Subject: diagonal dominant
Posted by Gompie on Wed, 30 Jan 2013 14:26:11 GMT
View Forum Message <> Reply to Message

Hi,
I have a matrix(256,256) and I want to convert it into diagonal dominant form.
What is the best way to do it. Does IDL have function to do it?
Gompie

Subject: Re: diagonal dominant
Posted by Brian Daniel on Wed, 30 Jan 2013 17:00:48 GMT
View Forum Message <> Reply to Message

On Wednesday, January 30, 2013 9:26:11 AM UTC-5, Gompie wrote:
> Hi,
>
> I have a matrix(256,256) and I want to convert it into diagonal dominant form.
>
> What is the best way to do it. Does IDL have function to do it?
>
> Gompie

Well, if you're building a matrix by hand, try DIAG_MATRIX. You can put values along the diagonal, or offset them by a set number of rows. For example:

```
print,diag_matrix([3,4,5],1)
      0       3       0       0
      0       0       4       0
      0       0       0       5
      0       0       0       0
```

-Brian

Subject: Re: diagonal dominant
Posted by bstecklu on Wed, 30 Jan 2013 17:06:05 GMT
View Forum Message <> Reply to Message

Gompie wrote:
> Hi,
> I have a matrix(256,256) and I want to convert it into diagonal dominant form.
> What is the best way to do it. Does IDL have function to do it?
> Gompie
You asked about the same question a few posts above, and were given advice. So what was the problem then which prevented to get a solution?

Bringfried

Subject: Re: diagonal dominant
Posted by Gompie on Wed, 30 Jan 2013 17:21:13 GMT
View Forum Message <> Reply to Message

The solution of AX=B, I got from svd (suggested earlier) is noisy. (perhaps it is an approximation). So I now want to construct A so that idl can compute its determinant value (and its inverse precisely) and the solution is more robust(i.e condition number is better).

Subject: Re: diagonal dominant
Posted by on Wed, 30 Jan 2013 23:13:01 GMT
View Forum Message <> Reply to Message

Den onsdagen den 30:e januari 2013 kl. 18:21:13 UTC+1 skrev Gompie:
> The solution of AX=B, I got from svd (suggested earlier) is noisy. (perhaps it is an approximation). So I now want to construct A so that idl can compute its determinant value (and its inverse precisely) and the solution is more robust(i.e condition number is better).

What do you mean by noisy?

In that previous thread (it's actually better if you continue your older thread if you are discussing the same problem) you said the equations system can be over determined but not under determined. How does that work if your matrix is square? Then you have as many equations as unknowns so how can it be over determined? It could be under determined if you have linear dependencies.

You also said "I have checked for duplicate rows and columns in A they have been removed", presumably with that routine you got from me. I can understand the removal of identical rows, at least if the corresponding element in the RHS vector is also a duplicate, because this corresponds to removing identical equations. But why would you want to remove identical columns? That corresponds to removing one of the unknowns, is that what you want to do?

As for the removal of duplicate equations, it is really unnecessary if you solve the system with SVD methods. SVD can handle linear dependencies and removing duplicate equations does not guarantee that your system is free from them anyway.

Subject: Re: diagonal dominant
Posted by Gompie on Thu, 31 Jan 2013 03:36:19 GMT
View Forum Message <> Reply to Message

Let me pose the question in better words.

I want to get the value of X in AX=B. Since X has 256 unknowns, I have selected 256 rows of A

from a database of many rows which i can generate by increasing the input range of my model. This selection is done after removing duplicate rows because duplicate rows will make matrix A singular.

So I have the flexibility of choosing any 256 non duplicate rows which can work best for me. I am not removing identical columns in this problem.

By noisy i mean two things.
1. The solution is large error bars
2. A slight change in B (without any change in A) makes the solution. This means that condition number of A should be ok and that it should be invertible

So the problem now is how make a selection so that A is invertible. But before that in idl when I give determ(A) it get 0.0000 Floating point error. So I am not sure if determinant is properly calulated A inverse is anyway wrong.

Subject: Re: diagonal dominant
Posted by Gompie on Thu, 31 Jan 2013 06:33:57 GMT
View Forum Message <> Reply to Message

Thanks again Mats !!!
As suggested by you i did not do any removal of duplicates and used svd..results are a bit better but still there is noise. Are there any variants or alternatives of svd that I can try and see if noise reduces.

It does appear that the X values are following a curve.By noise I mean that the computed X has some errors but when I make a polynomial fit of values in X, the fit resembles the solution.

Subject: Re: diagonal dominant
Posted by on Thu, 31 Jan 2013 08:09:33 GMT
View Forum Message <> Reply to Message

Den torsdagen den 31:e januari 2013 kl. 07:33:57 UTC+1 skrev Gompie:
> Thanks again Mats !!!
>
> As suggested by you i did not do any removal of duplicates and used svd..results are a bit better but still there is noise. Are there any variants or alternatives of svd that I can try and see if noise reduces.

SVD should be numerically better than any Gauss elimination scheme and it can handle linear dependencies, so you should probably stick with it.

Can you describe how you are using the SVD to solve your equation? You have mentioned calculating the inverse of A but that is not necessary. From what I've read, it is better to multiply the B vector by the inverse SVD components in succession than calculating the inverse of A and

then multiplying B with that.

Also, if your matrix is singular (or near-singular) are you zeroing any of the inverted singular values? What cutoff are you using for that? Are you using single or double precision?

Did you know that the SVD gives you the condition number as the ratio between the smallest and the largest singular values? If that determinant you are calculating really is zero your matrix is singular and you should see that in the condition number (or in the fact that the smallest singular values are zero or at least very small compared to the largest singular value.

> It does appear that the X values are following a curve.By noise I mean that the computed X has some errors but when I make a polynomial fit of values in X, the fit resembles the solution.

I have no idea what you meant by that. Are you first calculating a solution X to the equation AX=B and then fitting a polynomial to that X? If that polynomial fit is the end result you need, why don't you formulate your model so you can fit the polynomial directly to the measurements in B?

Subject: Re: diagonal dominant
Posted by Gompie on Thu, 31 Jan 2013 13:05:46 GMT
View Forum Message <> Reply to Message

> Can you describe how you are using the SVD to solve your equation? You have mentioned calculating the inverse of A but that is not necessary. From what I've read, it is better to multiply the B vector by the inverse SVD components in succession than calculating the inverse of A and then multiplying B with that

I am now using

svdc,A ,w,u,v,/double
X=SVSOL(U, W, V, rhs,/double), Here A is 255X400 matrix and the duplicate rows have not been removed. Do you think I should do it differently?

> I have no idea what you meant by that. Are you first calculating a solution X to the equation AX=B and then fitting a polynomial to that X? If that polynomial fit is the end result you need, why don't you formulate your model so you can fit the polynomial directly to the measurements in B?

Thats a good idea I can try that !!

Subject: Re: diagonal dominant
Posted by on Thu, 31 Jan 2013 13:26:18 GMT
View Forum Message <> Reply to Message

Den torsdagen den 31:e januari 2013 kl. 14:05:46 UTC+1 skrev Gompie:
>> Can you describe how you are using the SVD to solve your equation? You have mentioned

calculating the inverse of A but that is not necessary. From what I've read, it is better to multiply the B vector by the inverse SVD components in succession than calculating the inverse of A and then multiplying B with that
>
>
>
> I am now using
>
>
>
> svdc,A ,w,u,v,/double
>
> X=SVSOL(U, W, V, rhs,/double), Here A is 255X400 matrix and the duplicate rows have not been removed. Do you think I should do it differently?

From the documentation of svsol: "An n-element vector containing "singular values." Normally, W is returned from the SVDC procedure. Small values (close to machine floating-point precision) should be set to zero prior to calling SVSOL."

You don't mention doing this so I assume you don't. This is a crucial step if you want to be able to handle singular or near-singular matrices.

Note also that another source of confusion is the way the A matrix is represented in IDL. I don't know what conventions you are using but SVDC assumes row-major. You can change that with the COLUMN keyword.
