Subject: Re: KMean Clustering of RGB Images
Posted by Mort Canty on Tue, 08 May 2007 13:13:38 GMT
View Forum Message <> Reply to Message

helaha@gmx.net schrieb:
> Recently I tested the Kmeans clustering of an RGB image with the
> CLUSTER and CLUST_WTR commands.
> The results were not very promising, because they depended on the
> image orientation and on the number of image pixels.
> I'm not a specialist in clustering, but in my opinion the results
> should be identical, especially regardless of the order of the data
> points.
>
Not sure what you mean by "order of the data points". But any clustering algorithm that minimizes a cost function (like k-means) can get trapped in a local minimum. The IDL help for CLUST_WTR says

"Because the initial clusters are chosen randomly, your results may differ slightly each time the CLUST_WTS routine is invoked, even for the same input data. For data with well-defined clusters the differences should be slight. For randomly-scattered data (no distinguishable clusters), the results may be significantly different, which may indicate that k-means clustering is not appropriate for your data."

- Mort

Subject: Re: KMean Clustering of RGB Images
Posted by helaha on Wed, 09 May 2007 10:00:42 GMT
View Forum Message <> Reply to Message

> Not sure what you mean by "order of the data points". But any clustering
> algorithm that minimizes a cost function (like k-means) can get trapped
> in a local minimum.

Thank you Mort,
concerning the "order of data points" it is necessary to think about the construction of the RGB Data space. The original RGB image is a 3 dim. array e.g. [3,ImageSizeX, ImageSizeY]. The RGB Data space, which should be clustered concerning only the grey values and without consideration of the pixel locations, has two dimensions e.g. [3, NumberOfPixels]. Therefore the individual image pixels are transformed in a sequence of RGB triple values ("data points"). The first triple values correspond to the pixel located at [3, 0, 0] e.g. the left bottom corner of the image. If the image is mirrored, rotated ore transposed before RGB space construction, then the first RGB triple value will be altered. Nevertheless all the individual image pixels were transformed into the RGB space, only the sequence is

changed.

Thanks,
Helmut

Subject: Re: KMean Clustering of RGB Images
Posted by Mort Canty on Wed, 09 May 2007 11:36:57 GMT
View Forum Message <> Reply to Message

helaha@gmx.net schrieb:

> Thank you Mort,
> concerning the "order of data points" it is necessary to think about
> the construction of the RGB Data space. The original RGB image is a 3
> dim. array e.g. [3,ImageSizeX, ImageSizeY]. The RGB Data space, which
> should be clustered concerning only the grey values and without
> consideration of the pixel locations, has two dimensions e.g. [3,
> NumberOfPixels]. Therefore the individual image pixels are
> transformed in a sequence of RGB triple values ("data points"). The
> first triple values correspond to the pixel located at [3, 0, 0] e.g.
> the left bottom corner of the image. If the image is mirrored, rotated
> ore transposed before RGB space construction, then the first RGB
> triple value will be altered. Nevertheless all the individual image
> pixels were transformed into the RGB space, only the sequence is
> changed.
>
> Thanks,
> Helmut
>
>

Hi Helmut,

Now I see what you were getting at. But I think you're confusing things
a bit. K-means clustering takes place entirely in the feature space
(3-D RGB space in your case) and makes no reference to any ordering of
the pixels, spatial or otherwise. The Euclidean distances from each
pixel to each cluster mean are determined and the pixel is assigned to
the cluster for which the distance is smallest. Then the means are
recalculated and the procedure repeated until the means cease to change.
The order in which the distances are calculated in each iteration
clearly plays no role. The indeterminacy arises from the random
initialization of the cluster means.

Cheers

Mort

Subject: Re: KMean Clustering of RGB Images
Posted by helaha on Wed, 09 May 2007 12:08:30 GMT
View Forum Message <> Reply to Message

> The order in which the distances are calculated in each iteration
> clearly plays no role.
>
> Cheers
>
> Mort

Hello Mort,
yes, the order should not play any role, but nevertheless the
clustering of an RGB image, implemented with the example code above
showed a dependency and that's my problem. I can't imagine why the
CLUSTER_WTS Algorithm is sensitive to the sequence order.
The clustering results are very stable and I think really identical,
if the algorithm is started several times without changing the data
sequence. Changing the data sequence leads to a very distinct and yet
again very stable clustering result. But why is there this dependency
on data sequence?

Many thanks,
Helmut

Subject: Re: KMean Clustering of RGB Images
Posted by Mort Canty on Wed, 09 May 2007 14:43:05 GMT
View Forum Message <> Reply to Message

helaha@gmx.net schrieb:

> Hello Mort,
> yes, the order should not play any role, but nevertheless the
> clustering of an RGB image, implemented with the example code above
> showed a dependency and that's my problem. I can't imagine why the
> CLUSTER_WTS Algorithm is sensitive to the sequence order.
> The clustering results are very stable and I think really identical,
> if the algorithm is started several times without changing the data
> sequence. Changing the data sequence leads to a very distinct and yet
> again very stable clustering result. But why is there this dependency
> on data sequence?
>
> Many thanks,
> Helmut
>

Yes, I get the same effect. I had difficulty following your code so I

reduced it to essentials (I hope):

```
PRO Cluster_KMean_Image
    ImageName = DIALOG_PICKFILE()
    CD, FILE_DIRNAME(ImageName)
    READ_JPEG, ImageName, RGBImage
    Info = SIZE(RGBImage)
    ImageSizeX = Info[2]
    ImageSizeY = Info[3]
    PixelN = ImageSizeX*ImageSizeY
;  optional
;    RGBImage = TRANSPOSE(RGBImage, [0, 2, 1])
    print, CLUST_WTS(REFORM(RGBImage, 3, PixelN), $
                N_CLUSTERS = 2, N_ITERATIONS = 20)

END
```

It just prints out the cluster means. On a small JPEG image they are nicely reproducible. But they should also be the same whether the TRANSPOSE function is called or not. And as you say, they aren't. I hesitate to think that there is a bug in CLUSTER_WTS, but just now I don't see an alternative explanation.

- Mort
