Sherman-Morrison-Woodbury formula

Consider a non-singular n x n matrix A

$$B = A + uv^T$$

where u and v are n x 1 vectors. The 'outer' product of u and v is an n x n matrix of rank one. If we have computed the inverse of A, is there a short-cut for the computation of B?

The Sherman-Morrison-Woodbury formula shows how to update the inverse of a matrix altered by the addition of a rank-one matrix. This is also called the 'Matrix Inversion Lemma'.

Contents

Sherman Morrison formula

Given

$$B = A + uv^T$$

$$B^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}$$

where

$$1 + v^TA^{-1}u \neq 0$$

Why it is Useful

The Sherman-Morrison formula tells us that

- a rank-one change in a matrix results in a rank-one change in the inverse of that matrix.
- inverse matrix can be computed in n^2 operations rather than the order n^3 operations need to recompute the matrix inverse from scratch.

Matlab Demonstration

Construct a random matrix

```
A = randn(5,5);
Ainv = inv(A);
disp(Ainv)
```

```
    0.5974    0.7559   -0.1153   -0.0996    1.0363
   -0.3990   -0.2881    0.1566    0.4003   -1.0541
   -0.0133    0.6792   -0.6955   -1.0380   -1.0665
    0.4606   -0.4536    0.4473    0.4202   -0.4494
    1.2654   -0.0002   -0.4257    0.5346    0.8757
```

Rank-one Update

Here is a rank-one perturbation

```
u = randn(5,1);
v = randn(5,1);
B = A + u*v';
disp(inv(B))
```

```
    0.1002    0.9766   -0.1297   -0.5860    1.5737
    0.4253   -0.6538    0.1805    1.2068   -1.9451
    2.9463   -0.6341   -0.6100    1.8578   -4.2656
   -0.4701   -0.0406    0.4204   -0.4905    0.5566
   -0.3539    0.7183   -0.4725   -1.0497    2.6260
```

Computing Inverse of B

Note the grouping of operations used to exploit the rank-one nature of the Sherman-Morrison Formula

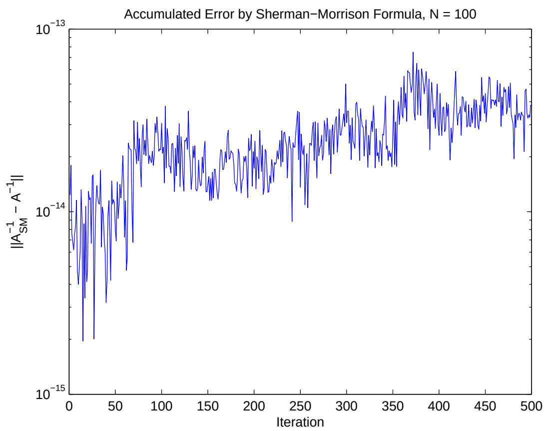
```
d = 1 + v'*Ainv*u;
Binv = Ainv - (Ainv*(u/d))*(v'*Ainv);

disp(Binv);
norm(Binv - inv(B))
```

```
    0.1002    0.9766   -0.1297   -0.5860    1.5737
    0.4253   -0.6538    0.1805    1.2068   -1.9451
    2.9463   -0.6341   -0.6100    1.8578   -4.2656
   -0.4701   -0.0406    0.4204   -0.4905    0.5566
   -0.3539    0.7183   -0.4725   -1.0497    2.6260


ans =

   5.9759e-15
```

Sherman-Morrison Formula is Unstable

One problem with the Sherman-Morrison Formula is that the approximation error grows with repeated use.

```
n = 100;
A = 100*randn(n,n);
Ainv = inv(A);
e = [];
for k = 1:500
    u = rand(n,1);
    v = rand(n,1);
    A = A + u*v';
    Ainv = Ainv - (Ainv*(u/(1+v'*Ainv*u)))*(v'*Ainv);
    e(k) = norm(Ainv - inv(A));
end

semilogy(e);
title(sprintf('Accumulated Error by Sherman-Morrison Formula, N = %3d',n'));
xlabel('Iteration');
ylabel('||A^{-1}_{SM} - A^{-1}||');
fig4pdf;
```



Application to Linear Programming

Linear Program

$$min_x f = c^T x$$
$$Ax \geq b$$

The Active Set Method is characterized by the updating a matrix of active constraints. At each iteration one of the currently active constraints is removed, and replaced by one of the currently inactive contraints.

Analysis

The pth row of the active constraints is replaced by the qth constraint:

$$A_{\mathcal{A}} + \underbrace{e_p(a_q^T - r_p^T)}_{rank-one\, update}$$

where e_p is a vector with 1 in the pth position and zeros everywhere else, and

$$A_{\mathcal{A}} = \begin{bmatrix} r_1^T \\ r_2^T \\ \vdots \\ r_n^T \end{bmatrix}$$

Some Observations

There are some very useful simplifications for the application

- $A_{\mathcal{A}}^{-1} e_p = d_p$ is the pth column of the inverse – the search direction in the active set method
- $r_p^T A_{\mathcal{A}}^{-1} = e_p^T$
- With these simplifications, we get $1 + (a_q^T - r_p^T) A_{\mathcal{A}}^{-1} e_p = a_q^T d_p$

Ultimately

$$(A_{\mathcal{A}} + e_p(a_q^T - r_p^T))^{-1} = A_{\mathcal{A}}^{-1} - \frac{d_p(a_q^T A_{\mathcal{A}}^{-1} - e_p^T)}{a_q^T d_p}$$

Matrix Determinant Lemma

The 'Matrix Determinant Lemma' is closely related to the Sherman-Morrison-Woodbury formula. Provided A^{-1} exists,

$$\det(A + uv^T) = \det(A)(1 + u^T A^{-1} v)$$

This situation comes up in state feedback control $u = -kx$ for a single-input system $\frac{dx}{dt} = Ax + bu$. The characteristic equation is then

$$\det(\lambda I - A + bk) = \det(\lambda I - A)(1 + k(\lambda I - A)^{-1} b)$$

Specifying n eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ for the closed-loop provides set of n linear equations for k

$$-1 = k(\lambda_i I - A)^{-1} b \qquad i = 1, 2, \ldots, n$$