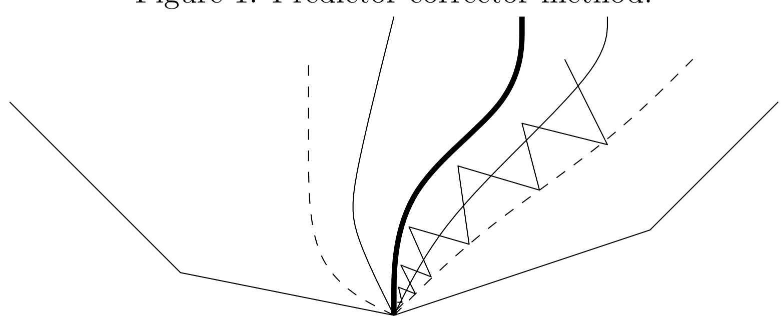
| | |
|---|---|
| **Semidefinite Programming** | **Lecture 21** |
| **OR 6327 Spring 2012** | **April 10, 2012** |
| **Scribe: Dmitriy Drusvyatskiy** | |

We will start by discussing methods that maintain a polynomial bound while hoping for better performance in practice.

The first idea is due to Monteiro and Adler (for LP problems). Compute $(\triangle X, \triangle y, \triangle S)$ as an affine function of $\sigma$ and then choose the smallest $0 \leq \sigma \leq 1$ satisfying $(X + \triangle X(\sigma), y + \triangle y(\sigma), S + \triangle S(\sigma)) \in \mathcal{N}_F(\beta)$. From our analysis, we have the upper bound $\sigma \leq 1 - \frac{\delta}{\sqrt{n}}$ but it may be much smaller.

The second idea is to use a predictor-corrector algorithm. Choose $0 < \beta < \frac{1}{2\sqrt{2}}$ and $(X_0, y_0, S_0) \in \mathcal{N}_F(\beta)$. At iteration $k$, with $k$ even, set $\sigma = 0$ and $\alpha$ as large as possible so that $(X_{k+1}, y_{k+1}, S_{k+1}) := (X_k, y_k, S_k) + \alpha(\triangle X, \triangle y, \triangle S) \in \mathcal{N}_F(2\beta)$. At iteration $k$, with $k$ odd, choose $\sigma = 1$ and then set $(X_{k+1}, y_{k+1}, S_{k+1}) := (X_k, y_k, S_k) + (\triangle X, \triangle y, \triangle S)$.

Figure 1: Predictor-corrector method.



Using arguments similar to those for the short-step algorithm, we can show that, for $\beta = 1/20$,

$$(X_k, y_k, S_k) \in \begin{cases} \mathcal{N}_F(\beta) & \text{if } k \text{ is even,} \\ \mathcal{N}_F(2\beta) & \text{if } k \text{ is odd,} \end{cases}$$

and

$$X_{k+1} \bullet S_{k+1} \leq \begin{cases} 1 - \frac{1}{7\sqrt{n}} X_k \bullet S_k & \text{if } k \text{ is even,} \\ X_k \bullet S_k & \text{if } k \text{ is odd.} \end{cases}$$

Alternatively, we may consider a long-step algorithm. Choose $0 < \beta < 1$ and $(X_0, y_0, S_0) \in \mathcal{N}_{-\infty}(\beta)$. Also choose $0 < \sigma < 1$ (independent of $n$). At iteration $k$, choose a scaling matrix $P = P_k$ so that $P_k X_k P_k^T$ and $P_k^{-T} S_k P_k^{-1}$ commute. Then compute the corresponding directions. Let

$$(X_{k+1}, S_{k+1}, S_{k+1}) = (X_k, S_k, S_k) + \alpha(\triangle X, \triangle y, \triangle S),$$

with $\alpha$ as large as possible so we remain in $\mathcal{N}_{-\infty}(\beta)$.

**Theorem 1** *The long-step algorithm terminates in $O(n^{\frac{3}{2}} \ln \frac{1}{\epsilon})$ iterations for the HKM and dual HKM directions, and in $O(n \ln \frac{1}{\epsilon})$ iterations for the NT direction.*

For the proof, see the paper of Monteiro and Zhang in the references page from the course website. Question 1 of Homework 4 asks you to establish that the directions are well-defined at any iterate as long as the commuting condition holds.

Even though the polynomial bound above is worse than for the short-step algorithm, the long-step method substantially outperforms all variants of the short-step method in practice.

We now consider the amount of computational work in each iteration. The main expense in every iteration is computing the search direction, and in particular, in solving

$$(\mathcal{A}\mathcal{E}^{-1}\mathcal{F}\mathcal{A}^*)\triangle y = -\mathcal{A}\mathcal{E}^{-1}R_{EF}. \tag{1}$$

Note that there are smaller, but significant, costs in calculating $\triangle X$ and $\triangle S$, and possibly in some Cholesky/eigenvalue decompositions ($O(n^3)$). Getting back to (1), we need to compute the $m \times m$ matrix in (1) whose $(i, j)$ entry is $A_i \bullet (\mathcal{E}^{-1}\mathcal{F}A_j)$.

For the AHO direction, this entry is $A_i \bullet ((S \odot I)^{-1}(X \odot I)A_j)$. Computing this expression requires an eigenvalue decomposition of $S$. A similar situation occurs for the dual HKM.

For the HKM direction, this entry is $A_i \bullet (XA_jS^{-1})$, whereas for the NT direction, this is $A_i \bullet (WA_jW)$. Computing $W$ in turn requires an eigenvalue decomposition. (Question 3 on Homework 4 asks you to show that only one such decomposition is needed, even though at first sight it seems two are required.)

Hence the total work involved is

$$O(m^2n^2) + O(mn^3) + O(m^3) + O(n^3).$$

Here $m$ can be $\Omega(n^2)$. Here the first term comes from computing $A_i \bullet B_j$ for all $i, j$ (or all $i \le j$), the second comes from computing the $B_j = XA_jS^{-1}$ or $WA_jW$, and the third from solving the system for $\Delta y$. Note that for the AHO direction, the $m \times m$ matrix is not symmetric, so the constants are larger.

For special situations, computations can be streamlined, e.g. in the MAX-CUT problem, where $A_i = e_i{e_i}^T$, the entry in question is $X_{i,j}(S^{-1})_{i,j}$ for HKM and $(W_{i,j})^2$ for NT. If all the matrices $A_i$ have a common sparsity pattern, then we only need entries of $XA_jS^{-1}$ or $WA_jW$ lying in the sparsity pattern (Fujisawa et al. on the references page). If all the matrices $A_i$ and $C$ are block-diagonal , so are $S$, $S^{-1}$, $A$ and $W$, and all the computations can be done block-wise.

Finally for the dual method, we need to compute a matrix with entries of the form $A_i \bullet (S^{-1}A_jS^{-1})$, and if we have common sparsity structure, then its Cholesky factor $L_S$ may be quite sparse, and then associating parenthesis as in

$$B_j = S^{-1}A_jS^{-1} = \left((L_S^{-T}(L_S^{-1}A_j))L_S^{-T}\right)L_S^{-1},$$

may save computational time.

**Matrix Completion**(Fukuda et al. on the references page)

We again assume that $C$ and all $A_i$ have a common sparsity structure. Suppose that if an $i, j$ entry is non-zero, then the inclusion $(i, j) \in E \subset \mathbb{N} \times \mathbb{N}$ holds, where $\mathbb{N} := \{1, 2, \ldots, n\}$.

Assume furthermore $(i, i) \in E$ for all $i \in \mathbb{N}$. Then all feasible dual slacks $S$ have the same sparsity structure, and only the entries of a primal iterate $X$ in $E$ contribute to its objective value and its feasibility in the linear constraints. Other entries have to be suitably chosen to ensure $X \succeq 0$. A partial symmetric matrix $\bar{X}$ has its entries in $E$ specified, but other entries are free. Specifying the other entries gives a *completion* of $\bar{X}$. We would like to find a unique "nice" positive definite completion of $\bar{X}$, whenever it has some positive definite completion.

**Theorem 2** *If $\bar{X}$ has a positive definite completion, then there is a unique solution $\widehat{X} = \widehat{X}(\bar{X})$ to the problem*

$$\min\{-\ln\det X : x_{i,j} + x_{j,i} = \bar{x}_{i,j} + \bar{x}_{j,i} \text{ for } (i,j) \in E\},$$

*characterized by $x_{i,j} = \bar{x}_{i,j}$ for $(i,j) \in E$ and $(X^{-1})_{i,j} = 0$ for $(i,j) \notin E$ .*

**Proof:** We will prove all the claims in the theorem, except for the existence. Uniqueness follows since $-\ln\det$ is strictly convex. The optimality conditions read

$$-X^{-1} + \sum_{(i,j)\in E} \lambda_{i,j}(e_i e_j^T + e_j e_i^T) = 0.$$

This implies $(X^{-1})_{i,j} = 0$ for each $(i,j) \notin E$. The converse holds by choosing $\lambda$ appropriately and observing that the condition above is both necessary and sufficient for optimality. $\square$