
Subject: Fuzzy classification

Posted by [joao.antunes](#) on Fri, 14 May 2004 15:34:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

I am looking for an IDL implementation of the fuzzy c-means (FCM) clustering algorithm and fuzzy maximum likelihood algorithm (FML) to classification of satellite images. The resulted cluster centers obtained by the FCM algorithm from the first step are input into the FML algorithm.

Any help would be greatly appreciated.

Thanks in advance.

Joao.

Subject: Re: Fuzzy classification

Posted by [Vincenzo Positano](#) on Tue, 18 May 2004 12:08:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Joao F." <joao.antunes@agr.unicamp.br> ha scritto nel messaggio
news:7568d93c.0405140734.5b3e4df4@posting.google.com...

> Hello,

>

> I am looking for an IDL implementation of the fuzzy c-means (FCM)
> clustering algorithm and fuzzy maximum likelihood algorithm (FML) to
> classification of satellite images. The resulted cluster centers
> obtained by the FCM algorithm from the first step are input into the
> FML algorithm.

>

This is an implementation of FCM for image segmentation.
mu initialization should be adapted to your problem.
The function seems to work on medical images with 3-4
classes.

Vincenzo Positano

```
;image segmentation based on fuzzy clustering
;
; image          input image
; num_class      number of classes
; m              fuzzy factor
; eps            threshold
```

```

; n_max      max iteration number
; a          cluster centers

function fuzzy_clustering,image,num_class,m,eps,n_max,CENTER=a

sz=SIZE(image,/DIMENSIONS)
x = REFORM(image,sz[0]*sz[1])
n=LONG(N_ELEMENTS(x))

c=num_class
a=DBLARR(c)
mu=DBLARR(c,n)

kk=0

;mu initialization
idx=WHERE(image le 20)
mu[0,idx]=1.0
idx=WHERE((image ge 20) and (image lt 200))
mu[1,idx]=1.0
idx=WHERE(image ge 200)
mu[2,idx]=1.0

mu_old=DBLARR(c,n)

me=(2/(m-1.0))

while (kk le n_max) do begin
  for i=0L,c-1 do begin
    tmp=REFORM(mu[i,*],n)
    num = tmp^m*x
    den = tmp^m
    a[i]=TOTAL(num)/TOTAL(den)
  endfor

  for i=0L,c-1 do begin
    num = DBLARR(c,n)
    den = DBLARR(c,n)
    num[0,*]=(ABS(x-a[i]))^me
    for k=0L,c-1 do begin
      den[k,*]=(ABS(x-a[k]))^me
      num[k,*]=num[0,*]
    endfor
    pippo=TOTAL(num/den,1)
    mu[i,*]=1.0/pippo
  endfor

  diff = TOTAL(ABS(mu-mu_old))/FLOAT(n)

```

```
    if (diff le eps) THEN KK=n_max  
    mu_old=mu  
    kk=kk+1  
  
endwhile  
  
return,mu  
  
end
```
