Subject: Re: pseudo code for doing SVD on 2D sparse array
Posted by spellucci on Sun, 21 Dec 2008 16:30:22 GMT
View Forum Message <> Reply to Message

In article <fb7c56d7-4124-4228-949b-aeb66116d298@t39g2000prh.googlegroups.com>,
 Brian Borchers <borchers.brian@gmail.com> writes:
> On Dec 21, 4:41 am, erano <eran.o...@gmail.com> wrote:
>> Hi,
>> I wish to solve Ax=B
>> A is sparse array (size m*n), in format of [x_index, y_index, value]
>> B is vector length m
>> x is unknown vector length n
>> n=1,000,000
>> m=2*n
>>
>
> The title of your posting refers to the SVD, but the body of the
> posting indicates that you want to solve a linear system of equations,
> perhaps in the least squares sense.
>

this is only the (unfortunately usual) sloppy kind to write down a
linear least squares problem

> Unfortunately, computing the SVD of your 2,000,000 by 1,000,000 sparse
> matrix is utterly impractical- it would require the storage of a
> 1,000,000 by 1,000,000 fully dense matrix and a 2,000,000 by 2,000,000
> fully dense matrix, which would take up about 2.4e13 bytes of
> storage...
>
> Finding a least squares solution to the system of equation should
> probably be done using an iterative method such as lsqr. In order to
> do this, you'll first want to convert your data into a MATLAB sparse
> matrix with
>
> As=sparse(A(:,1),A(:,2),A(:,3));
>
> Then solve with
>
> x=lsqr(As,b);
>
> Since your matrix is extremely large, this could take a long time or
> simply fail to converge. If so, you might want to loosen the default
> tolerance, introduce a preconditioner, etc. The documentation on lsqr
> explains how to do these things.

without matlab:
lsqr is available also through netlib (f77 code) but

what about netlib/svdpack, which has code just for this problem?
lsqr for such a large column space might run into trouble.

hth
peter
