Subject: Re: Nearest neighbors
Posted by Chris[6] on Fri, 09 Oct 2009 01:55:06 GMT
View Forum Message <> Reply to Message

On Oct 8, 2:08 pm, "N. Johnson" <evilish...@gmail.com> wrote:
> On Oct 7, 7:00 pm, Chris <beaum...@ifa.hawaii.edu> wrote:
>
>
>
>> On Oct 7, 2:01 pm, "N. Johnson" <evilish...@gmail.com> wrote:
>
>>> I have a set of latitude/longitude pairs and I need to find the n
>>> closest neighbors for all of them. I'm trying to use the
>>> nearest_neighbors() function found on this
page:http://www.dfanning.com/code_tips/slowloops.html
>
>>> However, when I attempt to run the function, I get an error on the
>>> line:
>>> p=c[c[point]:c[point+1]-1] ;start with this point's DT neighbors
>
>>> because c[point] is equal to c[point+1]. Since I don't know exactly
>>> what the function is doing, I don't know how to fix it. If it matters,
>>> I have a lot of lat/lon pairs (~1e6) and there may be duplicates.
>
>>> Any suggestions?
>>> Nathan Johnson
>
>> I have an alternative nearest neighbors routine that doesn't use
>> triangulation - it may be useful (it assumes a euclidian space, so it
>> won't work if your points are very spread out or near a pole)
>
>> Documentation:http://www.ifa.hawaii.edu/~beaumont/code/neare stn.html(lookat
>> nearestn, not nearestn_findneighbors)
>
>> Library:http://www.ifa.hawaii.edu/~beaumont/code/beaumont_li brary.tar
>
>> Chris
>
> Chris,
>
> Thanks that works well. Is there a way to get the nth nearest points
> by calling that function just once? Or do I have to call it n times?
>
> Thanks,
> Nathan

Yes, if you set the /all keyword, it will return a 2d array of the nth
nearest neighbors for each point. Note that it starts counting from

zero, so if you set n = 3, the resulting array will by (4, n_points)

chris