algorithm for computing the OG by iteration and projection

zh296

March 2021

1 Derivation

We start with the heat equation from equation 1.3 in "Drew's First Writeup". That equation states

$$M\dot{u} + Au = Bz \tag{1}$$

To make the notation more compatible with our current paper, we rewrite the equation with different letters and invert the mass matrix. Now, the differential equation is,

$$\begin{aligned} \dot{x}(t) &= -M^{-1}Ax(t) + M^{-1}Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{2}$$

Based on the above equation, the Lyapunov equation we want to solve is

$$(-M^{-1}A)P + P(-M^{-1}A)^T + CC^T = 0 \tag{3}$$

The P in the above equation is the observability gramian.

Now, based on the "low rank smith" paper, we can rewirte the above equation as

$$\begin{aligned} P = {} & (-M^{-1}A - \mu I)(-M^{-1}A + \mu I)^{-1}P(-M^{-1}A + \mu I)^{-1T}(-M^{-1}A - \mu I)^T \\ & - 2\mu(-M^{-1}A + \mu I)^{-1}CC^T((-M^{-1}A)^T + \mu I)^{-1} \end{aligned} \tag{4}$$

Let's denote $A_\mu = (-M^{-1}A - \mu I)(-M^{-1}A + \mu I)^{-1}$. Then, we can simplify the above equation as

$$\begin{aligned} P = {} & A_\mu P A_\mu^T \\ & - 2\mu(-M^{-1}A + \mu I)^{-1}CC^T((-M^{-1}A)^T + \mu I)^{-1} \end{aligned} \tag{5}$$

Now, suppose we are inside the greedy algorithm and want to update the observability gramian with just one more sensor. We can write $P_{next} = P_{pre} + D$.

$$
\begin{aligned}
&P_{pre} + D = A_\mu (P_{pre} + D) A_\mu^T \\
&\quad - 2\mu(-M^{-1}A + \mu I)^{-1}(C_{pre} + C_{rank1})(C_{pre} + C_{rank1})^T((-M^{-1}A)^T + \mu I)^{-1}
\end{aligned} \tag{6}
$$

When we do $(C_{pre} + C_{rank1})(C_{pre} + C_{rank1})^T$, the cross terms will cancel out. Furthermore, we know P_{pre} satisfies the Stein equation, which is equation 3.4 in the "low rank smith" paper. Therefore, we have

$$
\begin{aligned}
&D = A_\mu D A_\mu^T \\
&\quad - 2\mu(-M^{-1}A + \mu I)^{-1} C_{rank1} C_{rank1}^T ((-M^{-1}A)^T + \mu I)^{-1}
\end{aligned} \tag{7}
$$

Now, we use the infinite series expansion given in the stanford lecture,

$$
D = \sum_{t=0}^{\infty} A_\mu^t (-2\mu(-M^{-1}A + \mu I)^{-1} C_{rank1} C_{rank1}^T ((-M^{-1}A)^T + \mu I)^{-1})(A_\mu^T)^t \tag{8}
$$

or,

$$
D = -2\mu \sum_{t=0}^{\infty} A_\mu^t (-M^{-1}A + \mu I)^{-1} C_{rank1} C_{rank1}^T ((-M^{-1}A)^T + \mu I)^{-1}(A_\mu^T)^t \tag{9}
$$

Now, we can write the D as

$$
D = -2\mu(aa^T + bb^T + ...) = -2\mu A_{sylvester} B_{sylvester} \tag{10}
$$

Now, we have

$$
\begin{aligned}
det(P - 2\mu A_{sylester} B_{sylvester}) &= det(P(I - 2\mu P^{-1} A_{sylvester} B_{sylvester})) \\
&= det(P) det(I - 2\mu P^{-1} A_{sylvester} B_{sylvester}) \\
&= det(P) det(I - 2\mu B_{sylvester} P^{-1} A_{sylvester})
\end{aligned} \tag{11}
$$

Now, a problem with the current formulation is that P is a low rank matrix and P^{-1} is numerically low rank. Therefore, instead of using $logdet(P_{next})$ as the objective function, we propose the following. In each iteration of the greedy algorithm, we already know P_{pre} and can use its top eigenvectors form a subspace. We know that P_{pre} in this subspace spanned by its own top eigenvectors can be inverted stably. Therefore, the new objective is $U'_{pre} P_{next} U_{pre}$, where U_{pre} is obtained by performing a diagonalization of $P_{pre} = U_{pre} \Sigma_{pre} U'_{pre}$. Therefore,

$$
\begin{aligned}
Objective &= det(U'_{pre} P_{next} U_{pre}) \\
&= det(U'_{pre}(P_{pre} + D) U_{pre}) \\
&= det(\Sigma_{pre,top} + U'_{pre} D U_{pre}) \\
&= det(\Sigma_{pre,top} - 2\mu U'_{pre} A_{sylester} B_{sylvester} U_{pre}) \\
&= det(\Sigma_{pre,top}) det(I - 2\mu B_{sylvester} U_{pre} \Sigma_{pre,top}^{-1} U'_{pre} A_{sylvester})
\end{aligned} \tag{12}
$$

2 Implementation Detail

We notice that $A_\mu, (-M^{-1}A + \mu I)^{-1}$, and $(-M^{-1}A)^T + \mu I)^{-1}$ are the same throughout the entire process. Therefore, we can compute those three matrices in advance. Furthermore, we can diagonalize $A_\mu = A_{\mu v} A_{\mu d} A_{\mu v}^{-1}$

Now, we can write the D as

$$D = -2\mu(aa^T + bb^T + ...) \tag{13}$$

Furthermore, we have

$$D = -2\mu A_{sylvester} B_{sylvester} \tag{14}$$

$$A_{sylvester} = \begin{bmatrix} | & | & \\ a & b & \dots \\ | & | & \end{bmatrix}$$

$$B_{sylvester} = \begin{bmatrix} — & a^T & — \\ — & b^T & — \\ & \vdots & \end{bmatrix}$$

Algorithm 1: Optimal Sensor Placement With Approximate Observability Gramian

1.Initialize the algorithm with a very small set of sensors, *selected_obs*.
2.Set μ and max iteration for the Low Rank Smith method.
3.Set max terms for the infinite series and projection rank cutoff.

4.Compute A_μ, which is $A_\mu = (-M^{-1}A - \mu I)(-M^{-1}A + \mu I)^{-1}$.
5.Diagonalize A_μ so that$A_\mu = V_{A_\mu} D_{A_\mu} V_{A_\mu}^{-1}$.
6.Compute $L = (-M^{-1}A + \mu I)^{-1}$ and $R = (-M^{-1}A)^T + \mu I)^{-1}$.

for *i = 0,1,2,...max sensor* **do**
  1. Use the Low Rank Smith method to diagonalize the current observability gramian, Q, such that $Q = U_{pre}\Sigma_{pre}U_{pre}$

  for *j not in selected_obs* **do**
    1. Initialize $A_{sylvester,j} = L(:,j)$ and denote $L(;,j) = l_j$
    2. Initialize $B_{sylvester,j} = R(j,:)$ and denote $R(j,:) = r_j$

    for *k = 1,... max terms* **do**
      1. $A_{sylvester,column\_k} = V_{A_\mu} D_{A_\mu}^k V_{A_\mu}^{-1} l_j$.
      2. Append $A_{sylvester,column\_k}$ as a new column to $A_{sylvester,j}$.
      3. $B_{sylvester,row\_k} = r_j V_{A_\mu}^{-1\prime} D_{A_\mu}^k V_{A_\mu}'$.
      4. Append $B_{sylvester,row\_k}$ as a new row to $B_{sylvester,j}$.
    end

    3. Compute the objective function, $det(I - 2\mu B_{sylvester,j} U_{pre,top} \Sigma_{pre,top}^{-1} U_{pre,top}' A_{sylvester,j})$
  end
  2. Select the ith sensor location based on the maximum value of the objective function
  3. Add this selected sensor to *selected_obs*
end

Now, let's see the time complexity of the algorithm for adding one more sensor. The time complexity of diagonalizing the observability gramian at the first step of the i loop is bounded by $O(n^3)$

The time complexity of computing a term in the infinity series is $O(n^2)$ because l_j and r_j are rank 1 vectors. The rank of $I-2\mu B_{sylvester,j} U_{pre,top} \Sigma_{pre,top}^{-1} U_{pre,top}' A_{sylvester,j}$ is determined by the maximum number of terms in the infinity series expansion, and the time complexity of forming this matrix is $O(max\_terms\, n\, projection\_rank)$.

Therefore, the overall time complexity for computing one objective function is bounded by $O(kn^2)$, and k is the maximum number of terms we choose in the infinity series. Therefore, the overall time complexity of adding a new sensor is $O(kn^3)$. If we choose k to be 1, which seems to work. Then the time complexity of finding a new sensor is $O(n^3)$. The most naive method is $O(n^4)$.