Subject: Re: Vector comparison.
Posted by hunter on Thu, 20 Nov 2003 14:49:15 GMT
View Forum Message <> Reply to Message

Hi,

Thanks, Chris. I came to a similar routine using histogram and a loop. It may be more practical than the "memory hogging" answer, in my world anyway :). Although, I'll keep both on hand. If I think of anything better, I'll be sure to post it.

Thanks to everone forf their help.
Eli

"Christopher Lee" <cl@127.0.0.1> wrote in message
news:20031120.100922.1987231011.27332@buckley.atm.ox.ac.uk.. .
> In article <3fbbc86c$1@rutgers.edu>, "hunter" <elhunter@rci.rutgers.edu>
> wrote:
>
>
>> Hello,
>> These seems to be a fairly simple problem but I'm having difficulty
>> coming up with an elegant solution.
>> Let's say I have two vectors of type integer: A=[0,1,3,3,3,6,7,9,9]
>> B=[3,7]
>> I would like to design a function which returns the indices of all the
>> elements of A which appear in B.
>> i.e.
>> C=get_match(A,B)
>> should return
>> C=[2,3,4,6]
>> The simplest answer (I believe) is to loop through B and use the where
>> command. I just wonder if there is a way to do this without useing the
>> loop, as (in reality) the length of B may be very large. I suppose
>> another possibility is to use the histogram command with reverse_indices
>> set. But I think this would still require me to use a loop. Although it
>> may be faster since I would only have to call histogram once. Any
>> thoughts?
>> Thanks,
>> Eli
>>
>
> How much memory do you have.... (or, to put it another way, loops really
> aren't that bad when n_elements(array) > big_number :)
>
> anyway, the memory hogging non loopy answer.

```
>
>  a=[0,1,3,3,3,6,7,9,9]
>  b=[3,7]
>
>  ...........
>
>  function get_match,a,b
>
>  ;needs some up-to-date version of IDL, 5.4 I think.
>
>  na=n_elements(a)
>  nb=n_elements(b)
>
>  ;some checks go here to make sure the world won't explode, an exersice
>  ;for the reader.
>
>  reb_a=[na,nb]
>  ref_a=[na,1]
>  reb_b=[na,nb]
>  ref_b=[1,nb]
>
>  a_temp=rebin(reform(a, ref_a),reb_a)
>  b_temp=rebin(reform(b, ref_b),reb_b)
>
>  c_temp=a_temp-b_temp
>  a_temp=0
>  b_temp=0
>
>  w=where(c_temp eq 0, count)
>
>  c_temp=0
>  ;memory...
>
>  if( count gt 0) then c=w mod na
>  if( count eq 0) then c=-1
>
>  return, c
>  end
>
>  -----
>
>  if they are guaranteed to be integer (non-integer histogramming implies a
>  fuzzy match doesn't it?) and B is very big :-
>
>  hist=histogram(a,reverse_indices=r)
>
>  n=n_elements(b)
>
```

> if(n eq 0) then c=-1
>
> if(n ge 1) then c=r[r[b[0]-min(a)]:r[b[0]+1-min(a)]-1]
>
> ;does this work if b is not an array, something changed between 5.3 and
> ;5.5 but I forget.
>
> if(n gt 1) then for i=1, n-1 do begin
> c=[c,r[r[b[i]-min(a)]:r[b[i]+1-min(a)]-1]]
> endfor
>
> ;this last bit should work..... This still has the loop of course but
> It does grab all of the indices (unlike a UNIQ approach for example) and
> uses less memory (O(na) instead of O(na*nb))
>
> ..
>
> I think there must be a better way (a more IDL way), but inspiration
> hasn't hit yet.
>
> Chris.
