Subject: faster minimization needed - maybe mpfit?
Posted by rogass on Mon, 26 Mar 2012 13:15:30 GMT
View Forum Message <> Reply to Message

Hi folks,
the following expression runs successfully with AMOEBA but requires
for large matrices (columns < 512, rows up to 30000), for small
tolerances (e.g. ftol=1e-06) and a high number of iterations
(nmax>=10000) to converge years:

expr = total(abs(convol(im-rebin(p[*],size(im,/dim),/samp),
[-1.,0.,1.])))

where p is the parameter vector (one row) to be found and im is the
matrix.

Is there a way to do it faster? Maybe with mpfit (I don't get an idea
how...)

Thanks for any help

CR

Subject: Re: faster minimization needed - maybe mpfit?
Posted by Craig Markwardt on Mon, 26 Mar 2012 19:04:25 GMT
View Forum Message <> Reply to Message

On Monday, March 26, 2012 9:15:30 AM UTC-4, chris wrote:
> Hi folks,
> the following expression runs successfully with AMOEBA but requires
> for large matrices (columns < 512, rows up to 30000), for small
> tolerances (e.g. ftol=1e-06) and a high number of iterations
> (nmax>=10000) to converge years:
>
> expr = total(abs(convol(im-rebin(p[*],size(im,/dim),/samp),
> [-1.,0.,1.])))
>
> where p is the parameter vector (one row) to be found and im is the
> matrix.
>
> Is there a way to do it faster? Maybe with mpfit (I don't get an idea
> how...)

If you can express your problem as minimize{TOTAL(RESID^2)}, then you can use MPFIT, where RESID is signed. In your case you can do this, but there's a few little tricks.

Your problem looks like minimize{TOTAL(ABS(XXX))}.

You might want to define RESID=SQRT(ABS(XXX)), and then in principle it looks like an MPFIT problem. Unfortunately you need to preserve the sign of XXX. So this is what you do:
RESID = SIGN(XXX)*SQRT(ABS(XXX))
where SIGN(XXX) is the sign of XXX (-1 or +1 depending on XXX).

Happy equation solving...
Craig

Subject: Re: faster minimization needed - maybe mpfit?
Posted by rogass on Mon, 26 Mar 2012 19:59:41 GMT
View Forum Message <> Reply to Message

On 26 Mrz., 21:04, Craig Markwardt <craig.markwa...@gmail.com> wrote:
> On Monday, March 26, 2012 9:15:30 AM UTC-4, chris wrote:
>> Hi folks,
>> the following expression runs successfully with AMOEBA but requires
>> for large matrices (columns < 512, rows up to 30000), for small
>> tolerances (e.g. ftol=1e-06) and a high number of iterations
>> (nmax>=10000) to converge years:
>
>> expr = total(abs(convol(im-rebin(p[*],size(im,/dim),/samp),
>> [-1.,0.,1.])))
>
>> where p is the parameter vector (one row) to be found and im is the
>> matrix.
>
>> Is there a way to do it faster? Maybe with mpfit (I don't get an idea
>> how...)
>
> If you can express your problem as minimize{TOTAL(RESID^2)}, then you can use MPFIT, where RESID is signed. In your case you can do this, but there's a few little tricks.
>
> Your problem looks like minimize{TOTAL(ABS(XXX))}.
>
> You might want to define RESID=SQRT(ABS(XXX)), and then in principle it looks like an MPFIT problem. Unfortunately you need to preserve the sign of XXX. So this is what you do:
> RESID = SIGN(XXX)*SQRT(ABS(XXX))
> where SIGN(XXX) is the sign of XXX (-1 or +1 depending on XXX).
>
> Happy equation solving...
> Craig

Hi Craig,
thank you. Nevertheless, I don't think that I understood what you
suggests. So, i tried this:

function test2,p,x=x,err=err
temp=convol(x,rebin(p[*],size(x,/dim)))
return,signum(temp)*sqrt(abs(temp))
end

But what I got is this:

ENVI> st={x:im}&help,mpfit('test2',functargs=st,maxiter=100)
<Expression>   DOUBLE   =   NaN

What's wrong?

Thank you

Chris

Subject: Re: faster minimization needed - maybe mpfit?
Posted by Craig Markwardt on Mon, 26 Mar 2012 22:36:43 GMT
View Forum Message <> Reply to Message

On Monday, March 26, 2012 3:59:41 PM UTC-4, chris wrote:
> On 26 Mrz., 21:04, Craig Markwardt <craig.markwa...@gmail.com> wrote:
>> On Monday, March 26, 2012 9:15:30 AM UTC-4, chris wrote:
>>> Hi folks,
>>> the following expression runs successfully with AMOEBA but requires
>>> for large matrices (columns < 512, rows up to 30000), for small
>>> tolerances (e.g. ftol=1e-06) and a high number of iterations
>>> (nmax>=10000) to converge years:
>>
>>> expr = total(abs(convol(im-rebin(p[*],size(im,/dim),/samp),
>>> [-1.,0.,1.])))
>>
>>> where p is the parameter vector (one row) to be found and im is the
>>> matrix.
>>
>>> Is there a way to do it faster? Maybe with mpfit (I don't get an idea
>>> how...)
>>
>> If you can express your problem as minimize{TOTAL(RESID^2)}, then you can use MPFIT, where RESID is signed. In your case you can do this, but there's a few little tricks.
>>
>> Your problem looks like minimize{TOTAL(ABS(XXX))}.
>>
>> You might want to define RESID=SQRT(ABS(XXX)), and then in principle it looks like an MPFIT problem. Unfortunately you need to preserve the sign of XXX. So this is what you do:
>> RESID = SIGN(XXX)*SQRT(ABS(XXX))
>> where SIGN(XXX) is the sign of XXX (-1 or +1 depending on XXX).

>>
>> Happy equation solving...
>> Craig
>
> Hi Craig,
> thank you. Nevertheless, I don't think that I understood what you
> suggests. So, i tried this:
>
> function test2,p,x=x,err=err
> temp=convol(x,rebin(p[*],size(x,/dim)))
> return,signum(temp)*sqrt(abs(temp))
> end
>
> But what I got is this:
>
> ENVI> st={x:im}&help,mpfit('test2',functargs=st,maxiter=100)
> <Expression> DOUBLE = NaN
>
> What's wrong?

Problem #1. You need to provide starting values, for P, just like for AMOEBA.

Problem #2. You changed the function. Your residual in your original post was of the form convol(im-rebin(p)). Why did you change it?

Issue #3. Error checking. Use the STATUS and ERRMSG keywords to retrieve more error information about what went wrong.

By the way, are you sure you want to solve a least absolute deviation problem? Or would you be satisfied with a least squares solution? Least squares is so much easier, for example you can use MPFITFUN().

Craig

Subject: Re: faster minimization needed - maybe mpfit?
Posted by rogass on Tue, 27 Mar 2012 12:28:02 GMT

View Forum Message <> Reply to Message

Hi Craig,
sorry I made several typos. I would also be satisfied with a least squares solution as you can see if you compare function test2 with the previous posts. The function I want to minimize is test2. It doesnt matter for me at this stage whether total(abs(resid)) or total(resid^2) is minimized.

function test2,p,xval=x,errval=err
resid=convol(x-rebin(p[*],size(x,/dim)),[-1.,0.,1.])

```
return,total(resid^2)
end

ENVI> help,im
IM              INT       = Array[512, 7237]
ENVI> sz=size(im,/dim)
ENVI>  im2=im+fix(1000.*rebin((((add=randomn(seed,sz[0])))-mean(add ))/
stddev(add),sz))
ENVI> help,im2
IM2             INT       = Array[512, 7237]
ENVI>  p0=((p0=total(im2,2)/float(sz[1])))-smooth(p0,3,/edge_trunc)
ENVI> help,p0
P0              FLOAT     = Array[512]
ENVI> st={x:im2,errval:sqrt(p0)}
 &res=mpfit('test2',p0,functargs=st,maxiter=100,status=st atus,errmsg=errmsg)
&print,status,string(10b),errmsg
          0
ERROR: number of parameters must not exceed data


THANKS in advance

CR
```

Subject: Re: faster minimization needed - maybe mpfit?
Posted by Craig Markwardt on Tue, 27 Mar 2012 13:38:36 GMT

View Forum Message <> Reply to Message

On Tuesday, March 27, 2012 8:28:02 AM UTC-4, chris wrote:

```
>  Hi Craig,
>  sorry I made several typos. I would also be satisfied with a least
>  squares solution as you can see if you compare function test2 with the
>  previous posts. The function I want to minimize is test2. It doesnt
>  matter for me at this stage whether total(abs(resid)) or
>  total(resid^2) is minimized.
>
>  function test2,p,xval=x,errval=err
>  resid=convol(x-rebin(p[*],size(x,/dim)),[-1.,0.,1.])
>  return,total(resid^2)
>  end
>
>  ENVI> help,im
>  IM              INT       = Array[512, 7237]
>  ENVI> sz=size(im,/dim)
>  ENVI>  im2=im+fix(1000.*rebin((((add=randomn(seed,sz[0])))-mean(add ))/
>  stddev(add),sz))
>  ENVI> help,im2
```

> IM2 INT = Array[512, 7237]
> ENVI> p0=((p0=total(im2,2)/float(sz[1])))-smooth(p0,3,/edge_trunc)
> ENVI> help,p0
> P0 FLOAT = Array[512]
> ENVI> st={x:im2,errval:sqrt(p0)}
> &res=mpfit('test2',p0,functargs=st,maxiter=100,status=st atus,errmsg=errmsg)
> &print,status,string(10b),errmsg
> 0
> ERROR: number of parameters must not exceed data

Check out the documentation for MPFIT. It expects your user function to return a 1D array of residuals, not the sum of squares.

Craig

Subject: Re: faster minimization needed - maybe mpfit?
Posted by rogass on Thu, 29 Mar 2012 06:45:24 GMT
View Forum Message <> Reply to Message

On 27 Mrz., 15:38, Craig Markwardt <craig.markwa...@gmail.com> wrote:
> On Tuesday, March 27, 2012 8:28:02 AM UTC-4, chris wrote:
>> Hi Craig,
>> sorry I made several typos. I would also be satisfied with a least
>> squares solution as you can see if you compare function test2 with the
>> previous posts. The function I want to minimize is test2. It doesnt
>> matter for me at this stage whether total(abs(resid)) or
>> total(resid^2) is minimized.
>
>> function test2,p,xval=x,errval=err
>> resid=convol(x-rebin(p[*],size(x,/dim)),[-1.,0.,1.])
>> return,total(resid^2)
>> end
>
>> ENVI> help,im
>> IM INT = Array[512, 7237]
>> ENVI> sz=size(im,/dim)
>> ENVI> im2=im+fix(1000.*rebin((((add=randomn(seed,sz[0])))-mean(add))/
>> stddev(add),sz))
>> ENVI> help,im2
>> IM2 INT = Array[512, 7237]
>> ENVI> p0=((p0=total(im2,2)/float(sz[1])))-smooth(p0,3,/edge_trunc)
>> ENVI> help,p0
>> P0 FLOAT = Array[512]
>> ENVI> st={x:im2,errval:sqrt(p0)}
>> &res=mpfit('test2',p0,functargs=st,maxiter=100,status=st atus,errmsg=errmsg)
>> &print,status,string(10b),errmsg
>> 0

>> ERROR: number of parameters must not exceed data
>
> Check out the documentation for MPFIT. It expects your user function to return a 1D array of residuals, not the sum of squares.
>
> Craig

Dear Craig,
now it works perfect. Thank you!

CR
