Subject: Re: Distance between two sets of datapoints
Posted by Maxwell Peck on Thu, 25 Mar 2010 10:46:31 GMT
View Forum Message <> Reply to Message

That should say I am generating the full set of indices and then
subsetting.
Thanks
On Mar 25, 9:33 pm, Maxwell Peck <maxjp...@gmail.com> wrote:
> Hi All,
>
> I have two sets of data points, inputa and inputb, inputa has many
> more data points than the other. I need to find the distance between
> each point in inputa and all the points in inputb (hopefully without
> loops). At the moment I am using parts of distance_measure.pro to
> generate the full set of distances and subsetting indexes as required.
> The code follows. I can't see an easy way of generating the indexes
> though so that only the pairs i want are calculated (i.e. not
> calculating between inputb/inputb or inputa/inputa indexes).
>
> inputa and inputb can be very very large so I don't think i can use a
> matrix approach to do it vectorially and looping seems awfully slow.
>
> An alternative approach or other suggestions would be appreciated.
>
> Regards
>
> Max
>
> inputa = findgen(2,10)
> inputb = findgen(2,2)
>
> t = [[inputa],[inputb]]
> m=n_elements(t)/2
> n = m*(m-1)/2
>
> ii = 0L
> index0 = LINDGEN(m - 1) + 1   ; work array
> index1 = LONARR(n, /NOZERO)
> index2 = LONARR(n, /NOZERO)
>
> for i=0,m-2 do begin
>   n1 = m - (i+1)
>   index1[ii:ii+n1-1] = i
>   index2[ii] = index0[0:n1-1] + i
>   ii += n1
> endfor
>
> diff = abs(t[*,index1] - t[*,index2])

> res = sqrt(TOTAL(diff^2, 1))

Subject: Re: Distance between two sets of datapoints
Posted by Kenneth P. Bowman on Thu, 25 Mar 2010 14:55:47 GMT
View Forum Message <> Reply to Message

In article
<4707b3ed-9e43-4863-977f-7fd091e62868@a4g2000prb.googlegroups.com>,
Maxwell Peck <maxjpeck@gmail.com> wrote:

> Hi All,
>
> I have two sets of data points, inputa and inputb, inputa has many
> more data points than the other. I need to find the distance between
> each point in inputa and all the points in inputb (hopefully without
> loops). At the moment I am using parts of distance_measure.pro to
> generate the full set of distances and subsetting indexes as required.
> The code follows. I can't see an easy way of generating the indexes
> though so that only the pairs i want are calculated (i.e. not
> calculating between inputb/inputb or inputa/inputa indexes).
>
> inputa and inputb can be very very large so I don't think i can use a
> matrix approach to do it vectorially and looping seems awfully slow.
>
> An alternative approach or other suggestions would be appreciated.
>
> Regards
>
> Max
>
>
> inputa = findgen(2,10)
> inputb = findgen(2,2)
>
> t = [[inputa],[inputb]]
> m=n_elements(t)/2
> n = m*(m-1)/2
>
> ii = 0L
> index0 = LINDGEN(m - 1) + 1   ; work array
> index1 = LONARR(n, /NOZERO)
> index2 = LONARR(n, /NOZERO)
>
> for i=0,m-2 do begin
> n1 = m - (i+1)
> index1[ii:ii+n1-1] = i
> index2[ii] = index0[0:n1-1] + i

>   ii += n1
> endfor
>
> diff = abs(t[*,index1] - t[*,index2])
> res = sqrt(TOTAL(diff^2, 1))

If I understand you right, don't you want to do this?

```
na = 10
nb = 2
xa = FINDGEN(na)
ya = FINDGEN(na)
xb = FINDGEN(nb)
yb = FINDGEN(nb)

dist = FLTARR(na, nb)

FOR i = 0, nb-1 DO BEGIN
   d = SQRT((xb[i] - xa)^2 + (yb[i] - ya)^2)
   dist[0,i] = d
ENDFOR
```

I changed the notation to make the indexing simpler.

This computes the Cartesian distance between all pairs of
points in a and b.

You should loop over the smaller of na or nb.

Ken Bowman

Subject: Re: Distance between two sets of datapoints
Posted by Kenneth P. Bowman on Thu, 25 Mar 2010 19:04:20 GMT
View Forum Message <> Reply to Message

In article <k-bowman-EEB2FB.09554725032010@news.tamu.edu>,
 "Kenneth P. Bowman" <k-bowman@null.edu> wrote:

>
> FOR i = 0, nb-1 DO BEGIN
>   d = SQRT((xb[i] - xa)^2 + (yb[i] - ya)^2)
>   dist[0,i] = d
> ENDFOR

I just realized this could be simplified to

```
FOR i = 0, nb-1 DO dist[0,i] = SQRT((xb[i] - xa)^2 + (yb[i] - ya)^2)
```

which removes some unneeded memory access.

Ken

Subject: Re: Distance between two sets of datapoints
Posted by Maxwell Peck on Thu, 25 Mar 2010 19:38:49 GMT
View Forum Message <> Reply to Message

Thanks Ken, I'll give it a shot. I had tried a similar loop (I
thought) but it seemed to be very slow that is why i was looking at
vectorising it similar to distance_measure.

Thanks
Max

On Mar 26, 6:04 am, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
> In article <k-bowman-EEB2FB.09554725032...@news.tamu.edu>,
> "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
>
>
>
>> FOR i = 0, nb-1 DO BEGIN
>> d = SQRT((xb[i] - xa)^2 + (yb[i] - ya)^2)
>> dist[0,i] = d
>> ENDFOR
>
> I just realized this could be simplified to
>
> FOR i = 0, nb-1 DO dist[0,i] = SQRT((xb[i] - xa)^2 + (yb[i] - ya)^2)
>
> which removes some unneeded memory access.
>
> Ken

Subject: Re: Distance between two sets of datapoints
Posted by Kenneth P. Bowman on Fri, 26 Mar 2010 13:11:05 GMT
View Forum Message <> Reply to Message

In article
<eb0d2a64-900b-4c9e-9599-699dcd3da17b@k4g2000prh.googlegroups.com>,
Maxwell Peck <maxjpeck@gmail.com> wrote:

> Thanks Ken, I'll give it a shot. I had tried a similar loop (I
> thought) but it seemed to be very slow that is why i was looking at

> vectorising it similar to distance_measure.
>>
>> FOR i = 0, nb-1 DO dist[0,i] = SQRT((xb[i] - xa)^2 + (yb[i] - ya)^2)

This should vectorize reasonably well if na is large and nb is
not too large.

Ken

Subject: Re: Distance between two sets of datapoints
Posted by Gray on Fri, 26 Mar 2010 13:16:53 GMT
View Forum Message <> Reply to Message

On Mar 25, 3:38 pm, Maxwell Peck <maxjp...@gmail.com> wrote:
> Thanks Ken, I'll give it a shot. I had tried a similar loop (I
> thought) but it seemed to be very slow that is why i was looking at
> vectorising it similar to distance_measure.
>
> Thanks
> Max
>
> On Mar 26, 6:04 am, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
>
>
>
>> In article <k-bowman-EEB2FB.09554725032...@news.tamu.edu>,
>> "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
>
>>> FOR i = 0, nb-1 DO BEGIN
>>> d = SQRT((xb[i] - xa)^2 + (yb[i] - ya)^2)
>>> dist[0,i] = d
>>> ENDFOR
>
>> I just realized this could be simplified to
>
>> FOR i = 0, nb-1 DO dist[0,i] = SQRT((xb[i] - xa)^2 + (yb[i] - ya)^2)
>
>> which removes some unneeded memory access.
>
>> Ken

Take a look at JD Smith's match_2d (http://tir.astro.utoledo.edu/idl/
match_2d.pro), which uses histogram to quickly find distances between
lists of coordinates (and then matches them, which you don't need).

Subject: Re: Distance between two sets of datapoints
Posted by Maxwell Peck on Fri, 26 Mar 2010 22:11:38 GMT
View Forum Message <> Reply to Message

Thanks Ken,

It is definitely much much faster (from 30 minutes for looping only to 1 minute). It is difficult to characterise the speedup entirely because I was able to add a lot more of the calculations inside the loop, na is about 700 times larger than nb. I think I must have done something stupid the first time I tried this and looped over the other dimension hence my confusion.

Cheers,
Max

On Mar 27, 12:11 am, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
> In article
> < eb0d2a64-900b-4c9e-9599-699dcd3da...@k4g2000prh.googlegroups .com >,
> Maxwell Peck <maxjp...@gmail.com> wrote:
>
>> Thanks Ken, I'll give it a shot. I had tried a similar loop (I
>> thought) but it seemed to be very slow that is why i was looking at
>> vectorising it similar to distance_measure.
>
>>> FOR i = 0, nb-1 DO dist[0,i] = SQRT((xb[i] - xa)^2 + (yb[i] - ya)^2)
>
> This should vectorize reasonably well if na is large and nb is
> not too large.
>
> Ken