Subject: Re: Array intersections
Posted by J.D. Smith on Tue, 10 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

David Foster wrote:
>
> J.D. Smith wrote:
>>
>> Andy Loughe wrote:
>>>
>>>> What is the most efficient way (using IDL, of course) to return
>>>> the index at which two arrays intersect? e.g.
>>>> <snip>
>>>
>>> I believe the response of David Fanning does not return the "index"
>>> at which two arrays intersect, but the actual values themselves
>>> (right?).
>>> Here is one solution for what you have asked for...
>>
>> I made these comments about this method last year:
>>
>>> Check out the NASA library routine match(), which is array based. It uses a
>>> flag array and an index array, so the memory overhead is roughly 3 times the
>>> sum of the two arrays, but it's pretty fast. It's attached. Note that it takes
>>> vectors, so you've go to flatten your array upon input (with reform).
>>>
>>
>>> Just make sure you don't try and use [where_array] with big arrays -- it's an n^2 >algorithm (versus the order n algorithms posted prior). E.g., to compare two >floating 128x128 arrays for overlapping values, the program would create 3 arrays, >each of which takes 1 GB! The routine match() is likely much more efficient for >doing intersections on big arrays. (Unless you have some serious RAM on your >machine).
>>
>> JD
>
> Some time ago someone from RSI posted these routines for doing
> array computations. I have found them to be very fast, and memory
> efficient as well. If you need a routine to return the VALUES of
> the intersection, you can download FIND_ELEMENTS.PRO at:
>
> ftp://bial8.ucsd.edu pub/software/idl/share/idl_share.tar.gz
>
> This routine is quite fast! It returns the values, not the indices.
>
> Enjoy!
>
> Here are the routines posted by RSI:
>

> ---------- SNIP ----------

It seems that set_intersection does indeed return the *values*, as the information says... If you need the *indices*, use could use something like find_elements, but this is just the same n^2 algorithm that I was warning against. So, I repeat, if you want to find the indices, and you have large data sets, you will be better off with a slower, but order n algorithm.

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-4083
206 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|