = R_{ji}$ $(1 \leq i < j \leq n)$ if and only if the variables x_i and x_j appear simultaneously in a linear inequality (linear matrix inequality, second-order cone inequality) constraint. If the csp matrix $\boldsymbol{R}$ allows a symbolic sparse Cholesky factorization with no fill-in (under an appropriate simultaneous reordering of its rows and columns), we say that the LOP is correlatively-sparse. The objective of the paper is to show that if the LOP satisfies the correlative sparsity, then the sparsity pattern of the Schur complement matrix coincides with the csp matrix $\boldsymbol{R}$. This guarantees a sparse Cholesky factorization of the Schur complement matrix with no fill-in.

Although our major concern is a correlatively-sparse LOP, we deal with an almost-correlatively-sparse LOP, a slightly more practical LOP with a small-sized dense linear inequality (linear matrix inequality, second-order cone inequality) constraint and sparse constraints inducing a csp matrix $\boldsymbol{R}$. With this form of LOP, the correlatively-sparse LOP and the almost-correlatively-sparse LOP can be dealt with simultaneously because all the results for the correlatively-sparse LOP can be obtained by simply neglecting the dense constraint. The Schur complement matrix of the almost-correlatively-sparse LOP is dense in general. Its sparsity, however, can be exploited by splitting the matrix into two parts, the sparse part with the same sparsity pattern as the csp matrix $\boldsymbol{R}$ and the dense part of low-rank. A sparse Cholesky factorization can be used for the sparse part, and the well-known Sherman-Morrison-Woodbury formula for the dense part. This technique of splitting sparse and dense parts of the Schur complement matrix was used in an affine scaling variant of Karmarkar's algorithm for linear programs [1]. See also [2,5,12].

We also examine the link between the correlative sparsity of a POP and the sparsity of its SDP relaxation. When the sparse SDP relaxation ([16]) is applied to a POP satisfying the correlative sparsity, an equivalent polynomial SDP satisfying the same correlative sparsity is constructed as an intermediate optimization problem. It is then linearized to an SDP relaxation problem. We prove that the correlative sparsity of the POP is further maintained in the SDP relaxation problem. It was also observed through the numerical experiment in [16] that “the sparse SDP relaxation for a correlatively-sparse POP leads to an SDP that can maintain the sparsity for primal-dual interior-point methods”.

REFERENCES

1. I. Adler, M. G. C. Resende, G. Veiga and N. Karmarkar, “An implementation of Karmarkar’s algorithm for linear programming”, *Mathematical Programming* **44** (1989) 297–335.
2. K. D. Anderson, “A modified Schur complement method for handling dense columns in interior-point methods for linear programming”, *ACM Transactions on Mathematical Software* **22** (1996) 348–356.
3. K. Fujisawa, M. Kojima, K. Nakata “Exploiting sparsity in primal-dual Interior-point methods for Semidefinite Programming”, *Mathematical Programming* **79** (1997) 235–254.
4. M. Fukuda, M. Kojima, K. Murota and K. Nakata, “Exploiting sparsity in semidefinite programming via matrix completion I: general framework”, *SIAM Journal on Optimization* **11** (2001) 647–674.
5. D. Goldfarb, K. Scheinberg, “Product-form Cholesky factorization in interior-point methods for second order cone programming”, *Mathematical Programming* **103** (2005) 153–179.
6. A. Griewank and Ph. L. Toint, “On the unconstrained optimization of partially separable functions”, in M. J. D. Powell, ed., *Nonlinear Optimization 1981*, Academic Press, New York, 1982, pp.301–312.
7. S. Kim, M. Kojima and H. Waki, “Generalized Lagrangian duals and sums of squares relaxations of sparse polynomial optimization problems”, *SIAM Journal on Optimization* **15** (2005) 697–719.
8. S. Kim, M. Kojima, and Ph. L. Toint, “Recognizing underlying sparsity in optimization”, Research Report B-428, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152-8552, Japan, May 2006.
9. M. Kojima, S. Kim and H. Waki, “Sparsity in sums of squares of polynomials”, *Mathematical Programming* **103** (2005) 45–62.
10. K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima and K. Murota, “Exploiting sparsity in semidefinite programming via matrix completion II: implementation and numerical results”, *Mathematical Programming* **95** (2003) 303–327.
11. Yu. E. Nesterov and M. J. Todd, “Primal-dual interior-point for self-scaled cones”, *SIAM Journal on Optimization* **8** (1988) 324-364.
12. F. J. Sturm “Implementation of interior point methods for mixed semidefinite and second order cone optimization”, *Optimization Methods and Software* **17** (2002) 1105–1154.

13. K.C. Toh and M. Kojima, “Solving some large scale semidefinite programs via the conjugate residual method’, *SIAM Journal on Optimization* **12** (2002) 669–691.

14. K. C. Toh, “Solving large scale semidefinite programs via an iterative solver on the augmented systems”, *SIAM Journal on Optimization* **14** (2004) 670–698.

15. R. J. Vanderbei, “Symmetric quasidefinite matrices”, *SIAM Journal on Optimization* **5** (1995) 100–113.

16. H. Waki, S. Kim, M. Kojima and M. Muramatsu, “Sums of squares and semidefinite programming relaxations for polynomial optimization problems with structured sparsity” , *SIAM Journal on Optimization* **17** (2006) 218–242.