Subject: Re: Assign data point to n-Dimensional grid
Posted by Kenneth P. Bowman on Fri, 22 Jun 2012 15:14:01 GMT
View Forum Message <> Reply to Message

In article <ada6c25c-b056-4578-8d10-f9fb2fa9694d@googlegroups.com>,
antar3s86@gmail.com wrote:

> Hi
>
> I face a serious problem in the development of my algorithm.In principle it
> is very simple:
>
> I have a data point in a 9-dimensional parameter space (say,
> x1,x2,x3,y1,y2,y3,z1,z2,z3) with x,y,z being physical quantities with
> different units. Furthermore, I have an unequally spaced 9-dimensional
> reference grid and all i have to do is to compute which grid point is closest
> to my data point with respect to all 9 dimensions.
>
> I have to do this several billion times, so I really want to make sure to do
> it as fast as possible.
>
> Any help with that?
>
> cheers

Is your grid separable? That is, does the x-coordinate of
each grid point depend only on x? If it does, you can find
the index of each nearest neighbor independently of the others.

If your grids are regular, you should be able to compute the
nearest neighbor index. Something like this

```
   i = ROUND(nx*(x - xmin)/(xmax - xmin))
```

If your grids are not regular (not evenly spaced), use
VALUE_LOCATE to do a binary search.

If your grids are not separable, you have a much more difficult
problem.

Ken Bowman

Subject: Re: Assign data point to n-Dimensional grid
Posted by antar3s86 on Fri, 22 Jun 2012 15:38:57 GMT
View Forum Message <> Reply to Message

On Friday, June 22, 2012 5:14:01 PM UTC+2, Kenneth P. Bowman wrote:

>
>> Hi
>>
>> I face a serious problem in the development of my algorithm.In principle it
>> is very simple:
>>
>> I have a data point in a 9-dimensional parameter space (say,
>> x1,x2,x3,y1,y2,y3,z1,z2,z3) with x,y,z being physical quantities with
>> different units. Furthermore, I have an unequally spaced 9-dimensional
>> reference grid and all i have to do is to compute which grid point is closest
>> to my data point with respect to all 9 dimensions.
>>
>> I have to do this several billion times, so I really want to make sure to do
>> it as fast as possible.
>>
>> Any help with that?
>>
>> cheers
>
> Is your grid separable? That is, does the x-coordinate of
> each grid point depend only on x? If it does, you can find
> the index of each nearest neighbor independently of the others.
>
> If your grids are regular, you should be able to compute the
> nearest neighbor index. Something like this
>
>   i = ROUND(nx*(x - xmin)/(xmax - xmin))
>
> If your grids are not regular (not evenly spaced), use
> VALUE_LOCATE to do a binary search.
>
> If your grids are not separable, you have a much more difficult
> problem.
>
> Ken Bowman

Hi

Thanks a lot! My grid is unequally spaced but separable and I think VALUE_LOCATE was just the thing i have been looking for.

You saved my day (or better: week)

cheers

Subject: Re: Assign data point to n-Dimensional grid

Posted by antar3s86 on Fri, 22 Jun 2012 16:06:15 GMT
View Forum Message <> Reply to Message

Now I find that it is not exactly what I'm looking for

Suppose my grid is [5,1,12] and I want to find to which of these values a data point of 4 is closest to.

So I write

```
grid = [5,1,12]
print, VALUE_LOCATE(grid,4)
        1
```

But indeed it should be 0 since the 5 in the grid is closer to my data point...
So in fact I need the nearest neighbor... :(

Subject: Re: Assign data point to n-Dimensional grid

Posted by Craig Markwardt on Fri, 22 Jun 2012 17:49:21 GMT
View Forum Message <> Reply to Message

On Friday, June 22, 2012 12:06:15 PM UTC-4, (unknown) wrote:
> Now I find that it is not exactly what I'm looking for
>
> Suppose my grid is [5,1,12] and I want to find to which of these values a data point of 4 is closest to.
>
> So I write
>
> grid = [5,1,12]
> print, VALUE_LOCATE(grid,4)
>   1
>
> But indeed it should be 0 since the 5 in the grid is closer to my data point...
> So in fact I need the nearest neighbor... :(

By the way, your grid has to be strictly ascending. If you pass a randomly ordered grid, expect random results.

VALUE_LOCATE() always finds the next lowest grid point, not the nearest gridpoint.

On the other hand, it's easy enough to check for this.

```
  x = your data points
  grid = [1, 5, 12]
  ii = value_locate(grid, x)  ;; You already know this much
  ;; See if the ii+1 grid point is closer
  ;;      _no overflow_     ___ ii+1 sep ___    __ ii sep __
  wh = where(ii LT 2    AND (grid[ii+1] - x) LT (x-grid[ii]), ct)

  ;; If we found some, then use those instead
  if ct GT 0 then ii[wh] = ii[wh]+1
```

Craig

Subject: Re: Assign data point to n-Dimensional grid
Posted by Kenneth P. Bowman on Fri, 22 Jun 2012 18:28:09 GMT

View Forum Message <> Reply to Message

In article <60a53808-3670-43e1-85da-ee00537487f9@googlegroups.com>,
 Craig Markwardt <craig.markwardt@gmail.com> wrote:

```
>  On Friday, June 22, 2012 12:06:15 PM UTC-4, (unknown) wrote:
>>  Now I find that it is not exactly what I'm looking for
>>
>>  Suppose my grid is [5,1,12] and I want to find to which of these values a
>>  data point of 4 is closest to.
>>
>>  So I write
>>
>>  grid = [5,1,12]
>>  print, VALUE_LOCATE(grid,4)
>>            1
>>
>>  But indeed it should be 0 since the 5 in the grid is closer to my data
>>  point...
>>  So in fact I need the nearest neighbor... :(
>
>  By the way, your grid has to be strictly ascending.  If you pass a randomly
>  ordered grid, expect random results.
>
>  VALUE_LOCATE() always finds the next lowest grid point, not the nearest
>  gridpoint.
>
>  On the other hand, it's easy enough to check for this.
>
>    x = your data points
>    grid = [1, 5, 12]
>    ii = value_locate(grid, x)  ;; You already know this much
>
>    ;; See if the ii+1 grid point is closer
>    ;;      _no overflow_     ___ ii+1 sep ___    __ ii sep __
>    wh = where(ii LT 2    AND (grid[ii+1] - x) LT (x-grid[ii]), ct)
```

>
>   ;; If we found some, then use those instead
>   if ct GT 0 then ii[wh] = ii[wh]+1
>
> Craig

What he said!

Ken