Subject: clustering
Posted by nivedita.raghunath on Tue, 17 Jul 2007 15:36:53 GMT
View Forum Message <> Reply to Message

Hi all,

I am trying to cluster a (7 x n) array with n_clusters=5. Visually I
can see 5 distinct clusters, but when I do clust_wts the cluster
centroids don't end up right. No matter what options I give, clust_wts
refuses to find the clusters.

Any idea on whats going on ?

Thanks,
Nivedita

Subject: Re: clustering
Posted by Conor on Wed, 18 Jul 2007 19:07:53 GMT
View Forum Message <> Reply to Message

On Jul 17, 11:36 am, nivedita.raghun...@gmail.com wrote:
> Hi all,
>
> I am trying to cluster a (7 x n) array with n_clusters=5. Visually I
> can see 5 distinct clusters, but when I do clust_wts the cluster
> centroids don't end up right. No matter what options I give, clust_wts
> refuses to find the clusters.
>
> Any idea on whats going on ?
>
> Thanks,
> Nivedita

No idea. More information might be helpful. It's quite possible
though that the the clust_wts algorithm just doesn't work for your
particular data set, at least not as well as you apparently want it to.

Subject: Re: clustering
Posted by nivedita.raghunath on Thu, 19 Jul 2007 16:12:44 GMT
View Forum Message <> Reply to Message

Here is a subset of my data.

IDL> help,pos1
POS1 FLOAT = Array[7, 53]

IDL> print,pos1
    0.910300   0.413400   -0.0221000   0.00300000   -150.250
129.510   -13.0400
    0.910200   0.413400   -0.0223000   0.00370000   -150.280
129.460   -13.0200
    0.910200   0.413500   -0.0228000   0.00360000   -150.300
129.400   -13.1300
    0.910200   0.413400   -0.0231000   0.00310000   -150.190
129.520   -13.0700
    0.910200   0.413400   -0.0226000   0.00320000   -150.220
129.580   -13.0800
    0.910200   0.413600   -0.0224000   0.00460000   -150.510
129.040   -13.1000
    0.910200   0.413500   -0.0221000   0.00250000   -150.210
129.560   -13.0000
    0.910200   0.413500   -0.0223000   0.00340000   -150.310
129.420   -13.1000
    0.910200   0.413500   -0.0225000   0.00350000   -150.160
129.620   -13.0900
    0.910200   0.413500   -0.0224000   0.00240000   -150.090
129.720   -13.0100
    0.930600   0.365500   -0.0216000   0.00170000   -147.800
125.760   -16.7500
    0.930500   0.365600   -0.0220000 0.000900000   -147.650
125.160   -16.6800
    0.930500   0.365700   -0.0222000   0.00230000   -147.930
125.370   -16.8100
    0.930500   0.365700   -0.0217000   0.00280000   -148.090
125.750   -16.8600
    0.930400   0.365800   -0.0225000   0.00240000   -147.800
125.400   -16.8200
    0.930400   0.365800   -0.0213000   0.00430000   -148.490
124.950   -16.7800
    0.930400   0.365800   -0.0220000   0.00210000   -147.910
126.000   -16.7200
    0.930400   0.365800   -0.0220000   0.00200000   -147.830
125.560   -16.6900
    0.930400   0.365900   -0.0216000   0.00250000   -148.080
125.490   -16.7700
    0.930400   0.365800   -0.0224000   0.00230000   -147.870
125.980   -16.6200
    0.897600   0.439600   -0.0331000   0.00790000   -147.060
130.970   -6.02000
    0.897600   0.439500   -0.0334000   0.00720000   -146.790
130.520   -6.13000
    0.897500   0.439600   -0.0337000   0.00770000   -146.820
130.660   -6.13000

0.897500 0.439600 -0.0328000 0.00750000 -147.160
130.790 -6.13000
0.897600 0.439600 -0.0331000 0.00680000 -146.860
130.570 -6.07000
0.897600 0.439600 -0.0335000 0.00700000 -146.830
130.660 -6.12000
0.897600 0.439500 -0.0326000 0.00750000 -147.090
130.870 -6.08000
0.897600 0.439600 -0.0327000 0.00750000 -146.880
130.610 -6.14000
0.897600 0.439500 -0.0336000 0.00810000 -146.980
130.560 -6.25000
0.897600 0.439500 -0.0331000 0.00800000 -147.130
130.820 -6.19000
0.897500 0.439600 -0.0332000 0.00800000 -147.000
130.600 -6.25000
0.871700 0.488800 -0.0332000 0.0102000 -146.260
133.480 -1.14000
0.871600 0.488900 -0.0330000 0.0111000 -146.390
133.540 -1.29000
0.871600 0.488900 -0.0347000 0.00920000 -145.690
132.630 -1.26000
0.871700 0.488800 -0.0337000 0.0103000 -146.100
133.330 -1.44000
0.871700 0.488700 -0.0336000 0.0104000 -146.310
133.610 -1.58000
0.871700 0.488800 -0.0340000 0.00950000 -145.820
132.840 -1.33000
0.872000 0.488200 -0.0335000 0.00960000 -146.040
133.140 -1.95000
0.872000 0.488200 -0.0330000 0.00820000 -145.910
133.210 -1.83000
0.872000 0.488300 -0.0333000 0.0100000 -146.040
133.110 -1.82000
0.872100 0.488200 -0.0330000 0.00880000 -146.000
133.150 -1.83000
0.872000 0.488200 -0.0335000 0.00900000 -145.910
133.210 -1.85000
0.873000 0.487300 -0.0227000 0.000700000 -143.720
132.260 -6.08000
0.872900 0.487300 -0.0230000 0.000100000 -143.630
132.350 -6.07000
0.872900 0.487300 -0.0235000 0.000500000 -143.560
132.370 -6.14000
0.872900 0.487300 -0.0234000 -0.000300000 -143.430
132.520 -6.15000
0.872900 0.487300 -0.0231000 0.000700000 -143.670
132.280 -6.15000

```
    0.872900     0.487300   -0.0237000  0.000200000     -143.480
132.430     -6.07000
    0.872900     0.487300   -0.0231000  0.000500000     -143.550
132.450     -6.03000
    0.872900     0.487300   -0.0241000 -0.000200000     -143.440
132.450     -6.11000
    0.872900     0.487300   -0.0237000  0.000400000     -143.470
132.490     -6.05000
    0.873000     0.487300   -0.0228000  0.000600000     -143.700
132.270     -6.03000
    0.872900     0.487300   -0.0235000 -0.000200000     -143.430
132.450     -6.10000

IDL> weights=clust_wts(pos1,n_clusters=5)
IDL> print,weights
    0.159265     0.119451     0.113155     0.180601    0.0680267
0.243488     0.116014
    0.874568     0.483835   -0.0256644   0.00231699     -144.219
132.388     -5.40240
    0.113501     0.127323    0.0985566     0.247231     0.225678
0.0779656     0.109745
    0.238006     0.236222     0.127174    0.0261984     0.266028
0.0180832    0.0882878
   0.0301962     0.232814     0.209770     0.146116     0.235975
0.134589    0.0105386

IDL> result=cluster(pos1,weights,n_clusters=5)
IDL> print,result(uniq(result))
         1
```

The 5 clusters are pretty distinct but "cluster" does a hopeless job identifying them. I tried scaling the data but that too finds 3 clusters in the end. Any ideas?

On Jul 18, 3:07 pm, Conor <cmanc...@gmail.com> wrote:
> On Jul 17, 11:36 am, nivedita.raghun...@gmail.com wrote:
>
>> Hi all,
>
>> I am trying to cluster a (7 x n) array with n_clusters=5. Visually I
>> can see 5 distinct clusters, but when I do clust_wts the cluster
>> centroids don't end up right. No matter what options I give, clust_wts
>> refuses to find the clusters.
>
>> Any idea on whats going on ?

>
>> Thanks,
>> Nivedita
>
> No idea. More information might be helpful. It's quite possible
> though that the the clust_wts algorithm just doesn't work for your
> particular data set, at least not as well as you apparently want it to.

Subject: Re: clustering
Posted by Vince Hradil on Thu, 19 Jul 2007 17:30:35 GMT
View Forum Message <> Reply to Message

On Jul 18, 2:07 pm, Conor <cmanc...@gmail.com> wrote:
> On Jul 17, 11:36 am, nivedita.raghun...@gmail.com wrote:
>
>> Hi all,
>
>> I am trying to cluster a (7 x n) array with n_clusters=5. Visually I
>> can see 5 distinct clusters, but when I do clust_wts the cluster
>> centroids don't end up right. No matter what options I give, clust_wts
>> refuses to find the clusters.
>
>> Any idea on whats going on ?
>
>> Thanks,
>> Nivedita
>
> No idea. More information might be helpful. It's quite possible
> though that the the clust_wts algorithm just doesn't work for your
> particular data set, at least not as well as you apparently want it to.

agreed... more information please. Some code snippets, some better
idea about the data, etc.

Subject: Re: clustering
Posted by Vince Hradil on Thu, 19 Jul 2007 20:46:00 GMT
View Forum Message <> Reply to Message

On Jul 19, 11:12 am, nivedita.raghun...@gmail.com wrote:
> Here is a subset of my data.
>
> IDL> help,pos1
> POS1 FLOAT = Array[7, 53]
>
> IDL> print,pos1

> 0.910300 0.413400 -0.0221000 0.00300000 -150.250
> 129.510 -13.0400
> 0.910200 0.413400 -0.0223000 0.00370000 -150.280
> 129.460 -13.0200
> 0.910200 0.413500 -0.0228000 0.00360000 -150.300
> 129.400 -13.1300
> 0.910200 0.413400 -0.0231000 0.00310000 -150.190
> 129.520 -13.0700
> 0.910200 0.413400 -0.0226000 0.00320000 -150.220
> 129.580 -13.0800
> 0.910200 0.413600 -0.0224000 0.00460000 -150.510
> 129.040 -13.1000
> 0.910200 0.413500 -0.0221000 0.00250000 -150.210
> 129.560 -13.0000
> 0.910200 0.413500 -0.0223000 0.00340000 -150.310
> 129.420 -13.1000
> 0.910200 0.413500 -0.0225000 0.00350000 -150.160
> 129.620 -13.0900
> 0.910200 0.413500 -0.0224000 0.00240000 -150.090
> 129.720 -13.0100
> 0.930600 0.365500 -0.0216000 0.00170000 -147.800
> 125.760 -16.7500
> 0.930500 0.365600 -0.0220000 0.000900000 -147.650
> 125.160 -16.6800
> 0.930500 0.365700 -0.0222000 0.00230000 -147.930
> 125.370 -16.8100
> 0.930500 0.365700 -0.0217000 0.00280000 -148.090
> 125.750 -16.8600
> 0.930400 0.365800 -0.0225000 0.00240000 -147.800
> 125.400 -16.8200
> 0.930400 0.365800 -0.0213000 0.00430000 -148.490
> 124.950 -16.7800
> 0.930400 0.365800 -0.0220000 0.00210000 -147.910
> 126.000 -16.7200
> 0.930400 0.365800 -0.0220000 0.00200000 -147.830
> 125.560 -16.6900
> 0.930400 0.365900 -0.0216000 0.00250000 -148.080
> 125.490 -16.7700
> 0.930400 0.365800 -0.0224000 0.00230000 -147.870
> 125.980 -16.6200
> 0.897600 0.439600 -0.0331000 0.00790000 -147.060
> 130.970 -6.02000
> 0.897600 0.439500 -0.0334000 0.00720000 -146.790
> 130.520 -6.13000
> 0.897500 0.439600 -0.0337000 0.00770000 -146.820
> 130.660 -6.13000
> 0.897500 0.439600 -0.0328000 0.00750000 -147.160
> 130.790 -6.13000

> 0.897600 0.439600 -0.0331000 0.00680000 -146.860
> 130.570 -6.07000
> 0.897600 0.439600 -0.0335000 0.00700000 -146.830
> 130.660 -6.12000
> 0.897600 0.439500 -0.0326000 0.00750000 -147.090
> 130.870 -6.08000
> 0.897600 0.439600 -0.0327000 0.00750000 -146.880
> 130.610 -6.14000
> 0.897600 0.439500 -0.0336000 0.00810000 -146.980
> 130.560 -6.25000
> 0.897600 0.439500 -0.0331000 0.00800000 -147.130
> 130.820 -6.19000
> 0.897500 0.439600 -0.0332000 0.00800000 -147.000
> 130.600 -6.25000
> 0.871700 0.488800 -0.0332000 0.0102000 -146.260
> 133.480 -1.14000
> 0.871600 0.488900 -0.0330000 0.0111000 -146.390
> 133.540 -1.29000
> 0.871600 0.488900 -0.0347000 0.00920000 -145.690
> 132.630 -1.26000
> 0.871700 0.488800 -0.0337000 0.0103000 -146.100
> 133.330 -1.44000
> 0.871700 0.488700 -0.0336000 0.0104000 -146.310
> 133.610 -1.58000
> 0.871700 0.488800 -0.0340000 0.00950000 -145.820
> 132.840 -1.33000
> 0.872000 0.488200 -0.0335000 0.00960000 -146.040
> 133.140 -1.95000
> 0.872000 0.488200 -0.0330000 0.00820000 -145.910
> 133.210 -1.83000
> 0.872000 0.488300 -0.0333000 0.0100000 -146.040
> 133.110 -1.82000
> 0.872100 0.488200 -0.0330000 0.00880000 -146.000
> 133.150 -1.83000
> 0.872000 0.488200 -0.0335000 0.00900000 -145.910
> 133.210 -1.85000
> 0.873000 0.487300 -0.0227000 0.000700000 -143.720
> 132.260 -6.08000
> 0.872900 0.487300 -0.0230000 0.000100000 -143.630
> 132.350 -6.07000
> 0.872900 0.487300 -0.0235000 0.000500000 -143.560
> 132.370 -6.14000
> 0.872900 0.487300 -0.0234000 -0.000300000 -143.430
> 132.520 -6.15000
> 0.872900 0.487300 -0.0231000 0.000700000 -143.670
> 132.280 -6.15000
> 0.872900 0.487300 -0.0237000 0.000200000 -143.480
> 132.430 -6.07000

>   0.872900   0.487300   -0.0231000 0.000500000   -143.550
> 132.450   -6.03000
>   0.872900   0.487300   -0.0241000 -0.000200000   -143.440
> 132.450   -6.11000
>   0.872900   0.487300   -0.0237000 0.000400000   -143.470
> 132.490   -6.05000
>   0.873000   0.487300   -0.0228000 0.000600000   -143.700
> 132.270   -6.03000
>   0.872900   0.487300   -0.0235000 -0.000200000   -143.430
> 132.450   -6.10000
>
> IDL> weights=clust_wts(pos1,n_clusters=5)
> IDL> print,weights
>   0.159265   0.119451   0.113155   0.180601   0.0680267
> 0.243488   0.116014
>   0.874568   0.483835   -0.0256644   0.00231699   -144.219
> 132.388   -5.40240
>   0.113501   0.127323   0.0985566   0.247231   0.225678
> 0.0779656   0.109745
>   0.238006   0.236222   0.127174   0.0261984   0.266028
> 0.0180832   0.0882878
>   0.0301962   0.232814   0.209770   0.146116   0.235975
> 0.134589   0.0105386
>
> IDL> result=cluster(pos1,weights,n_clusters=5)
> IDL> print,result(uniq(result))
>   1
>
> The 5 clusters are pretty distinct but "cluster" does a hopeless job
> identifying them. I tried scaling the data but that too finds 3
> clusters in the end. Any ideas?
>
> On Jul 18, 3:07 pm, Conor <cmanc...@gmail.com> wrote:
>
>> On Jul 17, 11:36 am, nivedita.raghun...@gmail.com wrote:
>
>>> Hi all,
>
>>> I am trying to cluster a (7 x n) array with n_clusters=5. Visually I
>>> can see 5 distinct clusters, but when I do clust_wts the cluster
>>> centroids don't end up right. No matter what options I give, clust_wts
>>> refuses to find the clusters.
>
>>> Any idea on whats going on ?
>
>>> Thanks,
>>> Nivedita
>

>> No idea. More information might be helpful. It's quite possible
>> though that the the clust_wts algorithm just doesn't work for your
>> particular data set, at least not as well as you apparently want it to.

The scales of all the variables are very different. Looks like
everything is in the second cluster, due to the large (abs value) of
the 5th dimension.

You might want to standardize() your data first

Subject: Re: clustering
Posted by nivedita.raghunath on Mon, 23 Jul 2007 13:45:25 GMT
View Forum Message <> Reply to Message

> You might want to standardize() your data first

What is standardizing? I did scale my data but it still found only 3
clusters out of 5. Scaled using scaled=(pos[1,*]-min(pos[1,*]))/
(max(pos[1,*])-min(pos[1,*]))

Am I scaling right? Again, What is standardizing?

Thanks

Subject: Re: clustering
Posted by Conor on Mon, 23 Jul 2007 13:49:53 GMT
View Forum Message <> Reply to Message

On Jul 23, 9:45 am, nivedita.raghun...@gmail.com wrote:
>> You might want to standardize() your data first
>
> What is standardizing? I did scale my data but it still found only 3
> clusters out of 5. Scaled using scaled=(pos[1,*]-min(pos[1,*]))/
> (max(pos[1,*])-min(pos[1,*]))
>
> Am I scaling right? Again, What is standardizing?
>
> Thanks

standardize() is a built in function in IDL. You can read about it
and what it does in online_help

Subject: Re: clustering

Posted by little davey on Mon, 23 Jul 2007 16:08:34 GMT
View Forum Message <> Reply to Message

On Jul 17, 10:36 am, nivedita.raghun...@gmail.com wrote:
> Hi all,
>
> I am trying to cluster a (7 x n) array with n_clusters=5. Visually I
> can see 5 distinct clusters, but when I do clust_wts the cluster
> centroids don't end up right. No matter what options I give, clust_wts
> refuses to find the clusters.
>
> Any idea on whats going on ?
>
> Thanks,
> Nivedita

I seem to recall that "clust_wts.pro" just doesn't work. Because of
"work" I don't have time to dig into it, but I have glanced at a file
I have called "myclust_wts.pro" that I copied from the IDL directory.
I recall having stepped through the program, and I found that, oh, I
think it didn't "converge" or something like that. I hate saying this
without having the decency to do my homework, but my memory is that
CLUST_WTS just does not work. The algorithm I thought was not
adequate. I'm looking at where it says:

```
  ;Normalized uniformly random cluster weights.
  ;;Weights = RANDOMU(SEED, N_Variables, N_Clusters) + Zero
      av1 = average(array[0,*])
      av2 = average(array[1,*])
  Weights = RANDOMU(SEED, N_Variables, N_Clusters) + Zero
  for k = 0L, N_Clusters-1 do Weights[*,k] = $
    (Weights[*,k] / TOTAL(Weights[*,k])) * Variable_Wts
```

I recall something like the weights and the data simply didn't have
the same data range, so the algorithm failed. Sorry if I'm totally
wrong.

-- Dave K --

Subject: Re: clustering
Posted by little davey on Mon, 23 Jul 2007 17:14:49 GMT
View Forum Message <> Reply to Message

Is it the case that you MUST use standardize() before you call
CLUST_WTS()? The documentation does not say so, but I suspect from
the code, and, I actually tried it with the initial poster's data, and
got 4 clusters (he wanted 5, but this would appear to be a tough data

set to cluster, as variables "2" and "3" are close to each other).

As I posted an hour ago, part of the source code for CLUST_WTS() is:

```
  ;Normalized uniformly random cluster weights.
  ;;Weights = RANDOMU(SEED, N_Variables, N_Clusters) + Zero
      av1 = average(array[0,*])
      av2 = average(array[1,*])
  Weights = RANDOMU(SEED, N_Variables, N_Clusters) + Zero
  for k = 0L, N_Clusters-1 do Weights[*,k] = $
    (Weights[*,k] / TOTAL(Weights[*,k])) * Variable_Wts
```

However, the variables AV1 and AV2 are NOT USED ANYWHERE IN THE CODE, so I suspect that the data is not "normalized" correctly in CLUST_WTS. The use of STANDARDIZE() may be necessary for CLUST_WTS to work.

-- Dave K --

Subject: Re: clustering
Posted by James Kuyper on Mon, 23 Jul 2007 18:16:25 GMT
View Forum Message <> Reply to Message

little davey wrote:
> Is it the case that you MUST use standardize() before you call
> CLUST_WTS()?

No, you don't need to standardize. The scaling of the data affects the results you're going to get. The algorithm is built to treat a difference of 1.0 in a variable between two data points as being equally significant, no matter which of the variables that difference occurs in. This is a fine assumption for variables that have equivalent meanings; such as the x, y, and z coordinates when you're clustering stars.

However, in most contexts for most variables that's simply not true. You can use the scaling to tell CLUST_WTS() treat differences in one variable as more important than differences in another variable. That's fine if you have a clear idea as to which variables are more important than others, and by what factor.

However, the most common case is where the analyst doesn't have clear advance knowledge of the relative importance of the different variables; the analysis is being done to get some idea as to which are the important variables. Scaling the variables according their standard deviations, as STANDARDIZE() does,provides a comforting illusion of objectivity to the choice of scale factors. Unfortunately,

if there are only very small variations in an unimportant variable,
standardizing it would give that variable undue importance in the
clustering. Nothing can substitute for good judgment on the part of
the analyst. However, STANDARDIZE() does produce fairly good results
in many cases, which implies that the objectivity it provides not
quite as illusionary as I've suggested.

Subject: Re: clustering
Posted by Vince Hradil on Mon, 23 Jul 2007 18:22:42 GMT
View Forum Message <> Reply to Message

On Jul 23, 12:14 pm, little davey <dave-kel...@cox.net> wrote:
> Is it the case that you MUST use standardize() before you call
> CLUST_WTS()? The documentation does not say so, but I suspect from
> the code, and, I actually tried it with the initial poster's data, and
> got 4 clusters (he wanted 5, but this would appear to be a tough data
> set to cluster, as variables "2" and "3" are close to each other).
>
> As I posted an hour ago, part of the source code for CLUST_WTS() is:
>
> ;Normalized uniformly random cluster weights.
> ;;Weights = RANDOMU(SEED, N_Variables, N_Clusters) + Zero
> av1 = average(array[0,*])
> av2 = average(array[1,*])
> Weights = RANDOMU(SEED, N_Variables, N_Clusters) + Zero
> for k = 0L, N_Clusters-1 do Weights[*,k] = $
> (Weights[*,k] / TOTAL(Weights[*,k])) * Variable_Wts
>
> However, the variables AV1 and AV2 are NOT USED ANYWHERE IN THE CODE,
> so I suspect that the data is not "normalized" correctly in
> CLUST_WTS. The use of STANDARDIZE() may be necessary for CLUST_WTS to
> work.
>
> -- Dave K --

No, you just have to be careful about your data. Here's an example:

```
IDL> a=[[3,55],[4,54],[8,55],[9,56]]
IDL> plot, a[0,*], a[1,*], psym=4
```

"obviously", there are two clusters.

```
IDL> wts= clust_wts(a,n_clusters=2)
IDL> print, wts
     1.99307      27.5069
     6.42612      55.0837
IDL> oplot, wts[0,*], wts[1,*], psym=5
```

(one of the points doesn't even show up, oh, and fix the xrange, too)
IDL> plot, a[0,*], a[1,*], psym=4, xrange=[0,60], yrange=[0,60]
(uh-oh, maybe there is only one cluster)
IDL> oplot, wts[0,*], wts[1,*], psym=5
IDL> print, cluster(a,wts,n_clusters=2)
1
1
1
1
D'oh!

NOW:
IDL> sa = standardize(float(a))
IDL> plot, sa[0,*], sa[1,*], psym=4
IDL> swts= clust_wts(sa,n_clusters=2)
IDL> print, swts
-0.833454 -0.669169
0.864960 0.669170
IDL> oplot, swts[0,*], swts[1,*], psym=5
IDL> print, cluster(sa,swts,n_clusters=2)
0
0
1
1
Tada!

The reason the first (unstandardized) doesn't work is that the overall variance is much larger than the variance within each coordinate, so the clusters get "attracted" (if you will) to one of the coordinates and disregards the other.

You have to "stay in touch" with your data - Black boxes are okay, as long as you know what's going on inside.

Subject: Re: Clustering
Posted by on Fri, 28 Oct 2011 17:07:42 GMT
View Forum Message <> Reply to Message

Hi again,

Am 28.10.2011 14:29, schrieb Kai Muehlbauer:
> I stumbled over CLUSTER and CLUST_WTS and tried to get somthing useful
> out, but failed so far. For every stripe of 360 histograms I calculated
> the weights and CLUSTER, but this did not seem the right approach.

maybe this striped approach is wrong. I could insert the 2000*360 histograms of length 150 all in one 2D-Array and use then CLUSTER_WTS

and CLUSTER to group similar histograms. Then the weights are computed
over all histograms. So if someone can check if I'm on the right track
now or completely wrong...

Cheers,
Kai

P.S. I used Davids cgHistoplot to check several histograms which gave
the impression that such clustering might help for further processing
steps. But doing that for all would take for to long I think ;-)

Subject: Re: Clustering
Posted by kidpix on Mon, 31 Oct 2011 11:56:06 GMT
View Forum Message <> Reply to Message

Hi Kai,
to mee it make a lot of sense. It looks similar to what I'm doing with spectra.

I assume you have a 50 bands - 2000x360=720000 pixels images.

I'm using CLUSTER_TREE to classify all the spectra, there is as sample of what I do.

```
;- fake data
seed_value = 5L
K_max = 20
J_max = 36
A = RANDOMN(seed, K_max, 50, J_max)

;-- histogram from pixel K=J=0 A[K,*,J]
 K=0
 J=0
 plot,HISTOGRAM(A[K,*,J],BINSIZE=(max(A)-min(A))/149,MIN=min( A),MAX=max(A)),psym=10

help,A(*,0,*) ;-- 2000x360 pixels images
help,A(0,*,0) ;-- each image has 50 bands

;-- assemble the histogram array
Histo_A = fltarr(K_max, 150, J_max)

for K=0,K_max-1 do  for J=0,J_max-1 do   Histo_A[K,*,J]=
HISTOGRAM(A[K,*,J],BINSIZE=(max(A)-min(A))/149,MIN=min(A),MA X=max(A))

help,Histo_A

;-- arrange the data as a [K_max*J_max, 150] array
Histo_A_array = fltarr(K_max*J_max, 150)
```

```
 for K=0,K_max-1 do  for J=0,J_max-1 do Histo_A_array[K*J,*] = Histo_A[K,*,J]

;-- Clusterization using cluster_tree
distance_matrix = DISTANCE_MEASURE(Histo_A_array, MEASURE = 0,/matrix
clusters_dend   =  CLUSTER_TREE(distance_matrix,linkdistance,LINKAGE=2,data=His
to_A_array,MEASURE=0)

cluster_matrix  = cluster_member(clusters_dend)
```

The only problem is that you have to explore the whole cluster_matrix and decide which are meaningful clusters to you.
[the cluster_member comes from https://groups.google.com/forum/#!topic/comp.lang.idl-pvwave/UUVL0MkS0zE]

Cheers,
Mario.

Subject: Re: Clustering
Posted by Kai Muehlbauer on Mon, 31 Oct 2011 15:40:10 GMT
View Forum Message <> Reply to Message

Hi all,

I took a big step forward.

I slightly changed my histograms. I reduced the number of bins by increasing the binsize. I cut off noise before the histograms which also reduces number of bins. Then I fill the histograms in an array similar to Mario is doing.

```
FOR K=0L, 1999 DO BEGIN
  FOR J=0L, 359 DO BEGIN
    array = REFORM(source[K,*,J)]
    hist_arr = HISTOGRAM(array,BINSIZE=0.5, MAX=7.5, MIN=0)
    Array[*,K*360L+J] = hist_arr
  ENDFOR
ENDFOR
```

Then the weights for 10 Clusters are calculated and CLUSTER is called

```
weights = CLUST_WTS(array2, N_CLUSTERS = 10)
tmp_result = CLUSTER(array1, weights, N_CLUSTERS = 10)
```

Then the data needs REFORMing

result1 = REFORM(tmp_result,360,2000)

and in my case the dimensions need to be interchanged

FOR I=0,range1 - 1 DO BEGIN
result[I,*] = result1[*,I]

Anyway the results were not useful. I noticed that a great deal (about 90 percent) of the histograms could be grouped into one cluster. So I reduced the histograms used to calculate the weights to a reasonable amount, to get better weights also for the remaining 10 percent.

After that I get quite usable clusters of my data. I think with a little tweaking there should be even better results.

There should also be a speedup possible in the above code. But I'am still in the learning phase, so a little help is appreciated. I still struggle with those dimensions.

Thanks Mario for providing your example. I tried this but got an out of memory error while calculating the distance matrix. But that was before my reduction of histogram number of bins. I will test this later and come back with some results in november ;-)

Cheers,
Kai

Subject: Re: Clustering
Posted by on Tue, 01 Nov 2011 10:01:24 GMT
View Forum Message <> Reply to Message

Sorry,

there were some variables which are out of context. So here is my code again:

```
>  FOR K=0L, 1999 DO BEGIN
>    FOR J=0L, 359 DO BEGIN
>      array = REFORM(source[K,*,J)]
>      hist_arr = HISTOGRAM(array,BINSIZE=0.5, MAX=7.5, MIN=0)
>      array1[*,K*360L+J] = hist_arr
>    ENDFOR
>  ENDFOR
 > ;array2 is similar array1 but with reduced number of histograms
 > ;to get better weights
>  weights = CLUST_WTS(array2, N_CLUSTERS = 10)
>  tmp_result = CLUSTER(array1, weights, N_CLUSTERS = 10)
```

> result1 = REFORM(tmp_result,360,2000)
> FOR I=0L, 1999 DO BEGIN
> result[I,*] = result1[*,I]

Subject: Re: Clustering
Posted by Jeremy Bailin on Tue, 01 Nov 2011 19:16:53 GMT
View Forum Message <> Reply to Message

On 10/31/11 11:40 AM, Kai Muehlbauer wrote:
> Hi all,
>
> I took a big step forward.
>
> I slightly changed my histograms. I reduced the number of bins by
> increasing the binsize. I cut off noise before the histograms which also
> reduces number of bins. Then I fill the histograms in an array similar
> to Mario is doing.
>
> FOR K=0L, 1999 DO BEGIN
> FOR J=0L, 359 DO BEGIN
> array = REFORM(source[K,*,J)]
> hist_arr = HISTOGRAM(array,BINSIZE=0.5, MAX=7.5, MIN=0)
> Array[*,K*360L+J] = hist_arr
> ENDFOR
> ENDFOR
>
> Then the weights for 10 Clusters are calculated and CLUSTER is called
>
> weights = CLUST_WTS(array2, N_CLUSTERS = 10)
> tmp_result = CLUSTER(array1, weights, N_CLUSTERS = 10)
>
> Then the data needs REFORMing
>
> result1 = REFORM(tmp_result,360,2000)
>
> and in my case the dimensions need to be interchanged
>
> FOR I=0,range1 - 1 DO BEGIN
> result[I,*] = result1[*,I]
>
> Anyway the results were not useful. I noticed that a great deal (about
> 90 percent) of the histograms could be grouped into one cluster. So I
> reduced the histograms used to calculate the weights to a reasonable
> amount, to get better weights also for the remaining 10 percent.
>
> After that I get quite usable clusters of my data. I think with a little
> tweaking there should be even better results.

>
> There should also be a speedup possible in the above code. But I'am
> still in the learning phase, so a little help is appreciated. I still
> struggle with those dimensions.
>
> Thanks Mario for providing your example. I tried this but got an out of
> memory error while calculating the distance matrix. But that was before
> my reduction of histogram number of bins. I will test this later and
> come back with some results in november ;-)
>
> Cheers,
> Kai
>

You can probably use JD's HIST_ND to get rid of those for loops, which should speed things up.

-Jeremy.

Subject: Re: clustering
Posted by Russell[1] on Wed, 07 Mar 2012 17:32:18 GMT
View Forum Message <> Reply to Message

On Mar 6, 7:21 pm, kisCA <ki...@hotmail.com> wrote:
> Hi there,
>
> Why is it not possible to use the cluster function in n=1 dimension ?
> Is there anyways to make it possible ?
>
> Thanks for any kind of help

Can the function label_region work for you?

Russell

Subject: Re: clustering
Posted by kisCA on Wed, 07 Mar 2012 21:37:24 GMT
View Forum Message <> Reply to Message

Thank you Russel but it is not really accurate to detect thin structures in the image

---