Supplementary Material of "HONES: A Fast and Error-free Homotopy Method For Online Newton Step"

A Roadmap of Appendices

The general idea of HONES algorithm has been presented in section 2. However, to implement it efficiently, we need much more effort to explore the structure of the solution path and find common quantities which are used by multiple sub-routines. To make the derivation well-organized, we start from considering the case where only one of A and r is time-varying while the other parameter is fixed. The case where A is time-varying and the case where r is time-varying are considered separately in Appendix B and C. Then in Appendix D, we combine two components and state the implementation for the general case.

In each following appendix, we will first define a list of case-specific intermediate variables, which are the key ingredients to improve efficiency. Then we describe the whole procedure followed by details of each sub-routine. Finally, we give a detailed complexity analysis at the end of each appendix.

B Implementation of HONES Algorithm With Time-Varying A and Fixed r

B-1 Intermediate Variables

Although (8)-(10) completely define the solution, they involves messy terms. To simplify the notations, we define three lists of intermediate variables. We should emphasize that these variables also play important roles in the algorithm design since they capture the quantities repeatly appeared and unnecessary computation can be avoided by storing their values in memory.

The intermediate variables are defined as follows. First, let M be a $n \times n$ matrix such that

$$M_{SS} = A_{SS}^{-1}, \quad M_{S^cS} = -A_{S^cS}A_{SS}^{-1}, \quad M_{\cdot,S^c} = 0. \tag{B-1}$$

For large-scale problem where n is prohibitively large, we can only store a $n \times |S|$ matrix by removing the zero entries of M. This saves storage cost significantly. Then we define two vectors $\eta, \tilde{\eta} \in \mathbb{R}^n$ such that

$$\eta_S = M_{SS}g_S, \quad \eta_{S^c} = g_{S^c} + M_{S^cS}g_S, \quad \tilde{\eta}_S = M_{SS}\mathbf{1}_S, \quad \tilde{\eta}_{S^c} = \mathbf{1}_{S^c} + M_{S^cS}\mathbf{1}_S \in \mathbb{R}^n. \tag{B-2}$$

Last we define four scalars.

$$D = \mathbf{1}_S^T A_{SS}^{-1}\mathbf{1}_S, \quad D_g = \mathbf{1}_S^T A_{SS}^{-1}g_S, \quad D_{gg} = g_S^T A_{SS}^{-1}g_S \quad D_{gr} = -\eta_S^T r_S. \tag{B-3}$$

Note that all variables are functions of λ if A is replaced by $A + \lambda gg^T$ and we denote them by $\cdot(\lambda)$. For example,

$$D(\lambda) = \mathbf{1}_S^T(A_{SS} + \lambda g_S g_S^T)^{-1}\mathbf{1}_S,$$

and others can be defined in a similar fashion. The following lemma formulates these functions.

Lemma B-1 *Let* $\alpha(\lambda) = \frac{\lambda}{1+\lambda D_{gg}}$*. Before any entry of* (x_S, μ_{S^c}) *hitting* 0*, it holds that*

- $M_{\cdot,S}(\lambda) = M_{\cdot,S} - \alpha(\lambda)\eta\eta_S^T$;

- $\eta(\lambda) = \frac{\eta}{1+\lambda D_{gg}}$;
- $\tilde{\eta}(\lambda) = \tilde{\eta} - \alpha(\lambda) D_g \eta$;
- $D(\lambda) = D - \alpha(\lambda) D_g^2$;
- $(D_g(\lambda), D_{gg}(\lambda), D_{gr}(\lambda)) = \frac{1}{1+\lambda D_{gg}}(D_g, D_{gg}, D_{gr})$.

Proof By Sherman-Morrison-Woodbury formula,

$$M_{SS}(\lambda) = (A_{SS} + \lambda g_S g_S^T)^{-1} = A_{SS}^{-1} - \lambda \frac{A_{SS}^{-1} g_S g_S^T A_{SS}^{-1}}{1 + \lambda g_S^T A_{SS}^{-1} g_S} = M_{SS} - \alpha(\lambda) \eta_S \eta_S^T.$$

This implies that

$$\begin{aligned}
M_{S^cS}(\lambda) &= -(A_{S^cS} + \lambda g_{S^c} g_S^T)(A_{SS}^{-1} - \alpha(\lambda)\eta_S\eta_S^T) \\
&= M_{S^cS} - \lambda g_{S^c} g_S^T A_{SS}^{-1} + \lambda\alpha(\lambda) g_{S^c} g_S^T \eta_S \eta_S^T + \alpha(\lambda) A_{S^cS}\eta_S\eta_S^T \\
&= M_{S^cS} - \lambda g_{S^c}\eta_S^T + \lambda\alpha(\lambda) D_{gg} g_{S^c}\eta_S^T + \alpha(\lambda) A_{S^cS}\eta_S\eta_S^T \qquad [\textit{Use } D_{gg} = g_S^T \eta_S] \\
&= M_{S^cS} - (\lambda - \lambda\alpha(\lambda) D_{gg}) g_{S^c}\eta_S^T + \alpha(\lambda) A_{S^cS}\eta_S\eta_S^T \\
&= M_{S^cS} - \alpha(\lambda) g_{S^c}\eta_S^T + \alpha(\lambda) A_{S^cS}\eta_S\eta_S^T \qquad [\textit{Use } \lambda - \lambda\alpha(\lambda) D_{gg} = \alpha(\lambda)] \\
&= M_{S^cS} - \alpha(\lambda)(g_{S^c} - A_{S^cS} A_{SS}^{-1} g_S)\eta_S^T \\
&= M_{S^cS} - \alpha(\lambda)\eta_{S^c}\eta_S^T.
\end{aligned}$$

Putting pieces together, we obtain that

$$M_{\cdot,S}(\lambda) = M_{\cdot,S} - \alpha(\lambda)\eta\eta_S^T.$$

Based on $M_{\cdot,S}(\lambda)$, it is straightforward to derive other variables. For $\eta(\lambda)$,

$$\begin{aligned}
\eta_S(\lambda) &= M_{SS}(\lambda) g_S = \eta_S - \alpha(\lambda)\eta_S\eta_S^T g_S \\
&= (1 - \alpha(\lambda) D_{gg})\eta_S = \frac{\eta_S}{1 + \lambda D_{gg}}; \\
\eta_{S^c}(\lambda) &= g_{S^c} + M_{S^cS}(\lambda) g_S = \eta_{S^c} - \alpha(\lambda)\eta_{S^c}\eta_S^T g_S \\
&= (1 - \alpha(\lambda) D_{gg})\eta_{S^c} = \frac{\eta_{S^c}}{1 + \lambda D_{gg}}.
\end{aligned}$$

Thus,

$$\eta(\lambda) = \frac{\eta}{1 + \lambda D_{gg}}.$$

Similarly,,

$$\begin{aligned}
\tilde{\eta}_S(\lambda) &= M_{SS}(\lambda)\mathbf{1}_S = \tilde{\eta}_S - \alpha(\lambda)\eta_S\eta_S^T\mathbf{1}_S \\
&= \tilde{\eta}_S - \alpha(\lambda) D_g \eta_S; \\
\tilde{\eta}_{S^c}(\lambda) &= \mathbf{1}_{S^c} + M_{S^cS}(\lambda)\mathbf{1}_S = \mathbf{1}_{S^c} - \alpha(\lambda)\eta_{S^c}\eta_S^T\mathbf{1}_S \\
&= \tilde{\eta}_{S^c} - \alpha(\lambda) D_g \eta_{S^c},
\end{aligned}$$

and hence

$$\tilde{\eta}(\lambda) = \tilde{\eta}_S - \alpha(\lambda) D_g \eta.$$

The last four scalars are even easier to handle. In fact, $D(\lambda)$ can be derived directly by

$$D(\lambda) = \mathbf{1}_S^T \left(M_{SS} - \alpha(\lambda)\eta_S\eta_S^T\right)\mathbf{1}_S = D - \alpha(\lambda) D_g^2$$

By reformulating the other three variables, the last statement can be proved,

$$\begin{aligned}
(D_g(\lambda), D_{gg}(\lambda), D_{gr}(\lambda)) &= (\mathbf{1}_S^T\eta_S(\lambda), g_S^T\eta_S(\lambda), -r_S^T\eta_S(\lambda)) \\
&= \frac{1}{1 + \lambda D_{gg}}(\mathbf{1}_S^T\eta_S, g_S^T\eta_S, -r_S^T\eta_S) \\
&= \frac{1}{1 + \lambda D_{gg}}(D_g, D_{gg}, D_{gr}).
\end{aligned}$$

■

B-2 Implementation

Lemma B-1 implies that given the function values of the intermediate variables at $\lambda = 0$, the function values at a neighborhood of 0 can be calculated directly. Within time t, all intermediate variables will be update correspondingly when the support changes. It has been shown in Theorem B-4 that updating M requires $n|S|$ operations while updating other variables only requires n operations. When the problem transfer from time t to time $t+1$, the variables $(\eta, D_g, D_{gg}, D_{gr})$ needs to be recalculated since it depends on a new $g^{(t+1)}$. In contrast, $(M, \tilde{\eta}, D)$ can be updated in the same way as in time t. In summary, $(M, \tilde{\eta}, D)$ is shared by for all times while $(\eta, D_g, D_{gg}, D_{gr})$ is only used in a single time. For compact notations, we define Par_1 and Par_2 as

$$\mathrm{Par}_1 = \{M, \tilde{\eta}, D\}, \quad \mathrm{Par}_2 = \{\eta, D_g, D_{gg}, D_{gr}\}. \tag{B-4}$$

In addition, we denote v by the concatenation of x_S and $-\mu_{S^c}$, i.e.

$$v_S = x_S, \quad v_{S^c} = -\mu_{S^c}, \tag{B-5}$$

as a $n \times 1$ vector. It will be shown in the next subsection that $v(\lambda)$ can be expressed in a concise way.

Algorithm 1 describes the full implementation of HONES algorithm, which solves the online problem (2) with r fixed. The sub-routines involved will be discussed separately in following subsections. Roughly speaking, after initialization, we enter into the outer-loop and try to solve (1) at time t using the information from time $t-1$. Starting from $\lambda = 0$, we search for the next λ that pushes one entry of v to zero. FIND_LAMBDA fulfills this goal and also reports the corresponding entry j. If $j \in S$ then j is removed from S and otherwise j is added into S. Since $(v, \mu_0, \mathrm{Par}_1, \mathrm{Par}_2)$ are all functions of λ, we update them by UPDATE_BY_LAMBDA, in which λ^{inc} denotes the increment to reach the next turning point from the current one. Unlike (v, μ_0), $(\mathrm{Par}_1, \mathrm{Par}_2)$ has discontinuity at each turning point λ due to the change of support S. They are updated by UPDATE_SHRINK_SUPPORT and UPDATE_EXPAND_SUPPORT depending on whether S is shrinked or expanded. The procedure is repeated until λ cross over 1 and an inner-loop finishes. At the end, Par_2 is recomputed for new $g^{(t+1)}$, which is achieved by DIRECT_UPDATE.

B-3 FIND_LAMBDA

With the help of intermediate variables, we can express (x_S, μ_{S^c}, μ_0) in a compact way.

Theorem B-4 *Before any entry of (x_S, μ_{S^c}) hitting 0, it holds that*

$$\mu_0(\lambda) = \mu_0 + \frac{\alpha(\lambda)}{D - \alpha(\lambda) D_g^2} \cdot D_g (D_g \mu_0 - D_{gr}). \tag{B-6}$$

and

$$v(\lambda) \triangleq \begin{pmatrix} x_S(\lambda) \\ -\mu_{S^c}(\lambda) \end{pmatrix} = v + \frac{\alpha(\lambda)}{D - \alpha(\lambda) D_g^2} \cdot (D_g \mu_0 - D_{gr}) \cdot (D_g \tilde{\eta} - D\eta), \tag{B-7}$$

Proof First we prove (B-6). By definition,

$$\mu_0(\lambda) = \frac{1 - \mathbf{1}_S^T A_{SS}(\lambda)^{-1} r_S}{\mathbf{1}_S^T A_{SS}(\lambda)^{-1} \mathbf{1}_S} = \frac{1 - \mathbf{1}_S^T M_{SS}(\lambda) r_S}{D(\lambda)}.$$

By Lemma B-1,

$$\begin{aligned} &- M_{SS}(\lambda) r_S = -(M_{SS} - \alpha(\lambda) \eta_S \eta_S^T) r_S \\ =& - M_{SS} r_S - \alpha(\lambda) \eta_S \eta_S^T r_S = -M_{SS} r_S + \alpha(\lambda) D_{gr} \eta_S. \end{aligned}$$

Thus the numerator of $\mu_0(\lambda)$ can be written as

$$1 - \mathbf{1}_S^T y_S + \mathbf{1}_S^T (M_{S^c S}^T y_{S^c} - M_{SS} r_S) - \alpha(\lambda) D_{gr} \mathbf{1}_S^T \eta_S = D\mu_0 - \alpha(\lambda) D_{gr} D_g.$$

The denominator of $\mu_0(\lambda)$, by Lemma B-1, is formulated as

$$D(\lambda) = D - \alpha(\lambda) D_g^2.$$

Algorithm 1 HONES Algorithm for time-varing A and fixed r

Inputs: Initial matrix $A^{(0)}$, vectors r, matrix-update-vectors $\{g^{(t)}, t = 1, 2, \ldots\}$.
Initialization:
$x \leftarrow$ as the optimum corresponding to $A^{(0)}$;
$S \leftarrow \mathrm{supp}(x)$;
Calculate (x, μ, μ_0) via (8)-(10)
$v_S \leftarrow x_S, v_{S^c} \leftarrow -\mu_{S^c}$;
Calculate intermediate variables $(\mathrm{Par}_1, \mathrm{Par}_2)$ via (B-1)-(B-3) with $g = g^{(1)}$.
Procedure:
for $t = 1, 2, \cdots$. **do**
  $\lambda \leftarrow 0$;
  while $\lambda < 1$ **do**
    $(\lambda^{\mathrm{inc}}, j, S^{\mathrm{new}}) \leftarrow \mathrm{FIND\_LAMBDA}(S, v, \mu_0; \mathrm{Par}_1, \mathrm{Par}_2)$;
    $\lambda^{\mathrm{inc}} \leftarrow \min\{\lambda^{\mathrm{inc}}, 1 - \lambda\}$;
    $\lambda \leftarrow \lambda + \lambda^{\mathrm{inc}}$;
    $(v, \mu_0; \mathrm{Par}_1, \mathrm{Par}_2) \leftarrow \mathrm{UPDATE\_BY\_LAMBDA}(\lambda^{\mathrm{inc}}; v, \mu_0; \mathrm{Par}_1, \mathrm{Par}_2)$;
    if $S^{\mathrm{new}} = S \cup \{j\}$ **then**
      $(\mathrm{Par}_1, \mathrm{Par}_2) \leftarrow \mathrm{UPDATE\_EXPAND\_SUPPORT}(\lambda, S, j; r, g^{(t)}, \mathrm{Par}_1, \mathrm{Par}_2)$;
    else if $S^{\mathrm{new}} = S \setminus \{j\}$ **then**
      $(\mathrm{Par}_1, \mathrm{Par}_2) \leftarrow \mathrm{UPDATE\_SHRINK\_SUPPORT}(S, j; r, g^{(t)}, \mathrm{Par}_1, \mathrm{Par}_2)$;
    end if
    $S \leftarrow S^{\mathrm{new}}$.
  end while
  $\mathrm{Par}_2 \leftarrow \mathrm{DIRECT\_UPDATE}(S, r, g^{(t+1)}; \mathrm{Par}_1, \mathrm{Par}_2)$;
  $A \leftarrow A + g^{(t)}(g^{(t)})^T$;
  $x_S^{(t)} \leftarrow x_S, \quad x_{S^c}^{(t)} \leftarrow 0$.
end for
Output: $x^{(1)}, x^{(2)}, \cdots$.

Putting the pieces together results in

$$\mu_0(\lambda) = \frac{D\mu_0 - \alpha(\lambda)D_{gr}D_g}{D - \alpha(\lambda)D_g^2} = \mu_0 + \frac{\alpha(\lambda)}{D - \alpha(\lambda)D_g^2} \cdot D_g(D_g\mu_0 - D_{gr}).$$

Plug $\mu_0(\lambda)$ into (8), we obtain that

$$\begin{aligned}
x_S(\lambda) &= \mu_0(\lambda)\tilde{\eta}_S(\lambda) + A_{SS}(\lambda)^{-1}r_S \\
&= A_{SS}^{-1}r_S + \alpha(\lambda)D_{gr}\eta_S + (\mu_0(\lambda) - \mu_0)\tilde{\eta}_S(\lambda) + \mu_0\tilde{\eta}_S(\lambda) \\
&= x_S + \alpha(\lambda)D_{gr}\eta_S + (\mu_0(\lambda) - \mu_0)\tilde{\eta}_S(\lambda) + \mu_0(\tilde{\eta}_S(\lambda) - \tilde{\eta}_S) \\
&= x_S + \alpha(\lambda)D_{gr}\eta_S + \frac{\alpha(\lambda)}{D - \alpha(\lambda)D_g^2} \cdot D_g(D_g\mu_0 - D_{gr})\tilde{\eta}_S(\lambda) - \mu_0\alpha(\lambda)D_g\eta_S \qquad [\textit{Use}\ \mathrm{LemmaB} - 1] \\
&= x_S + \frac{\alpha(\lambda)}{D - \alpha(\lambda)D_g^2} \cdot D_g(D_g\mu_0 - D_{gr})\tilde{\eta}_S(\lambda) - \alpha(\lambda)(D_g\mu_0 - D_{gr})\eta_S \\
&= x_S + \frac{\alpha(\lambda)}{D - \alpha(\lambda)D_g^2} \cdot (D_g\mu_0 - D_{gr}) \cdot (D_g\tilde{\eta}_S(\lambda) - (D - \alpha(\lambda)D_g^2)\eta_S) \\
&= x_S + \frac{\alpha(\lambda)}{D - \alpha(\lambda)D_g^2} \cdot (D_g\mu_0 - D_{gr}) \cdot \left(D_g\tilde{\eta}_S - \alpha(\lambda)D_g^2\eta_S - (D - \alpha(\lambda)D_g^2)\eta_S\right) \\
&= x_S + \frac{\alpha(\lambda)}{D - \alpha(\lambda)D_g^2} \cdot (D_g\mu_0 - D_{gr}) \cdot (D_g\tilde{\eta}_S - D\eta_S).
\end{aligned}$$

Similarly, it follows from (9) that

$$\begin{aligned}
-\mu_{S^c}(\lambda) &= \mu_0(\lambda)\tilde{\eta}_{S^c}(\lambda) + r_{S^c} + M_{S^cS}(\lambda)r_S \\
&= -\mu_{S^c} - \mu_0\tilde{\eta}_{S^c} + (M_{S^cS}(\lambda) - M_{S^cS})r_S + \mu_0(\lambda)\tilde{\eta}_{S^c}(\lambda)
\end{aligned}$$

$$
\begin{aligned}
&= -\mu_{S^c} - \mu_0 \tilde{\eta}_{S^c} + \alpha(\lambda) D_{gr} \eta_{S^c} + \mu_0(\lambda) \tilde{\eta}_{S^c}(\lambda) \\
&= -\mu_{S^c} + \alpha(\lambda) D_{gr} \eta_{S^c} + (\mu_0(\lambda) - \mu_0) \tilde{\eta}_{S^c}(\lambda) + \mu_0(\tilde{\eta}_{S^c}(\lambda) - \tilde{\eta}_{S^c}) \\
&= -\mu_{S^c} + \alpha(\lambda) D_{gr} \eta_{S^c} + \frac{\alpha(\lambda)}{D - \alpha(\lambda) D_g^2} \cdot D_g (D_g \mu_0 - D_{gr}) \tilde{\eta}_{S^c}(\lambda) - \mu_0 \alpha(\lambda) D_g \eta_{S^c} \\
&= -\mu_{S^c} + \frac{\alpha(\lambda)}{D - \alpha(\lambda) D_g^2} \cdot D_g (D_g \mu_0 - D_{gr}) \tilde{\eta}_{S^c}(\lambda) - \alpha(\lambda)(D_g \mu_0 - D_{gr}) \eta_{S^c} \\
&= -\mu_{S^c} + \frac{\alpha(\lambda)}{D - \alpha(\lambda) D_g^2} \cdot (D_g \mu_0 - D_{gr})(D_g \tilde{\eta}_{S^c}(\lambda) - (D - \alpha(\lambda) D_g^2) \eta_{S^c}) \\
&= -\mu_{S^c} + \frac{\alpha(\lambda)}{D - \alpha(\lambda) D_g^2} \cdot (D_g \mu_0 - D_{gr})(D_g \tilde{\eta}_{S^c} - \alpha(\lambda) D_g^2 \eta_{S^c} - (D - \alpha(\lambda) D_g^2) \eta_{S^c}) \\
&= -\mu_{S^c} + \frac{\alpha(\lambda)}{D - \alpha(\lambda) D_g^2} \cdot (D_g \mu_0 - D_{gr})(D_g \tilde{\eta}_{S^c} - D \eta_{S^c})
\end{aligned}
$$

In sum,

$$
\begin{pmatrix} x_S(\lambda) \\ -\mu_{S^c}(\lambda) \end{pmatrix} = \begin{pmatrix} x_S \\ -\mu_{S^c} \end{pmatrix} + \frac{\alpha(\lambda)}{D - \alpha(\lambda) D_g^2} \cdot (D_g \mu_0 - D_{gr}) \cdot (D_g \tilde{\eta} - D\eta).
$$

■

Theorem B-4 indicates that searching for next λ is equivalent to solve n linear equations. In fact, (B-7) can be abbreviated as

$$
v(\lambda) = v + \frac{\alpha(\lambda)}{D - \alpha(\lambda) D_g^2} u = \frac{Dv - (D_g^2 v - u)\alpha(\lambda)}{D - \alpha(\lambda) D_g^2},
$$

for $u = (D_g \mu_0 - D_{gr}) \cdot (D_g \tilde{\eta} - D\eta)$. Let

$$
\alpha = \min_+ \left\{ \frac{Dv_i}{D_g^2 v_i - u_i} : i = 1, 2, \dots, n \right\}
$$

where $\min_+(\Omega)$ denotes the minimum of all positive numbers contained in set Ω. Then the target λ is the solution of $\alpha(\lambda) = \alpha$, i.e.

$$
\lambda = \frac{\alpha}{1 - \alpha D_{gg}}.
$$

We should emphasize that the right-handed side might be negative if $\alpha D_{gg} \geq 1$ in which case v never hits 0. Thus, we should set λ to be infinity. The implementation of FIND_LAMBDA is stated in Algorithm 2

B-4 Variables Update

B-4.1 UPDATE_BY_LAMBDA

Once the next λ has been calculated, all variables can be updated via Lemma B-1 and Theorem B-4.

B-4.2 UPDATE_EXPAND_SUPPORT

Suppose S is updated to $S \cup \{j\}$ for some $j \in S^c$. Denote $\tilde{S}$ by $S \cup \{j\}$ and we add a supscript $+$ to each variable to denote the value after update. The key tool is the following formula showing the relation between matrix inverses after adding one row and one column.

Proposition 1 *Let* $\tilde{A}_{jj} = A_{jj} - A_{jS} A_{SS}^{-1} A_{Sj}$,

$$
A_{\tilde{S}\tilde{S}}^{-1} = \begin{pmatrix} A_{SS}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{\tilde{A}_{jj}} \cdot \begin{pmatrix} -A_{SS}^{-1} A_{Sj} \\ 1 \end{pmatrix} (-A_{jS} A_{SS}^{-1} \quad 1).
$$

Similar to section 4.1, the key is to update M and other variables are easy to update based on M. Denote a class of operator $\{\mathcal{R}_j : j \in S^c\}$ for matrix $W \in \mathbb{R}^{n \times n}$, $\mathcal{R}_j(W)$ sets the j-th row and j-th

Algorithm 2 FIND_LAMBDA

Input: Support S, iterate $v = \begin{pmatrix} x_S \\ -\mu_{S^c} \end{pmatrix}$, μ_0, intermediate variables $\mathrm{Par}_1, \mathrm{Par}_2$.

Procedure:

$u \leftarrow (D_g\mu_0 - D_{gr})(D_g\tilde{\eta} - D\eta)$;
$\alpha \leftarrow \min_+ \left\{ \frac{Dv_i}{D_g^2 v_i - u_i} : i = 1, 2, \ldots, n \right\}$;
$j \leftarrow \mathrm{argmin}_+ \left\{ \frac{Dv_i}{D_g^2 v_i - u_i} : i = 1, 2, \ldots, n \right\}$;
if $\alpha D_{gg} < 1$ **then**
    $\lambda^{\text{inc}} \leftarrow \frac{\alpha}{1 - \alpha D_{gg}}$;
else
    $\lambda^{\text{inc}} \leftarrow \infty$;
end if
if $j \in S$ **then**
    $S^{\text{new}} = S \setminus \{j\}$;
else
    $S^{\text{new}} = S \cup \{j\}$.
end if

Output: $\lambda^{\text{inc}}, j, S^{\text{new}}$.

Algorithm 3 UPDATE_BY_LAMBDA

Input: Increment λ^{inc}; iterate $v = \begin{pmatrix} x_S \\ -\mu_{S^c} \end{pmatrix}$, μ_0; intermediate variables $\mathrm{Par}_1, \mathrm{Par}_2$.

Procedure:

$\alpha_0 \leftarrow \frac{1}{1 + \lambda^{\text{inc}} \cdot D_{gg}}$;
$\alpha \leftarrow \lambda^{\text{inc}} \cdot \alpha_0$;
$\tilde{\alpha} \leftarrow \frac{\alpha}{D - \alpha D_g^2}$;
$v \leftarrow v + \tilde{\alpha} \cdot (D_g\mu_0 - D_{gr})(D_g\tilde{\eta} - D\eta)$;
$\mu_0 \leftarrow \mu_0 + \tilde{\alpha} \cdot D_g(D_g\mu_0 - D_{gr})$;
$D \leftarrow D - \alpha D_g^2$;
$(D_g, D_{gg}, D_{gr}) \leftarrow \alpha_0(D_g, D_{gg}, D_{gr})$.
$M_{\cdot,S} \leftarrow M_{\cdot,S} - \alpha\eta\eta_S^T$;
$\tilde{\eta} \leftarrow \tilde{\eta} - \alpha D_g \eta$;
$\eta \leftarrow \alpha_0 \eta$;

Output: $v, \mu_0, \mathrm{Par}_1, \mathrm{Par}_2$.

column of W to be zero and for vector $z \in \mathbb{R}^{n\times 1}$, $\mathcal{R}_j(z)$ sets the j-th coordinate of z to be zero. One property of $\mathcal{R}_j$ to be used is that For any matrix-vector pair (W, z),

$$\mathcal{R}_j(W)z = \mathcal{R}_j(Wz) - z_j\mathcal{R}_j(W_j) \tag{B-8}$$

where W_j is j-th column of W.

Theorem B-5 *Let γ and $\tilde{\gamma}$ be two $n \times 1$ vectors with*

$$\gamma_{\tilde{S}} = \tilde{\gamma}_{\tilde{S}} = (M_{jS} \quad 1)^T, \quad \gamma_{\tilde{S}^c} = -A_{\tilde{S}^c j} - A_{\tilde{S}^c S}M_{jS}^T, \quad \tilde{\gamma}_{\tilde{S}^c} = 0.$$

Then

$$M^+ = \mathcal{R}_j(M) + \frac{1}{\tilde{A}_{jj}} \cdot \gamma\tilde{\gamma}^T.$$

Proof By definition,

$$M^+_{\tilde{S}\tilde{S}} = A^{-1}_{\tilde{S}\tilde{S}}, \quad M^+_{\tilde{S}^c\tilde{S}} = -A_{\tilde{S}^c\tilde{S}}A^{-1}_{\tilde{S}\tilde{S}}.$$

By Proposition 1,

$$M^+_{\tilde{S}\tilde{S}} = \begin{pmatrix} M_{SS} & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{\tilde{A}_{jj}} \begin{pmatrix} M_{jS}^T \\ 1 \end{pmatrix} (M_{jS} \quad 1) = \begin{pmatrix} M_{SS} & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{\tilde{A}_{jj}} \gamma_{\tilde{S}}\tilde{\gamma}_{\tilde{S}}^T,$$

and

$$M^+_{\tilde{S}^c\tilde{S}} = -(A_{\tilde{S}^c S} \quad A_{\tilde{S}^c j})\left\{\begin{pmatrix} M_{SS} & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{\tilde{A}_{jj}}\begin{pmatrix} M_{jS}^T \\ 1 \end{pmatrix}(M_{jS} \quad 1)\right\}$$
$$= (M_{\tilde{S}^c S} \quad 0) + \frac{1}{\tilde{A}_{jj}}\gamma_{\tilde{S}^c}\tilde{\gamma}_{\tilde{S}}^T.$$

Note that $M_{\cdot,\tilde{S}^c}$ is always a zero matrix by definition, the above results imply that

$$M^+ = \mathcal{R}_j(M) + \frac{1}{\tilde{A}_{jj}} \cdot \gamma\tilde{\gamma}^T.$$

■

The update of other parameters can be derived as a consequence of Theorem B-5. Theorem B-6 summarizes the result.

Theorem B-6 *Let* $b_{j,S} = -r_{\tilde{S}}^T\gamma_{\tilde{S}}$, *then*

- $\eta^+ = \mathcal{R}_j(\eta) + \frac{\eta_j}{\tilde{A}_{jj}}\gamma$;
- $\tilde{\eta}^+ = \mathcal{R}_j(\tilde{\eta}) + \frac{\tilde{\eta}_j}{\tilde{A}_{jj}}\gamma$;
- $D^+ = D + \frac{\tilde{\eta}_j^2}{\tilde{A}_{jj}}$
- $D_g^+ = D_g + \frac{\eta_j\tilde{\eta}_j}{\tilde{A}_{jj}}$;
- $D_{gg}^+ = D_{gg} + \frac{\eta_j^2}{\tilde{A}_{jj}}$;
- $D_{gr}^+ = D_{gr} + \frac{\eta_j b_{j,S}}{\tilde{A}_{jj}}$;

Proof Since $M_j = 0$, (B-8) implies that for any $z \in \mathbb{R}^{n\times 1}$

$$\mathcal{R}_j(Mz) = \mathcal{R}_j(M)z.$$

By definition,

$$\eta = \begin{pmatrix} 0 \\ g_{S^c} \end{pmatrix} + Mg, \quad \tilde{\eta} = \begin{pmatrix} 0 \\ \mathbf{1}_{S^c} \end{pmatrix} + M\mathbf{1}.$$

Also notice that $\tilde{\gamma}_{\tilde{S}}^T g_{\tilde{S}} = g_j + M_{jS}^T g_S = \eta_j$ and $\gamma_{\tilde{S}}^T\mathbf{1}_{\tilde{S}} = 1 + M_{jS}\mathbf{1}_S = \tilde{\eta}_j$, thus,

$$\eta^+ = \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + M^+g = \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \mathcal{R}_j(M)g + \frac{\tilde{\gamma}^T g}{\tilde{A}_{jj}}\gamma$$
$$= \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \mathcal{R}_j(Mg) + \frac{\tilde{\gamma}_{\tilde{S}}^T g_{\tilde{S}}}{\tilde{A}_{jj}}\gamma$$
$$= \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \mathcal{R}_j(\eta) - \mathcal{R}_j\left(\begin{pmatrix} 0 \\ g_{S^c} \end{pmatrix}\right) + \frac{\eta_j}{\tilde{A}_{jj}}\gamma$$
$$= \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \mathcal{R}_j(\eta) - \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \frac{\eta_j}{\tilde{A}_{jj}}\gamma$$
$$= \mathcal{R}_j(\eta) + \frac{\eta_j}{\tilde{A}_{jj}}\gamma.$$

The update of $\tilde{\eta}$ can be obtained by replacing g by $\mathbf{1}$ in the above derivation. The four scalars D, D_g, D_{gg}, D_{gr} can be updated as follows.

$$D^+ = \mathbf{1}_{\tilde{S}}^T\tilde{\eta}_{\tilde{S}}^+ = \mathbf{1}_{\tilde{S}}^T\left(\mathcal{R}_j(\tilde{\eta})_{\tilde{S}} + \frac{\tilde{\eta}_j}{\tilde{A}_{jj}}\gamma_{\tilde{S}}\right) = D + \frac{\tilde{\eta}_j^2}{\tilde{A}_{jj}};$$

$$
\begin{aligned}
D_g^+ &= \mathbf{1}_{\tilde{S}}^T \eta_{\tilde{S}}^+ = \mathbf{1}_{\tilde{S}}^T \left(\mathcal{R}_j(\eta)_{\tilde{S}} + \frac{\eta_j}{\tilde{A}_{jj}} \gamma_{\tilde{S}} \right) = D_g + \frac{\tilde{\eta}_j \eta_j}{\tilde{A}_{jj}};\\
D_{gg}^+ &= g_{\tilde{S}}^T \eta_{\tilde{S}}^+ = g_{\tilde{S}}^T \left(\mathcal{R}_j(\eta)_{\tilde{S}} + \frac{\eta_j}{\tilde{A}_{jj}} \gamma_{\tilde{S}} \right) = D_{gg} + \frac{\eta_j^2}{\tilde{A}_{jj}};\\
D_{gr}^+ &= -r_{\tilde{S}}^T \eta_{\tilde{S}}^+ = -r_{\tilde{S}}^T \left(\mathcal{R}_j(\eta_{\tilde{S}}) + \frac{\eta_j}{\tilde{A}_{jj}} \gamma_{\tilde{S}} \right) = D_{gr} - \frac{\eta_j}{\tilde{A}_{jj}} r_{\tilde{S}}^T \gamma_{\tilde{S}}\\
&= D_{gr} + \frac{\eta_j b_{j,S}}{\tilde{A}_{jj}}.
\end{aligned}
$$

■

The implementation of UPDATE_EXPAND_SUPPORT is summarized in Algorithm 4. Note that both $\tilde{A}_{jj}$ and $\gamma_{\tilde{S}^c}$ depends on λ and it is easy to see that

$$
\begin{aligned}
&\tilde{A}_{jj}(\lambda) = A_{jj} + \lambda g_j^2 + M_{jS}(A_{Sj} + \lambda g_j g_S) = A_{jj} + M_{jS} A_{Sj} + \lambda g_j \eta_j\\
&\gamma_{\tilde{S}^c}(\lambda) \leftarrow -(A_{\tilde{S}^c j} + \lambda g_{\tilde{S}^c} g_j) - (A_{\tilde{S}^c S} + \lambda g_{\tilde{S}^c} g_S^T) M_{jS}^T = -A_{\tilde{S}^c j} - A_{\tilde{S}^c S} M_{jS}^T - \lambda \eta_j g_{\tilde{S}^c}.
\end{aligned}
$$

Algorithm 4 UPDATE_EXPAND_SUPPORT

Inputs: Current λ, original support S, new index j, matrix A, vectors y, r, g, intermediate variables $\mathrm{Par}_1, \mathrm{Par}_2$.

Procedure:

1: $\tilde{A}_{jj} \leftarrow A_{jj} + M_{jS} A_{Sj} + \lambda g_j \eta_j$;
2: $\gamma_{\tilde{S}} \leftarrow (M_{jS}, 1)^T, \gamma_{\tilde{S}^c} \leftarrow -A_{\tilde{S}^c j} - A_{\tilde{S}^c S} M_{jS}^T - \lambda \eta_j g_{\tilde{S}^c}$;
3: $\tilde{\gamma}_{\tilde{S}} \leftarrow (M_{jS}, 1)^T, \tilde{\gamma}_{\tilde{S}^c} \leftarrow 0$;
4: $b \leftarrow -r_{\tilde{S}}^T \gamma_{\tilde{S}}$;
5: $D \leftarrow D + \frac{\tilde{\eta}_j^2}{\tilde{A}_{jj}}$;
6: $D_g \leftarrow D_g + \frac{\eta_j \tilde{\eta}_j}{\tilde{A}_{jj}}$;
7: $D_{gg} \leftarrow D_{gg} + \frac{\eta_j^2}{\tilde{A}_{jj}}$;
8: $D_{gr} \leftarrow D_{gr} + \frac{\eta_j b}{\tilde{A}_{jj}}$;
9: $M_{\cdot,\tilde{S}} \leftarrow \mathcal{R}_j(M_{\cdot,\tilde{S}}) + \frac{1}{\tilde{A}_{jj}} \gamma \tilde{\gamma}_{\tilde{S}}^T$;
10: $\eta \leftarrow \mathcal{R}_j(\eta) + \frac{\eta_j}{\tilde{A}_{jj}} \gamma$.
11: $\tilde{\eta} \leftarrow \mathcal{R}_j(\tilde{\eta}) + \frac{\tilde{\eta}_j}{\tilde{A}_{jj}} \gamma$;

Output: $\mathrm{Par}_1, \mathrm{Par}_2$.

B-4.3 UPDATE_SHRINK_SUPPORT

Suppose S is updated to $S \setminus \{j\}$ for some $j \in S^c$. Denote $\tilde{S}$ by $S \setminus \{j\}$ and we add a supscript $-$ to each variable to denote the value after update. Similar to last subsection, we start from deriving M^- and apply the result to calculate other variables.

Theorem B-7 *Let β and $\tilde{\beta}$ be two $n \times 1$ vectors with*

$$
\beta_{\tilde{S}} = \tilde{\beta}_{\tilde{S}} = M_{\tilde{S}j}, \quad \beta_{\tilde{S}^c} = \begin{pmatrix} -1 \\ M_{S^c j} \end{pmatrix} \quad \tilde{\beta}_{\tilde{S}^c} = 0.
$$

Then

$$
M^- = \mathcal{R}_j(M) - \frac{1}{M_{jj}} \cdot \beta \tilde{\beta}^T.
$$

Proof By definition,

$$\begin{pmatrix} M_{\tilde{S}\tilde{S}} & M_{\tilde{S}j} \\ M_{j\tilde{S}} & M_{jj} \end{pmatrix} = A_{SS}^{-1} = \begin{pmatrix} A_{\tilde{S}\tilde{S}} & A_{\tilde{S}j} \\ A_{j\tilde{S}} & A_{jj} \end{pmatrix}^{-1}.$$

Then Proposition 1 implies that

$$\begin{pmatrix} M_{\tilde{S}\tilde{S}} & M_{\tilde{S}j} \\ M_{j\tilde{S}} & M_{jj} \end{pmatrix} = \begin{pmatrix} A_{\tilde{S}\tilde{S}}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{A_{jj} - A_{j\tilde{S}} A_{\tilde{S}\tilde{S}}^{-1} A_{\tilde{S}j}} \cdot \begin{pmatrix} -A_{\tilde{S}\tilde{S}}^{-1} A_{\tilde{S}j} \\ 1 \end{pmatrix} (-A_{j\tilde{S}} A_{\tilde{S}\tilde{S}}^{-1} \quad 1). \tag{B-9}$$

This entails that

$$A_{\tilde{S}\tilde{S}}^{-1} = M_{\tilde{S}\tilde{S}} - \frac{M_{\tilde{S}j} M_{j\tilde{S}}}{M_{jj}}, \quad -A_{j\tilde{S}} A_{\tilde{S}\tilde{S}}^{-1} = \frac{M_{j\tilde{S}}}{M_{jj}}. \tag{B-10}$$

On the other hand,

$$\begin{aligned} (M_{S^c\tilde{S}} \quad M_{S^c j}) &= -A_{S^c S} A_{SS}^{-1} = -(A_{S^c\tilde{S}} \quad A_{S^c j}) \begin{pmatrix} M_{\tilde{S}\tilde{S}} & M_{\tilde{S}j} \\ M_{j\tilde{S}} & M_{jj} \end{pmatrix} \\ &= -(A_{S^c\tilde{S}} M_{\tilde{S}\tilde{S}} + A_{S^c j} M_{j\tilde{S}} \quad A_{S^c\tilde{S}} M_{\tilde{S}j} + A_{S^c j} M_{jj}). \end{aligned} \tag{B-11}$$

If follows from (B-9), (B-10) and (B-11) that

$$\begin{aligned} -A_{S^c\tilde{S}} A_{\tilde{S}\tilde{S}}^{-1} &= -A_{S^c\tilde{S}} \left(M_{\tilde{S}\tilde{S}} - \frac{M_{\tilde{S}j} M_{j\tilde{S}}}{M_{jj}} \right) \\ &= M_{S^c\tilde{S}} + A_{S^c j} M_{j\tilde{S}} + \frac{A_{S^c\tilde{S}} M_{\tilde{S}j} M_{j\tilde{S}}}{M_{jj}} \\ &= M_{S^c\tilde{S}} + (A_{S^c j} M_{jj} + A_{S^c\tilde{S}} M_{\tilde{S}j}) \frac{M_{j\tilde{S}}}{M_{jj}} \\ &= M_{S^c\tilde{S}} - \frac{M_{S^c j} M_{j\tilde{S}}}{M_{jj}}. \end{aligned} \tag{B-12}$$

Putting (B-10) and (B-11) together, we obtain that

$$M^{-}_{\cdot,\tilde{S}} = \begin{pmatrix} A_{\tilde{S}\tilde{S}}^{-1} \\ -A_{\tilde{S}^c\tilde{S}} A_{\tilde{S}\tilde{S}}^{-1} \end{pmatrix} = \begin{pmatrix} A_{\tilde{S}\tilde{S}}^{-1} \\ -A_{j\tilde{S}} A_{\tilde{S}\tilde{S}}^{-1} \\ -A_{S^c\tilde{S}} A_{\tilde{S}\tilde{S}}^{-1} \end{pmatrix} = \mathcal{R}_j(M)_{\cdot,\tilde{S}} - \frac{1}{M_{jj}} \cdot \beta \tilde{\beta}_{\tilde{S}}^T.$$

Since $M_{\cdot,\tilde{S}^c}$ is a zero matrix,

$$M^{-} = \mathcal{R}_j(M) - \frac{1}{M_{jj}} \cdot \beta \tilde{\beta}^T.$$

■

Theorem B-8 *Let* $\tilde{b}_{j,S} = -r_{\tilde{S}}^T \beta_{\tilde{S}} - r_j M_{jj}$, *then*

- $\eta^{-} = \mathcal{R}_j(\eta) - \frac{\eta_j}{M_{jj}} \beta$;
- $\tilde{\eta}^{-} = \mathcal{R}_j(\tilde{\eta}) - \frac{\tilde{\eta}_j}{M_{jj}} \beta$;
- $D^{-} = D - \frac{\tilde{\eta}_j^2}{M_{jj}}$;
- $D_g^{-} = D_g - \frac{\eta_j \tilde{\eta}_j}{M_{jj}}$;
- $D_{gg}^{-} = D_{gg} - \frac{\eta_j^2}{M_{jj}}$;
- $D_{gr}^{-} = D_{gr} - \frac{\eta_j \tilde{b}_{j,S}}{M_{jj}}$.

Algorithm 5 UPDATE_SHRINK_SUPPORT

Inputs: Original support S, new index j, matrix A, vector y, r, g, intermediate variables $\mathrm{Par}_1, \mathrm{Par}_2$.

Procedure:

$\beta_{\tilde{S}} \leftarrow M_{j\tilde{S}}^T, \beta_{\tilde{S}^c} \leftarrow \begin{pmatrix} -1 \\ M_{\tilde{S}^c j} \end{pmatrix}, \tilde{\beta}_{\tilde{S}} \leftarrow M_{j\tilde{S}}^T, \tilde{\beta}_{\tilde{S}^c} \leftarrow 0;$

$\tilde{b} \leftarrow -r_{\tilde{S}}^T \beta_{\tilde{S}} - r_j M_{jj};$

$D \leftarrow D - \frac{\tilde{\eta}_j^2}{M_{jj}};$

$D_g \leftarrow D_g - \frac{\eta_j \tilde{\eta}_j}{M_{jj}};$

$D_{gg} \leftarrow D_{gg} - \frac{\eta_j^2}{M_{jj}};$

$D_{gr} \leftarrow D_{gr} - \frac{\eta_j \tilde{b}}{M_{jj}};$

$M_{\cdot,\tilde{S}} \leftarrow \mathcal{R}_j(M_{\cdot,\tilde{S}}) - \frac{1}{M_{jj}} \beta \tilde{\beta}_{\tilde{S}}^T, \;\; M_{\cdot,j} \leftarrow 0;$

$\eta \leftarrow \mathcal{R}_j(\eta) - \frac{\eta_j}{M_{jj}} \beta;$

$\tilde{\eta} \leftarrow \mathcal{R}_j(\tilde{\eta}) - \frac{\tilde{\eta}_j}{M_{jj}} \beta.$

Output: $\mathrm{Par}_1, \mathrm{Par}_2$.

Proof By (B-8),

$$\mathcal{R}_j(M)g = \mathcal{R}_j(Mg) - g_j \mathcal{R}_j(M_{\cdot,j})$$

Let e_j is the j-th basis vector with j-th entry equal to 1 and all other entries equal to 0. Then

$$\begin{aligned}
\eta^- &= \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + M^- g = \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \mathcal{R}_j(M)g - \frac{\tilde{\beta}^T g}{M_{jj}} \beta \\
&= \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \mathcal{R}_j(Mg) - g_j \mathcal{R}_j(M_{\cdot,j}) - \frac{\tilde{\beta}_{\tilde{S}}^T g_{\tilde{S}}}{M_{jj}} \beta \\
&= \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \mathcal{R}_j(Mg) - g_j e_j - g_j \beta - \frac{\tilde{\beta}_{\tilde{S}}^T g_{\tilde{S}}}{M_{jj}} \beta \\
&= \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \mathcal{R}_j(Mg) - g_j e_j - \frac{M_{jS} g_S}{M_{jj}} \beta \\
&= \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \mathcal{R}_j(\eta) - \mathcal{R}_j\left(\begin{pmatrix} 0 \\ g_{S^c} \end{pmatrix}\right) - g_j e_j - \frac{\eta_j}{M_{jj}} \beta \\
&= \begin{pmatrix} 0 \\ g_{\tilde{S}^c} \end{pmatrix} + \mathcal{R}_j(\eta) - \begin{pmatrix} 0 \\ g_{S^c} \end{pmatrix} - g_j e_j - \frac{\eta_j}{M_{jj}} \beta \\
&= \mathcal{R}_j(\eta) - \frac{\eta_j}{M_{jj}} \beta.
\end{aligned}$$

Substitute g by $\mathbf{1}$, we obtain the update for $\tilde{\eta}$. Together with (B-10) and the fact that $j \in S$, we obtain that

$$\begin{aligned}
D^- &= \mathbf{1}_{\tilde{S}}^T \tilde{\eta}_{\tilde{S}}^- = \mathbf{1}_{\tilde{S}}^T \left(\tilde{\eta}_{\tilde{S}} - \frac{\tilde{\eta}_j}{M_{jj}} \beta_{\tilde{S}}\right) = D - \tilde{\eta}_j - \frac{\tilde{\eta}_j}{M_{jj}} (\mathbf{1}_{\tilde{S}}^T M_{\tilde{S}j}) \\
&= D - \frac{\tilde{\eta}_j}{M_{jj}} (M_{jj} + \mathbf{1}_{\tilde{S}}^T M_{\tilde{S}j}) = D - \frac{\tilde{\eta}_j^2}{M_{jj}}; \\
D_g^- &= \mathbf{1}_{\tilde{S}}^T \eta_{\tilde{S}}^- = \mathbf{1}_{\tilde{S}}^T \left(\eta_{\tilde{S}} - \frac{\eta_j}{M_{jj}} \beta_{\tilde{S}}\right) = D_g - \eta_j - \frac{\eta_j}{M_{jj}} (\mathbf{1}_{\tilde{S}}^T M_{\tilde{S}j}) \\
&= D_g - \frac{\eta_j}{M_{jj}} (M_{jj} + \mathbf{1}_{\tilde{S}}^T M_{\tilde{S}j}) = D_g - \frac{\eta_j \tilde{\eta}_j}{M_{jj}}; \\
D_{gg}^- &= g_{\tilde{S}}^T \eta_{\tilde{S}}^- = g_{\tilde{S}}^T \left(\eta_{\tilde{S}} - \frac{\eta_j}{M_{jj}} \beta_{\tilde{S}}\right) = D_{gg} - g_j \eta_j - \frac{\eta_j}{M_{jj}} (g_{\tilde{S}}^T M_{\tilde{S}j})
\end{aligned}$$

$$= D_{gg} - \frac{\eta_j}{M_{jj}}(g_j M_{jj} + g_{\tilde{S}}^T M_{\tilde{S}j}) = D_{gg} - \frac{\eta_j^2}{M_{jj}};$$

$$D_{gr}^- = -r_{\tilde{S}}^T \eta_{\tilde{S}}^- = -r_{\tilde{S}}^T \left(\eta_{\tilde{S}} - \frac{\eta_j}{M_{jj}}\beta_{\tilde{S}}\right) = -r_S^T \eta_S + r_j \eta_j + \frac{\eta_j}{M_{jj}} r_{\tilde{S}}^T \beta_{\tilde{S}}$$

$$= D_{gr} - \frac{\eta_j \tilde{b}_{j,S}}{M_{jj}}.$$

■

The implementation of UPDATE_SHRINK_SUPPORT is summarized in Algorithm 5.

B-4.4 DIRECT_UPDATE

At the beginning of each time t, we need to recompute $\mathrm{Par}_2 = \{\eta, D_g, D_{gg}, D_{gr}\}$. The implementation is summarized in Algorithm 6.

Algorithm 6 DIRECT_UPDATE

Inputs: Support S, vector y, r, g, intermediate variables $\mathrm{Par}_1, \mathrm{Par}_2$.
Procedure:

$\eta_S \leftarrow M_{SS} g_S, \eta_{S^c} \leftarrow g_{S^c} + M_{S^c S} g_S$;
$D_g \leftarrow \mathbf{1}_S^T \eta_S$;
$D_{gg} \leftarrow \eta_S^T g_S$;
$D_{gr} \leftarrow -\eta_S r_S$.

Output: Par_2.

B-5 Update of A

As will be shown in next subsection, the complexities of all above sub-routines are at most $O(ns)$ where $s = |S|$. However, the complexity of line 16 is $O(n^2)$ which might dominate when the solution is sparse and the number of turning points is small. Fortunately, UPDATE_EXPAND_SUPPORT is the only sub-routine which extracts information from A. In fact, in line 1 and line 2,

$$\begin{pmatrix} \tilde{A}_{jj} \\ \gamma_{\tilde{S}^c} \end{pmatrix} = \begin{pmatrix} A_{jj} + M_{jS} A_{Sj} \\ -A_{\tilde{S}^c j} - A_{\tilde{S}^c S} M_{jS}^T \end{pmatrix} + \lambda \eta_j \begin{pmatrix} g_j \\ g_{\tilde{S}^c} \end{pmatrix}.$$

This only requires the j-th column of A. Let S_* be the union of all supports appeared in Algorithm 1. Suppose we know S_* apriori, we can only update the columns of A with indices in S_*. In other words, we update $A_{\cdot, S_*}$ by $A_{\cdot, S_*} + \lambda g g_{S_*}^T$ at the beginning of each step and hence the complexity is reduced to $O(n|S_*|)$.

Although agnostic to S_* in reality, we can initialize it by $\mathrm{supp}(x_k)$ for some positive k, e.g. $k = 1$, and keep track it by adding index into S_* once the index is not included in S_*. Once a new index j is detected, we update j-th column of A by using all previous $g^{(t)}$. The implementation is stated in Algorithm 7.

B-6 Complexity Analysis

In this subsection, we analyze the complexity of the algorithm. We distinguish four types of computation, namely matrix-vector product, outer-product of two vectors, inner-product of two vectors and vector-scalar product. Denote by $W \in \mathbb{R}^{m \times p}$, $(z, \tilde{z}) \in \mathbb{R}^p \times \mathbb{R}^q$ and $a \in \mathbb{R}$ the generic matrix, vector and scalar respectively. As a convention, the complexity is defined as the number of scalar-scalar multiplications. The addition is omitted here for simplicity. Note that the complexities of Wz, $z\tilde{z}^T$, $z^T z$ and az are mp, pq, p and p, respectively. The results for a single step are summarized in Table 1 where $s_* = |S_*|$ be the size of S_* at the final round. We should emphasize that our complexity analysis is exact.

[1] S_* might be updated, in which step nt-computations are involved in line 17 of Algorithm 7. However, on average, ns_* computations are involved since s_* represents the size of S_* at the last round.

Algorithm 7 HONES Algorithm for time-varing A and fixed r with sparse update of A

Inputs: Initial matrix $A^{(0)}$, vectors r, matrix-update-vectors $\{g^{(t)}, t = 1, 2, \ldots\}$.

Initialization:

$x \leftarrow$ as the optimum corresponding to $A^{(0)}$;
$S \leftarrow \mathrm{supp}(x), S_* \leftarrow S$;
Calculate (x, μ, μ_0) via (8)-(10)
$v_S \leftarrow x_S, \quad v_{S^c} \leftarrow -\mu_{S^c}$;
Calculate intermediate variables $(\mathrm{Par}_1, \mathrm{Par}_2)$ via (B-1)-(B-3) based on $g^{(1)}$.

Procedure:

for $t = 1, 2, \cdots$. **do**
    $\lambda \leftarrow 0$;
    while $\lambda < 1$ **do**
        $(\lambda^{\mathrm{inc}}, j, S^{\mathrm{new}}) \leftarrow \mathrm{FIND\_LAMBDA}(S, v; \mathrm{Par}_1, \mathrm{Par}_2)$;
        $\lambda^{\mathrm{inc}} \leftarrow \min\{\lambda^{\mathrm{inc}}, 1 - \lambda\}$;
        $\lambda \leftarrow \lambda + \lambda^{\mathrm{inc}}$;
        $(v, \mu_0; \mathrm{Par}_1, \mathrm{Par}_2) \leftarrow \mathrm{UPDATE\_BY\_LAMBDA}(\lambda^{\mathrm{inc}}; v, \mu_0; \mathrm{Par}_1, \mathrm{Par}_2)$;
        if $S^{\mathrm{new}} = S \cup \{j\}$ **then**
            $(\mathrm{Par}_1, \mathrm{Par}_2) \leftarrow \mathrm{UPDATE\_EXPAND\_SUPPORT}(\lambda, S, j; A, r, g^{(t)}; \mathrm{Par}_1, \mathrm{Par}_2)$;
            if $j \notin S_*$ **then**
                $G \leftarrow (g^{(1)}, \ldots, g^{(t-1)})$;
                $A_{\cdot,j} \leftarrow A_{\cdot,j} + G G^T_{j,\cdot}$;
                $S_* = S_* \cup \{j\}$;
            end if
        else if $S^{\mathrm{new}} = S \setminus \{j\}$ **then**
            $(\mathrm{Par}_1, \mathrm{Par}_2) \leftarrow \mathrm{UPDATE\_SHRINK\_SUPPORT}(S, j; r, g^{(t)}; \mathrm{Par}_1, \mathrm{Par}_2)$;
        end if
        $S \leftarrow S^{\mathrm{new}}$.
    end while
    $\mathrm{Par}_2 \leftarrow \mathrm{DIRECT\_UPDATE}(S, r, g^{(t+1)}; \mathrm{Par}_1, \mathrm{Par}_2)$;
    $A_{\cdot,S_*} \leftarrow A_{\cdot,S_*} + g^{(t)} (g^{(t)}_{S_*})^T$;
    $x^{(t)}_S \leftarrow x_S, \quad x^{(t)}_{S^c} \leftarrow 0$.
end for

Output: $x^{(1)}, x^{(2)}, \cdots$..

Table 1: Computation complexity of each sub-routine in Algorithm 1.

	(Wz)-type	$(z\tilde{z}^T)$-type	$(z^T z)$-type	(az)-type
FIND_LAMBDA	0	0	n	$2n$
UPDATE_BY_LAMBDA	0	ns	0	$4n$
UPDATE_EXPAND_SUPPORT	$s(n-s-1)$	$n(s+1)$	n	$2n$
UPDATE_SHRINK_SUPPORT	0	$n(s-1)$	n	$2n$
DIRECT_UPDATE	ns	0	$n+s$	0
Update [1] of A.	0	ns_*	0	0

For given t, denote k_A^+ by the number of turning points which add element to S and k_A^- by the number of turning points which delete element from S. Let $k_A = k_A^+ + k_A^-$ be the total number of turning points and s be the maximum size of S in the iteration Then the complexity of HONES for a single time t is at most

$$ns(3k_A^+ + 2k_A^-) + n(12k_A^+ + 10k_A^-) + O(k_A),$$

Therefore, the complexity at time t is at most

$$C_{1t} \le ns_* + ns(3k_A^+ + 2k_A^- + 1) + n(12k_A^+ + 10k_A^- + 2) + O(k_A) \le ns_* + ns(3k_A + 1) + n(12k_A + 2) + O(k_A).$$

C Implementation of HONES Algorithm With Time-Varying r and Fixed A

C-1 Intermediate Variables

Similar to Appendix B, we define $\mathrm{Par}_1 = \{M, \tilde{\eta}, D\}$ where the parameters are defined in (B-1)-(B-3). Moreover, we define a vector ξ such that

$$\xi_S = -A_{SS}^{-1}\ell_S, \quad \xi_{S^c} = -\ell_{S^c} + A_{S^cS}A_{SS}^{-1}\ell_S,$$

and a scalar D_ℓ as

$$D_\ell = \mathbf{1}_S^T \xi_S.$$

We write Par_3 for $\{\xi, D_\ell\}$ for convenience.

C-2 Implementation

Algorithm 8 describes the full implementation in this case and the sub-routines will be discussed separately in following subsections.

Algorithm 8 HONES Algorithm for constant A, y and time-varying r

Inputs: Initial matrix A, vector $r^{(0)}$, vector-update-vector $\{\ell^{(t)} = r^{(t)} - r^{(t-1)} : t = 1, 2, \ldots\}$.
Initialization:

$x \leftarrow$ as the optimum corresponding to $r^{(0)}$.
$S \leftarrow \mathrm{supp}(x)$;
Calculate (x, μ, μ_0) via (8)-(10)
$v_S \leftarrow x_S, v_{S^c} \leftarrow -\mu_{S^c}$;
Calculate intermediate variables $\mathrm{Par}_1, \mathrm{Par}_3$ via (B-1)-(B-3) based on $\ell^{(1)}$.

Procedure:

for $t = 1, 2, \cdots$. **do**
  $\underset{\sim}{\lambda} \leftarrow 0$;
  while $\underset{\sim}{\lambda} < 1$ **do**
    $(\underset{\sim}{\lambda}^{\mathrm{inc}}, j, S^{\mathrm{new}}) \leftarrow \mathrm{FIND\_UTILDE\_LAMBDA}(S, v; \mathrm{Par}_1, \mathrm{Par}_3)$;
    $\underset{\sim}{\lambda}^{\mathrm{inc}} \leftarrow \min\{\underset{\sim}{\lambda}^{\mathrm{inc}}, 1 - \underset{\sim}{\lambda}\}$;
    $(v, \mu_0; \mathrm{Par}_1, \mathrm{Par}_3) \leftarrow \mathrm{UPDATE\_BY\_UTILDE\_LAMBDA}(\underset{\sim}{\lambda}^{\mathrm{inc}}; v, \mu_0, \mathrm{Par}_1, \mathrm{Par}_3)$;
    if $S^{\mathrm{new}} = S \cup \{j\}$ **then**
      $(\mathrm{Par}_1, \mathrm{Par}_3) \leftarrow \mathrm{UPDATE\_UTILDE\_EXPAND\_SUPPORT}(S, j, A, \ell; \mathrm{Par}_1, \mathrm{Par}_3)$;
    else if $S^{\mathrm{new}} = S \setminus \{j\}$ **then**
      $(\mathrm{Par}_1, \mathrm{Par}_3) \leftarrow \mathrm{UPDATE\_UTILDE\_SHRINK\_SUPPORT}(S, j, \ell; \mathrm{Par}_1, \mathrm{Par}_3)$;
    end if
    $S \leftarrow S^{\mathrm{new}}$;
    $\underset{\sim}{\lambda} \leftarrow \underset{\sim}{\lambda} + \underset{\sim}{\lambda}^{\mathrm{inc}}$.
  end while
  $\mathrm{Par}_3 \leftarrow \mathrm{DIRECT\_UTILDE\_UPDATE}(S, \mathrm{Par}_1, h^{(t+1)})$;
  $x_S^{(t)} \leftarrow x_S, \quad x_{S^c}^{(t)} \leftarrow 0$.
end for

Output: $x^{(1)}, x^{(2)}, \cdots$.

C-3 FIND_UTILDE_LAMBDA

Define v as in (B-5). Then Theorem E-11 implies that

$$v(\underset{\sim}{\lambda}) = v(0) - \left(\xi - \frac{D_\ell}{D}\tilde{\eta}\right)\underset{\sim}{\lambda}, \quad \mu_0(\underset{\sim}{\lambda}) = \mu_0 + \frac{D_\ell}{D}\underset{\sim}{\lambda}.$$

Thus, searching for $\underset{\sim}{\lambda}$ is equivalent to solve simple linear equations. Algorithm 9

Algorithm 9 FIND_UTILDE_LAMBDA

Input: Support S, iterate $v = \begin{pmatrix} x_S \\ -\mu_{S^c} \end{pmatrix}$, intermediate variables $\mathrm{Par}_1, \mathrm{Par}_3$.

Procedure:

$\underset{\sim}{\lambda}^{\text{inc}} \leftarrow \min_+ \left\{\frac{v_i}{\xi_i - \frac{D_\ell}{D}\tilde{\eta}_i} : i = 1, 2, \ldots, n\right\}$;

$j \leftarrow \mathrm{argmin}_+ \left\{\frac{v_i}{\xi_i - \frac{D_\ell}{D}\tilde{\eta}_i} : i = 1, 2, \ldots, n\right\}$;

if $j \in S$ **then**

  $S^{\text{new}} = S \setminus \{j\}$;

else

  $S^{\text{new}} = S \cup \{j\}$.

end if

Output: $\underset{\sim}{\lambda}^{\text{inc}}, j, S^{\text{new}}$.

C-4 Variables Update

C-4.1 UPDATE_BY_UTILDE_LAMBDA

Note that all intermediate variables are not affected by $\underset{\sim}{\lambda}$, we only need to update v and μ_0 accordingly.

Algorithm 10 UPDATE_BY_UTILDE_LAMBDA

Input: Increment $\underset{\sim}{\lambda}^{\text{inc}}$; iterate $v = \begin{pmatrix} x_S \\ -\mu_{S^c} \end{pmatrix}$, μ_0; intermediate variables $\mathrm{Par}_1, \mathrm{Par}_3$.

Procedure:

$v \leftarrow v - \left(\xi - \frac{D_\ell}{D}\tilde{\eta}\right)\underset{\sim}{\lambda}^{\text{inc}}$;

$\mu_0 \leftarrow \mu_0 + \frac{D_\ell}{D}\underset{\sim}{\lambda}^{\text{inc}}$.

Output: v, μ_0.

C-4.2 UPDATE_UTILDE_EXPAND_SUPPORT

Since M is exactly the same as in Appendix B, we can directly apply Theorem B-5 to obtain an update of M and the updates of other parameters as a consequence.

Theorem C-9 *Let γ and $\tilde{\gamma}$ be defined in Theorem B-5, i.e.*

$$\gamma_{\tilde{S}} = \tilde{\gamma}_{\tilde{S}} = (M_{jS} \quad 1)^T, \quad \gamma_{\tilde{S}^c} = -A_{\tilde{S}^c j} - A_{\tilde{S}^c S} M_{jS}^T, \quad \tilde{\gamma}_{\tilde{S}^c} = 0,$$

then

- $M^+ = \mathcal{R}_j(M) + \frac{1}{\tilde{A}_{jj}}\gamma\tilde{\gamma}^T$;
- $\tilde{\eta}^+ = \mathcal{R}_j(\tilde{\eta}) + \frac{\tilde{\eta}_j}{\tilde{A}_{jj}}\gamma$;
- $D^+ = D + \frac{\tilde{\eta}_j^2}{\tilde{A}_{jj}}$;

- $\xi^+ = \mathcal{R}_j(\xi) + \frac{\xi_j}{\tilde{A}_{jj}}\gamma$;
- $D_\ell^+ = D_\ell + \frac{\xi_j \tilde{\eta}_j}{\tilde{A}_{jj}}$.

Proof The update of M, $\tilde{\eta}$ and Dhas been proved in Theorem B-5. For any subset S, let I_S denote the matrix with j-th diagonal element equal to 1 for any $j \in S$ and all other elements equal to 0. Then ξ and ξ^+ can be rewritten as

$$\xi = -\left(M + I_{S^c}\right)\ell, \quad \xi^+ = -\left(M^+ + I_{\tilde{S}^c}\right)\ell.$$

Note that $I_{S^c} - I_{\tilde{S}^c} = e_j e_j^T$ where e_j is the j-th basis vector, then we have

$$\xi^+ - \xi = \left(M - M^+ + e_j e_j^T\right)\ell = \left(M - \mathcal{R}_j(M) - \frac{1}{\tilde{A}_{jj}}\gamma\tilde{\gamma}^T + e_j e_j^T\right)\ell = -\frac{\tilde{\gamma}^T\ell}{\tilde{A}_{jj}}\gamma + (\ell_j - M_{jS}\ell_S)e_j$$

$$\Longrightarrow \xi^+ = \xi - \frac{\tilde{\gamma}^T\ell}{\tilde{A}_{jj}}\gamma + (\ell_j - M_{jS}\ell_S)e_j = \mathcal{R}_j(\xi) - \frac{\tilde{\gamma}^T\ell}{\tilde{A}_{jj}}\gamma.$$

Note that $\tilde{\gamma}^T\ell = \ell_j + M_{jS}\ell_S = -\xi_j$, we obtain that

$$\xi^+ = \mathcal{R}_j(\xi) + \frac{\xi_j}{\tilde{A}_{jj}}\gamma.$$

For D_ℓ^+, we have

$$D_\ell^+ = \mathbf{1}_{\tilde{S}}^T \xi_{\tilde{S}}^+ = D_\ell + \frac{\xi_j}{\tilde{A}_{jj}} \cdot \mathbf{1}_{\tilde{S}}^T \gamma_{\tilde{S}} = D_\ell + \frac{\xi_j \tilde{\eta}_j}{\tilde{A}_{jj}}.$$

■

The implementation of UPDATE_TILDE_EXPAND_SUPPORT is summarized in Algorithm 11.

Algorithm 11 UPDATE_UTILDE_EXPAND_SUPPORT

Inputs: Original support S, new index j, matrix A, vector ℓ, intermediate variables $\mathrm{Par}_1, \mathrm{Par}_3$.
Procedure:

1: $\tilde{A}_{jj} \leftarrow A_{jj} + M_{jS}A_{Sj}$;
2: $\gamma_{\tilde{S}} \leftarrow (M_{jS}, 1)^T, \gamma_{\tilde{S}^c} \leftarrow -A_{\tilde{S}^c j} - A_{\tilde{S}^c S}M_{jS}^T$;
3: $\tilde{\gamma}_{\tilde{S}} \leftarrow (M_{jS}, 1)^T, \tilde{\gamma}_{\tilde{S}^c} \leftarrow 0$;
4: $D \leftarrow D + \frac{\tilde{\eta}_j^2}{\tilde{A}_{jj}}$;
5: $D_\ell \leftarrow D_\ell + \frac{\xi_j \tilde{\eta}_j}{\tilde{A}_{jj}}$;
6: $\xi \leftarrow \mathcal{R}_j(\xi) + \frac{\xi_j}{\tilde{A}_{jj}}\gamma$;
7: $\tilde{\eta} \leftarrow \mathcal{R}_j(\tilde{\eta}) + \frac{\tilde{\eta}_j}{\tilde{A}_{jj}}\gamma$;
8: $M_{\cdot,\tilde{S}} \leftarrow \mathcal{R}_j(M_{\cdot,\tilde{S}}) + \frac{1}{\tilde{A}_{jj}}\gamma\tilde{\gamma}_{\tilde{S}}^T$.

Output: $\mathrm{Par}_1, \mathrm{Par}_3$.

C-4.3 UPDATE_UTILDE_SHRINK_SUPPORT

Since M is exactly the same as in Appendix B, we can directly apply Theorem B-7 to obtain an update of M and the updates of other parameters as a consequence.

Theorem C-10 *Let β and $\tilde{\beta}$ be defined in Theorem B-7, i.e.*

$$\beta_{\tilde{S}} = \tilde{\beta}_{\tilde{S}} = M_{\tilde{S}j}, \quad \beta_{\tilde{S}^c} = \begin{pmatrix} -1 \\ M_{S^c j} \end{pmatrix} \quad \tilde{\beta}_{\tilde{S}^c} = 0,$$

then

- $M^{-} = \mathcal{R}_j(M) - \frac{1}{M_{jj}} \cdot \beta\tilde{\beta}^T$;
- $\tilde{\eta}^{-} = \mathcal{R}_j(\tilde{\eta}) - \frac{\tilde{\eta}_j}{M_{jj}}\beta$;
- $D^{-} = D - \frac{\tilde{\eta}_j^2}{M_{jj}}$;
- $\xi^{-} = \mathcal{R}_j(\xi) - \frac{\xi_j}{M_{jj}}\beta$;
- $D_\ell^{-} = D_h - \frac{\xi_j\tilde{\eta}_j}{M_{jj}}$.

Proof The update of M, $\tilde{\eta}$ and D has been proved in Theorem B-7 and Theorem B-8. For any subset S, let I_S denote the matrix with j-th diagonal element equal to 1 for any $j \in S$ and all other elements equal to 0. Then ξ and ξ^{-} can be rewritten as

$$\xi = -\left(M + I_{S^c}\right)\ell, \quad \xi^{-} = -\left(M^{-} + I_{\tilde{S}^c}\right)\ell.$$

Note that $I_{\tilde{S}^c} - I_{S^c} = e_j e_j^T$ where e_j is the j-th basis vector, then we have

$$\begin{aligned}\xi^{-} - \xi &= \left(M - M^{-} - e_j e_j^T\right)\ell = \left(\frac{1}{M_{jj}}\beta\tilde{\beta}^T + M - \mathcal{R}_j(M) - e_j e_j^T\right)\ell \\ &= \frac{\beta_{\tilde{S}}^T \ell_{\tilde{S}}}{M_{jj}}\beta + \left(M - \mathcal{R}_j(M) - e_j e_j^T\right)\ell \triangleq \frac{\beta_{\tilde{S}}^T \ell_{\tilde{S}}}{M_{jj}}\delta + \tilde{\xi}.\end{aligned}$$

By definition of $\tilde{\xi}$

$$\tilde{\xi}_{\tilde{S}} = \ell_j M_{\tilde{S}j}, \quad \tilde{\xi}_j = M_{j\tilde{S}}\ell_{\tilde{S}} + \ell_j M_{jj} - \ell_j = -\xi_j - \ell_j, \quad \tilde{\xi}_{S^c} = \ell_j M_{S^c j},$$

and thus,

$$\tilde{\xi} = \ell_j\beta - \xi_j e_j.$$

This implies that

$$\xi^{-} = \xi - \xi_j e_j + \frac{\beta_{\tilde{S}}^T \ell_{\tilde{S}} + \ell_j M_{jj}}{M_{jj}}\beta = \mathcal{R}_j(\xi) - \frac{\xi_j}{M_{jj}}\beta.$$

For D_ℓ^{-}, we have

$$D_\ell^{-} = \mathbf{1}_{\tilde{S}}^T \xi_{\tilde{S}}^{-} = D_\ell - \xi_j - \frac{\xi_j}{M_{jj}} \cdot \mathbf{1}_{\tilde{S}}^T\beta_{\tilde{S}} = D_\ell - \frac{\xi_j(\mathbf{1}_{\tilde{S}}^T\beta_{\tilde{S}} + M_{jj})}{M_{jj}} = D_\ell - \frac{\xi_j\tilde{\eta}_j}{M_{jj}}.$$

■

The implementation of UPDATE_TILDE_SHRINK_SUPPORT is summarized in Algorithm 12.

Algorithm 12 UPDATE_UTILDE_SHRINK_SUPPORT

Inputs: Original support S, new index j, vector ℓ, intermediate variables $\mathrm{Par}_1, \mathrm{Par}_3$.

Procedure:

1: $\beta_{\tilde{S}} \leftarrow M_{j\tilde{S}}^T, \beta_{\tilde{S}^c} \leftarrow \begin{pmatrix} -1 \\ M_{\tilde{S}^c j} \end{pmatrix}, \tilde{\beta}_{\tilde{S}} \leftarrow M_{j\tilde{S}}^T, \tilde{\beta}_{\tilde{S}^c} \leftarrow 0$;

2: $D \leftarrow D - \frac{\tilde{\eta}_j^2}{M_{jj}}$;

3: $D_\ell \leftarrow D_\ell - \frac{\xi_j\tilde{\eta}_j}{M_{jj}}$;

4: $\xi \leftarrow \mathcal{R}_j(\xi) - \frac{\xi_j}{M_{jj}}\beta$;

5: $\tilde{\eta} \leftarrow \mathcal{R}_j(\tilde{\eta}) - \frac{\tilde{\eta}_j}{M_{jj}}\beta$;

6: $M_{\cdot,\tilde{S}} \leftarrow \mathcal{R}_j(M_{\cdot,\tilde{S}}) - \frac{1}{M_{jj}}\beta\tilde{\beta}_{\tilde{S}}^T, \ M_{\cdot,j} \leftarrow 0$.

Output: $\mathrm{Par}_1, \mathrm{Par}_3$.

Algorithm 13 DIRECT_UTILDE_UPDATE

Inputs: Support S, vector-update-vector ℓ, intermediate variables M.

Procedure:

1: $\xi_S \leftarrow -M_{SS}\ell_S,\ \xi_{S^c} \leftarrow -\ell_{S^c} - M_{S^cS}\ell_S$;

2: $D_\ell \leftarrow \mathbf{1}_S^T \xi_S$.

Output: ξ, D_ℓ.

C-4.4 DIRECT_UTILDE_UPDATE

At the beginning of each time t, we need to recompute ξ and D_ℓ. The implementation is summarized in Algorithm 13.

C-5 Complexity Analysis

Similar to Appendix B, we can analyze the computation complexity. The analysis here is much simpler than the last case since the implementation is quite straightforward. Table 2 summarizes the results.

Table 2: Computation complexity of each sub-routine in Algorithm 8.

	(Wz)-type	$(z\tilde{z}^T)$-type	(z^Tz)-type	(az)-type
FIND_UTILDE_LAMBDA	0	0	$2n$	0
UPDATE_BY_UTILDE_LAMBDA	0	0	0	n
UPDATE_UTILDE_EXPAND_SUPPORT	$(n-s-1)s$	$n(s+1)$	s	$2n$
UPDATE_UTILDE_SHRINK_SUPPORT	0	$n(s-1)$	0	$2n$
DIRECT_UTILDE_UPDATE	ns	0	s	0

Let k_r^+ and k_r^- be the number of turning points that S is expanded and shrinked respectively and $k_r = k_r^+ + k_r^-$ be the total number of tuning points, then the complexity is

$$C_{2t} \le ns + n + 3nk_r + n(2s+3)k_r^+ + n(s+1)k_r^- + O(k_r) = ns(2k_r+1) + n(6k_r+1) + O(k_r).$$

D Implementation of HONES Algorithm With Time-Varying A, r

D-1 Intermediate Variables

Based on the results in Appendix B and Appendix C, we can concatenate Algorithm 1 and Algorithm 8. Thus we define $\mathrm{Par}_1, \mathrm{Par}_2, \mathrm{Par}_3$ as $\mathrm{Par}_1 = \{M, \tilde{\eta}, D\}, \mathrm{Par}_2 = \{\eta, D_g, D_{gg}, D_{gr}\}, \mathrm{Par}_3 = \{\xi, D_\ell\}$ where all parameters are defined in previous appendices.

D-2 Implementation

Note that only two sub-routines involves the matrix A, namely UPDATE_EXPAND_SUPPORT and UPDATE_UTILDE_EXPAND_SUPPORT, and moreover they only involve the j-th column of A. Thus, we can use the sparse update of A as in Algorithm 7 for acceleration. Algorithm 14 below describes the implementation.

D-3 Complexity Analysis

The complexity of Algorithm 14 is just the sum of that of Algorithm 1 and Algorithm 8, i.e.

$$C_t = C_{1t} + C_{2t} = ns_* + ns(3k_A + 2k_r + 2) + n(12k_A + 6k_r + 3) + O(k_A + k_r).$$

Algorithm 14 HONES Algorithm for time-varing A, r with sparse update of A

Inputs: Initial parameters $A^{(0)}$, vectors $\{r^{(t)} : t = 1, 2, \ldots\}$,
matrix-update-vectors $\{g^{(t)}, t = 1, 2, \ldots\}$.

Initialization:

$x \leftarrow$ as the optimum corresponding to $A^{(0)}, r^{(0)}$.
$S \leftarrow \mathrm{supp}(x), S_* \leftarrow S$;
Calculate (x, μ, μ_0) via (8)-(10)
$v \leftarrow (x_S, -\mu_{S^c})$;
Calculate intermediate variables $(\mathrm{Par}_1, \mathrm{Par}_2)$ via (B-1)-(B-3) based on $r^{(0)}, g^{(1)}$;

Procedure:

for $t = 1, 2, \cdots$. **do**
  $\lambda \leftarrow 0$;
  while $\lambda < 1$ **do**
    $(\lambda^{\mathrm{inc}}, j, S^{\mathrm{new}}) \leftarrow \mathrm{FIND\_LAMBDA}(S, v; \mathrm{Par}_1, \mathrm{Par}_2)$;
    $\lambda^{\mathrm{inc}} \leftarrow \min\{\lambda^{\mathrm{inc}}, 1 - \lambda\}$;
    $\lambda \leftarrow \lambda + \lambda^{\mathrm{inc}}$;
    $(v, \mu_0; \mathrm{Par}_1, \mathrm{Par}_2) \leftarrow \mathrm{UPDATE\_BY\_LAMBDA}(\lambda^{\mathrm{inc}}; v, \mu_0, \mathrm{Par}_1, \mathrm{Par}_2)$;
    if $S^{\mathrm{new}} = S \cup \{j\}$ **then**
      $(\mathrm{Par}_1, \mathrm{Par}_2) \leftarrow \mathrm{UPDATE\_EXPAND\_SUPPORT}(\lambda, S, j; A, r^{(t-1)}, g^{(t)}, \mathrm{Par}_1, \mathrm{Par}_2)$;
      if $j \notin S_*$ **then**
        $G \leftarrow (g^{(1)}, \ldots, g^{(t-1)})$;
        $A_{\cdot,j} \leftarrow A_{\cdot,j} + GG^T_{j,\cdot}$;
        $S_* = S_* \cup \{j\}$;
      end if
    else if $S^{\mathrm{new}} = S \setminus \{j\}$ **then**
      $(\mathrm{Par}_1, \mathrm{Par}_2) \leftarrow \mathrm{UPDATE\_SHRINK\_SUPPORT}(S, j; r^{(t-1)}, g^{(t)}, \mathrm{Par}_1, \mathrm{Par}_2)$;
    end if
    $S \leftarrow S^{\mathrm{new}}$;
  end while
  $A_{\cdot,S_*} \leftarrow A_{\cdot,S_*} + g^{(t)}(g^{(t)}_{S_*})^T$;
  $\ell^{(t)} \leftarrow r^{(t)} - r^{(t-1)}$;
  $\mathrm{Par}_3 \leftarrow \mathrm{DIRECT\_UTILDE\_UPDATE}(S, \mathrm{Par}_1, \ell^{(t)})$;
  $\underset{\sim}{\lambda} \leftarrow 0$;
  while $\underset{\sim}{\lambda} < 1$ **do**
    $(\underset{\sim}{\lambda}^{\mathrm{inc}}, j, S^{\mathrm{new}}) \leftarrow \mathrm{FIND\_UTILDE\_LAMBDA}(v; \mathrm{Par}_1, \mathrm{Par}_3)$;
    $\underset{\sim}{\lambda}^{\mathrm{inc}} \leftarrow \min\{\underset{\sim}{\lambda}^{\mathrm{inc}}, 1 - \underset{\sim}{\lambda}\}$;
    $(v, \mu_0) \leftarrow \mathrm{UPDATE\_BY\_UTILDE\_LAMBDA}(\underset{\sim}{\lambda}^{\mathrm{inc}}; v, \mu_0, \mathrm{Par}_1, \mathrm{Par}_3)$;
    if $S^{\mathrm{new}} = S \cup \{j\}$ **then**
      $(\mathrm{Par}_1, \mathrm{Par}_3) \leftarrow \mathrm{UPDATE\_UTILDE\_EXPAND\_SUPPORT}(S, j, A, \ell^{(t)}; \mathrm{Par}_1, \mathrm{Par}_3)$;
      if $j \notin S_*$ **then**
        $G \leftarrow (g^{(1)}, \ell^{(t)} dots, g^{(t-1)})$;
        $A_{\cdot,j} \leftarrow A_{\cdot,j} + GG^T_{j,\cdot}$;
        $S_* = S_* \cup \{j\}$;
      end if
    else if $S^{\mathrm{new}} = S \setminus \{j\}$ **then**
      $(\mathrm{Par}_1, \mathrm{Par}_3) \leftarrow \mathrm{UPDATE\_UTILDE\_SHRINK\_SUPPORT}(S, j, \ell^{(t)}; \mathrm{Par}_1, \mathrm{Par}_3)$;
    end if
    $S \leftarrow S^{\mathrm{new}}$;
    $\underset{\sim}{\lambda} \leftarrow \underset{\sim}{\lambda} + \underset{\sim}{\lambda}^{\mathrm{inc}}$.
  end while
  $\mathrm{Par}_2 \leftarrow \mathrm{DIRECT\_UPDATE}(S, r^{(t)}, g^{(t+1)}; \mathrm{Par}_1, \mathrm{Par}_2)$;
  $x^{(t)}_S \leftarrow x_S, \quad x^{(t)}_{S^c} \leftarrow 0$.
end for

Output: $x^{(1)}, x^{(2)}, \cdots$.

E Dealing With General Linear Constraints

Now we consider the problem with general linear constraints:

$$\min \frac{1}{2} x^T A^{(t)} x - (r^{(t)})^T x, \quad s.t. \quad Bx = b, x \geq 0$$

where $B \in \mathbb{R}^{m \times n}$ and $A^{(t)}$ is a stream with

$$A^{(t+1)} = A^{(t)} + g^{(t)} (g^{(t)})^T.$$

Similar to (8)-(10), the KKT condition for a single problem (with the supscript (t) erased temporarily) can be written as

$$Ax - B^T \mu_0 - \mu - r = 0; \tag{E-13}$$

$$Bx = b; \tag{E-14}$$

$$\mu_i x_i = 0, \mu_i \geq 0, x_i \geq 0, \forall i = 1, \ldots, n, \tag{E-15}$$

where $\mu_0 \in \mathbb{R}^m, \mu \in \mathbb{R}^n$. Partitioning the matrix A by the support of x, we have

$$\begin{aligned} &\begin{pmatrix} A_{SS} & A_{SS^c} \\ A_{S^cS} & A_{S^cS^c} \end{pmatrix} \begin{pmatrix} x_S \\ 0 \end{pmatrix} \\ = &\begin{pmatrix} B_S^T \\ B_{S^c}^T \end{pmatrix} \mu_0 + \begin{pmatrix} 0 \\ \mu_{S^c} \end{pmatrix} + \begin{pmatrix} r_S \\ r_{S^c} \end{pmatrix}. \end{aligned}$$

As a consequence, we can solve x_S and μ_{S^c} by

$$x_S = A_{SS}^{-1} B_S^T \mu_0 + A_{SS}^{-1} r_S; \tag{E-16}$$

$$\mu_{S^c} = A_{S^cS} x_S - B_{S^c}^T \mu_0 - r_{S^c} = -(B_{S^c}^T - A_{S^cS} A_{SS}^{-1} B_S^T) \mu_0 - (r_{S^c} - A_{S^cS} A_{SS}^{-1} r_S). \tag{E-17}$$

Using the constraint E-14, we can solve μ_0 as

$$\mu_0 = (B_S A_{SS}^{-1} B_S^T)^{-1} (b - B_S A_{SS}^{-1} r_S). \tag{E-18}$$

Noticing that (E-16)-(E-18) share similar forms to (8)-(10), we can easily derive a counterpart of Theorem E-11.

Theorem E-11

1. *Fixing $\underset{\sim}{\lambda}$, there exists vectors $u_1, u_2 \in \mathbb{R}^{n+m}$ and scalars $D_1, D_2 \in \mathbb{R}$, which only depend on S, such that*

$$\begin{pmatrix} x_S(\lambda) \\ -\mu_{S^c}(\lambda) \\ \mu_0(\lambda) \end{pmatrix} = \frac{u_1 - u_2 \lambda}{D_1 - D_2 \lambda}. \tag{E-19}$$

2. *Fixing λ, there exists vectors $\underset{\sim}{u}_1, \underset{\sim}{u}_2 \in \mathbb{R}^{n+m}$, which only depend on S, such that*

$$\begin{pmatrix} x_S(\underset{\sim}{\lambda}) \\ -\mu_{S^c}(\underset{\sim}{\lambda}) \\ \mu_0(\underset{\sim}{\lambda}) \end{pmatrix} = \underset{\sim}{u}_1 - \underset{\sim}{u}_2 \underset{\sim}{\lambda}. \tag{E-20}$$

Proof

1. By Lemma B-1, we have

$$A_{SS}(\lambda)^{-1} = A_{SS}^{-1} - \alpha(\lambda) g_S g_S^T$$

and

$$A_{S^cS}(\lambda) A_{SS}(\lambda)^{-1} = A_{S^cS} A_{SS}^{-1} - \alpha(\lambda) g_{S^c} g_S^T.$$

As a result,

$$\mu_0(\lambda) = \left(B_S A_{SS}^{-1} B_S^T - \alpha(\lambda) \cdot B_S g_S (B_S g_S)^T\right)^{-1} (b - B_S A_{SS}^{-1} r_S + (g_S^T r_S) \alpha(\lambda) \cdot B_S g_S)$$

$$
\begin{aligned}
&= \left((B_S A_{SS}^{-1} B_S^T)^{-1} + \frac{\alpha(\lambda)(B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S g_S^T B_S^T (B_S A_{SS}^{-1} B_S^T)^{-1}}{1 - \alpha(\lambda) g_S^T B_S^T (B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S} \right) \cdot \\
&\quad (b - B_S A_{SS}^{-1} r_S + (g_S^T r_S) \alpha(\lambda) \cdot B_S g_S) \\
&= \mu_0 + \frac{\alpha(\lambda)(B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S g_S^T B_S^T \mu_0}{1 - \alpha(\lambda) g_S^T B_S^T (B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S} \\
&\quad + \frac{\alpha(\lambda)(B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S}{1 - \alpha(\lambda) g_S^T B_S^T (B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S} \cdot g_S^T r_S \\
&= \mu_0 + \frac{\alpha(\lambda)(B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S}{1 - \alpha(\lambda) g_S^T B_S^T (B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S} \cdot g_S^T (B_S^T \mu_0 + r_S).
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
x_S(\lambda) &= \left(A_{SS}^{-1} - \alpha(\lambda) g_S g_S^T \right) B_S^T \left(\mu_0 + \frac{\alpha(\lambda)(B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S}{1 - \alpha(\lambda) g_S^T B_S^T (B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S} \cdot g_S^T (B_S^T \mu_0 + r_S) \right) \\
&\quad + \left(A_{SS}^{-1} - \alpha(\lambda) g_S g_S^T \right) r_S \\
&= x_S - \alpha(\lambda) g_S \cdot g_S^T (B_S^T \mu_0 + r_S) + \\
&\quad + \left(A_{SS}^{-1} - \alpha(\lambda) g_S g_S^T \right) \cdot \frac{\alpha(\lambda) B_S^T (B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S}{1 - \alpha(\lambda) g_S^T B_S^T (B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S} \cdot g_S^T (B_S^T \mu_0 + r_S) \\
&= x_S - \frac{\alpha(\lambda)}{1 - \alpha(\lambda) g_S^T B_S^T (B_S A_{SS}^{-1} B_S^T)^{-1} B_S g_S} \cdot g_S^T (B_S^T \mu_0 + r_S) \cdot \left(I - A_{SS}^{-1} B_S^T (B_S A_{SS}^{-1} B_S^T)^{-1} B_S \right) g_S,
\end{aligned}
$$

and

$$
\mu_{S^c}(\lambda) = -\left(B_{S^c}^T - A_{S^c S} A_{SS}^{-1} B_S^T + \alpha(\lambda) g_{S^c} g_S^T B_S^T \right)
$$

■