Subject: k-mean clustering idl
Posted by smnadoum on Sun, 12 Feb 2017 23:13:55 GMT
View Forum Message <> Reply to Message

I was wondering if I could get some help on clustering in IDL. I found a good example on Harris Geospatial that explains the method, however, I am confused on how to run the clustering on my own data (ASCII) to perform the K-mean analysis. How can I use my data instead of the 'random' function that generates random numbers
Below is the code I found on Harris:

```
n = 50
c1 = RANDOMN(seed, 3, n)
c1[0:1,*] -= 3
c2 = RANDOMN(seed, 3, n)
c2[0,*] += 3
c2[1,*] -= 3
c3 = RANDOMN(seed, 3, n)
c3[1:2,*] += 3
array = [[c1], [c2], [c3]]
; Compute cluster weights, using three clusters:
weights = CLUST_WTS(array, N_CLUSTERS = 3)
; Compute the classification of each sample:
result = CLUSTER(array, weights, N_CLUSTERS = 3)
```

My data is in ASCII format and I have already wrote a code that opens and read ascii (below) but not sure how to run the k-mean clustering analysis on my data. I can't find good IDL resources that explains the clustering.

```
pro read_text,  file, dir

dir='path'
file= 'path*file'

n = file_lines(file)
gv= fltarr(n)
npv= fltarr(n)
soil= fltarr(n)
gv0= 0.0
npv0= 0.0
soil0=0.0


openr, iunit, file, /get_lun
for i= 0, n-1 do begin

readf, iunit, gv0, npv0, soil0

gv[i]= gv0
```

```
npv[i]= npv0
soil[i]=soil0

endfor


free_lun, iunit

for i = 0, n-1 do print, gv[i], npv[i], soil[i]

mwell=gv[0,*,*]     ;this doesn't work
mwell=gv[1,*,*]
mwell=gv[2,*,*]

end
```

Thank you.

Subject: Re: k-mean clustering idl
Posted by Dick Jackson on Thu, 16 Feb 2017 06:16:37 GMT

View Forum Message <> Reply to Message

Hi Cheryl,

I answered another question by you the other day, and now it seems (on Google Groups anyway) that the question was deleted, leaving only my answer. Curious.

Find it here, and it can help you to get your data into the "mwell" array that you seem to want to create (use your gv, npv and soil variable names you have below, if those are better).
https://groups.google.com/forum/#!topic/comp.lang.idl-pvwave /Gsp1JNUZSxs

Once you have "mwell" as a (3, n) array, you simply run:

```
nClusters = 3     ; or as many as you wish
centroidsXYZ = CLUST_WTS(mwell, N_CLUSTERS = nClusters)
```

Now centroidsXYZ is a (3, nClusters) array where each row describes a point in the space of (gv, npv, soil) values, which is the "centroid" of the cluster. To determine which cluster the "n" original points belong to (and you can do this with any points you're interested in):

```
clusterIDs = CLUSTER(mwell, centroidsXYZ, N_CLUSTERS = nClusters)
```

"clusterIDs" will be a (1, n) array (since mwell is (3, n)), with values from 0 to nClusters-1, telling which of the centroids each of the "n" original points was nearest to.

Be aware that if the range of values seen in your three variables (gv, npv and soil) are very different, k-means clustering may not give the results you expect. If you normalize the variables to have similar ranges, work in the normalized space, and do the reverse "un-normalizing" process when you need real-world numbers for them again.

I hope this helps!

Cheers,
-Dick

Dick Jackson Software Consulting Inc.
Victoria, BC, Canada --- http://www.d-jackson.com

On Sunday, 12 February 2017 15:13:57 UTC-8, Cheryl wrote:

> I was wondering if I could get some help on clustering in IDL. I found a good example on Harris Geospatial that explains the method, however, I am confused on how to run the clustering on my own data (ASCII) to perform the K-mean analysis. How can I use my data instead of the 'random' function that generates random numbers
>
> Below is the code I found on Harris:

```
>
>  n = 50
>  c1 = RANDOMN(seed, 3, n)
>  c1[0:1,*] -= 3
>  c2 = RANDOMN(seed, 3, n)
>  c2[0,*] += 3
>  c2[1,*] -= 3
>  c3 = RANDOMN(seed, 3, n)
>  c3[1:2,*] += 3
>  array = [[c1], [c2], [c3]]
>  ; Compute cluster weights, using three clusters:
>  weights = CLUST_WTS(array, N_CLUSTERS = 3)
>  ; Compute the classification of each sample:
>  result = CLUSTER(array, weights, N_CLUSTERS = 3)
>
```

> My data is in ASCII format and I have already wrote a code that opens and read ascii (below) but not sure how to run the k-mean clustering analysis on my data. I can't find good IDL resources that explains the clustering.

```
>
>  pro read_text,  file, dir
>
>  dir='path'
>  file= 'path*file'
>
>  n = file_lines(file)
>  gv= fltarr(n)
>  npv= fltarr(n)
>  soil= fltarr(n)
>  gv0= 0.0
>  npv0= 0.0
>  soil0=0.0
>
>
>  openr, iunit, file, /get_lun
>  for i= 0, n-1 do begin
>
>  readf, iunit, gv0, npv0, soil0
>
>  gv[i]= gv0
>  npv[i]= npv0
>  soil[i]=soil0
>
>  endfor
>
>
>  free_lun, iunit
>
>  for i = 0, n-1 do print, gv[i], npv[i], soil[i]
>
>  mwell=gv[0,*,*]     ;this doesn't work
>  mwell=gv[1,*,*]
>  mwell=gv[2,*,*]
>
>  end
>
>
>
>  Thank you.
```

Subject: Re: k-mean clustering idl
Posted by smnadoum on Fri, 17 Feb 2017 22:39:15 GMT

View Forum Message <> Reply to Message

On Wednesday, February 15, 2017 at 10:16:41 PM UTC-8, Dick Jackson wrote:

> Hi Cheryl,
>
> I answered another question by you the other day, and now it seems (on Google Groups anyway) that the question was deleted, leaving only my answer. Curious.
>
> Find it here, and it can help you to get your data into the "mwell" array that you seem to want to create (use your gv, npv and soil variable names you have below, if those are better).
> https://groups.google.com/forum/#!topic/comp.lang.idl-pvwave /Gsp1JNUZSxs
>
> Once you have "mwell" as a (3, n) array, you simply run:
>

```
>  nClusters = 3     ; or as many as you wish
>  centroidsXYZ = CLUST_WTS(mwell, N_CLUSTERS = nClusters)
>
```

> Now centroidsXYZ is a (3, nClusters) array where each row describes a point in the space of (gv, npv, soil) values, which is the "centroid" of the cluster. To determine which cluster the "n" original points belong to (and you can do this with any points you're interested in):

```
>
>  clusterIDs = CLUSTER(mwell, centroidsXYZ, N_CLUSTERS = nClusters)
>
```

> "clusterIDs" will be a (1, n) array (since mwell is (3, n)), with values from 0 to nClusters-1, telling which of the centroids each of the "n" original points was nearest to.
>
> Be aware that if the range of values seen in your three variables (gv, npv and soil) are very different, k-means clustering may not give the results you expect. If you normalize the variables to have similar ranges, work in the normalized space, and do the reverse "un-normalizing" process when you need real-world numbers for them again.
>
> I hope this helps!
>
> Cheers,
> -Dick
>
> Dick Jackson Software Consulting Inc.
> Victoria, BC, Canada --- http://www.d-jackson.com
>
>
> On Sunday, 12 February 2017 15:13:57 UTC-8, Cheryl wrote:
>> I was wondering if I could get some help on clustering in IDL. I found a good example on Harris Geospatial that explains the method, however, I am confused on how to run the clustering on my own data (ASCII) to perform the K-mean analysis. How can I use my data instead of the 'random' function that generates random numbers
>> Below is the code I found on Harris:

```
>>
>>  n = 50
>>  c1 = RANDOMN(seed, 3, n)
>>  c1[0:1,*] -= 3
>>  c2 = RANDOMN(seed, 3, n)
>>  c2[0,*] += 3
>>  c2[1,*] -= 3
>>  c3 = RANDOMN(seed, 3, n)
>>  c3[1:2,*] += 3
>>  array = [[c1], [c2], [c3]]
>>  ; Compute cluster weights, using three clusters:
>>  weights = CLUST_WTS(array, N_CLUSTERS = 3)
>>  ; Compute the classification of each sample:
>>  result = CLUSTER(array, weights, N_CLUSTERS = 3)
>>
```

>> My data is in ASCII format and I have already wrote a code that opens and read ascii (below)

but not sure how to run the k-mean clustering analysis on my data. I can't find good IDL resources that explains the clustering.

```
>>
>>  pro read_text,  file, dir
>>
>>  dir='path'
>>  file= 'path*file'
>>
>>  n = file_lines(file)
>>  gv= fltarr(n)
>>  npv= fltarr(n)
>>  soil= fltarr(n)
>>  gv0= 0.0
>>  npv0= 0.0
>>  soil0=0.0
>>
>>
>>  openr, iunit, file, /get_lun
>>  for i= 0, n-1 do begin
>>
>>  readf, iunit, gv0, npv0, soil0
>>
>>  gv[i]= gv0
>>  npv[i]= npv0
>>  soil[i]=soil0
>>
>>  endfor
>>
>>
>>  free_lun, iunit
>>
>>  for i = 0, n-1 do print, gv[i], npv[i], soil[i]
>>
>>  mwell=gv[0,*,*]     ;this doesn't work
>>  mwell=gv[1,*,*]
>>  mwell=gv[2,*,*]
>>
>>  end
>>
>>
>>
>>  Thank you.
```

Thank you so much.

I think there is a technical problem with my previous post, cause I can only see what I posted and can't see your answer anymore. But I really appreciate your help. I will carefully read your explanation about clustering and try to work on it.

Thanks again