Subject: Re: the (Moore-Penrose) pseudo-inverse of a matrix - anything like scipy.linalg's pinv2 in IDL?
Posted by JP on Wed, 03 Apr 2013 23:06:51 GMT
View Forum Message <> Reply to Message

Thanks Heinz,

After my post yesterday I tested Paul's svd_matrix_invert comparing with scipy's pinv2 and it looks like they do the same. I added a rcond keyword too and it also mimics pinv2 behaviour.
From a quick look to your code it looks like it's also doing the same thing but haven't tested.

cheers

Juan

On Thursday, 4 April 2013 00:33:36 UTC+11, Heinz Stege wrote:
> On Tue, 2 Apr 2013 22:18:16 -0700 (PDT), JP wrote:
>
>
>
>> Is that an equivalent to the scipy pinv2 i am looking for? And if so, I will appreciate if someone will better algebra skills than me (likely 95% of this community) could suggest how to introduce the rcond keyword available in pinv2.
>
>>
>
> I am very sure, that I am one of the 5%. So be very careful with the
>
> following code. From the description it looks like the scipy function
>
> is doing something like this:
>
>
>
> function pinv2,a,rcond=rcond
>
> ;
>
> compile_opt defint32,strictarr,logical_predicate
>
> ;
>
> svdc,a,w,u,v   ; singular value decomposition
>
> ;
>

```
>  n=n_elements(w)
>
>  threshold=n_elements(rcond)? max(w)*rcond : 0.
>
>  ii=where(w gt threshold,count)
>
>  if count lt n then begin
>
>    message,/info,strtrim(n-count,2)+' small singular values.'
>
>    if count le 0 then message,'All singular values are too small.'
>
>    end
>
>  ;
>
>  jj=(indgen(n))[ii]*(n+1)   ; diagonal elements
>
>  matrix=make_array(n,n,type=size(w,/type))
>
>  matrix[jj]=1./w[ii]
>
>  result=transpose(u)#matrix#v
>
>  ;
>
>  return,result
>
>  end
>
>
>
```

> If you want to use double precision, take a look at the IDL function
>
> LA_SVD.
>
>
>
> Cheers, Heinz