Subject: Radon forward projection problem
Posted by Wox on Tue, 22 Apr 2008 15:07:48 GMT
View Forum Message <> Reply to Message

Hi All,

I'm trying to implement an alternative for the FBP(filtered backprojection) method for reconstructing objects measured in tomography experiments. It's supposed to give less noisy tomograms.

Anyway, it's called OSEM and it's some iterative procedure using forward and backward projection until the real sinogram and the calculated sinogram are close.

For the projection, I use IDL's radon function. But I noticed something strange with the forward projection. Try the code below. It calculates the sinogram of a tomogram which is an image with all pixels equal to 1. If you look at the result, something strange is going on in the corners of the sinogram image. Does anyone know what causes it and whether it is an intrinsic radon transform problem?

I would like to get rid of it, because this "estsinogram" is calculated in each iteration of the OSEM (only in the first iteration on an image with 1's) and used to normalize the measured sinogram before adapting the tomogram. The resulting tomogram has some artifacts because of it.

Thanks in advance,

Wout

```
pro test
; Detector
N=80
projcen=(N-1)/2.

; Angles
anglestart=0.
anglerange=180.
NAngles=anglerange/2.

angleinc=anglerange/(NAngles-1)
angles=anglestart+angleinc*findgen(NAngles)
angles*=!pi/180

; Reconstructing an object with 1's
```

```
tomogram=replicate(1.,N,N)
 estsinogram=radon(tomogram,theta=angles,xmin=-projcen,RMIN=-
projcen,drho=1,NRHO=N,/LINEAR)

loadct,0
window
tvscl,not bytscl(rebin(estsinogram,NAngles*3,N*3,/sample))
end
```

Subject: Re: Radon forward projection problem
Posted by Vontobel Peter on Wed, 23 Apr 2008 07:42:07 GMT
View Forum Message <> Reply to Message

On 22 Apr., 17:07, Wox <nom...@hotmail.com> wrote:
> Hi All,
>
> I'm trying to implement an alternative for the FBP(filtered
> backprojection) method for reconstructing objects measured in
> tomography experiments. It's supposed to give less noisy tomograms.
>
> Anyway, it's called OSEM and it's some iterative procedure using
> forward and backward projection until the real sinogram and the
> calculated sinogram are close.
>
> For the projection, I use IDL's radon function. But I noticed
> something strange with the forward projection. Try the code below. It
> calculates the sinogram of a tomogram which is an image with all
> pixels equal to 1. If you look at the result, something strange is
> going on in the corners of the sinogram image. Does anyone know what
> causes it and whether it is an intrinsic radon transform problem?
>
> I would like to get rid of it, because this "estsinogram" is
> calculated in each iteration of the OSEM (only in the first iteration
> on an image with 1's) and used to normalize the measured sinogram
> before adapting the tomogram. The resulting tomogram has some
> artifacts because of it.
>
> Thanks in advance,
>
> Wout
>
> pro test
> ; Detector
> N=80
> projcen=(N-1)/2.
>
> ; Angles

> anglestart=0.
> anglerange=180.
> NAngles=anglerange/2.
>
> angleinc=anglerange/(NAngles-1)
> angles=anglestart+angleinc*findgen(NAngles)
> angles*=!pi/180
>
> ; Reconstructing an object with 1's
> tomogram=replicate(1.,N,N)
> estsinogram=radon(tomogram,theta=angles,xmin=-projcen,RMIN=-
projcen,drho=1,NRHO=N,/LINEAR)
>
> loadct,0
> window
> tvscl,not bytscl(rebin(estsinogram,NAngles*3,N*3,/sample))
> end

Hi

please compare your sinogram with the following:

estsinogram=radon(tomogram,rho=rho,theta=theta,ntheta=nangle s)

compare the rho and theta values and read the IDL radon help pages.

HTH

Peter

Subject: Re: Radon forward projection problem
Posted by Wox on Wed, 23 Apr 2008 10:02:37 GMT
View Forum Message <> Reply to Message

On Wed, 23 Apr 2008 00:42:07 -0700 (PDT), VP <peter.vontobel@psi.ch>
wrote:

> Hi
>
> please compare your sinogram with the following:
>
> estsinogram=radon(tomogram,rho=rho,theta=theta,ntheta=nangle s)
>
> compare the rho and theta values and read the IDL radon help pages.
>
> HTH
>

> Peter

So basically what you're saying is: don't undersample.

However, now there are more zero's in the sinogram. Let me explain what I want to do. The algorithm I was talking about (OSEM, although I think it's really called MLEM, I'm not sure) goes like this (BP=backprojection, FP=forward projection):

========Pseudo-code========
sino1=sinogram with all 1's
tomo1=BP(sino1)

tomo=tomogram with all 1's
for i=0,niter-1 do begin
 estsino=FP(tomo)
 tomo = tomo * BP(sino_measured/estsino)/tomo1
endfor
========================

You see that "estsino" and "tomo1" can't have zeroed pixels. Off course, I tried replacing the zeroed pixels by 1, max(rest), min(rest), etc... But this gives some artifacts in the resulting tomogram. Any ideas?

Subject: Re: Radon forward projection problem
Posted by Vontobel Peter on Wed, 23 Apr 2008 10:42:10 GMT
View Forum Message <> Reply to Message

On 23 Apr., 12:02, Wox <nom...@hotmail.com> wrote:
> On Wed, 23 Apr 2008 00:42:07 -0700 (PDT), VP <peter.vonto...@psi.ch>
> wrote:
>
>> Hi
>
>> please compare your sinogram with the following:
>
>> estsinogram=radon(tomogram,rho=rho,theta=theta,ntheta=nangle s)
>
>> compare the rho and theta values and read the IDL radon help pages.
>
>> HTH
>
>> Peter
>
> So basically what you're saying is: don't undersample.
>

> However, now there are more zero's in the sinogram. Let me explain
> what I want to do. The algorithm I was talking about (OSEM, although I
> think it's really called MLEM, I'm not sure) goes like this
> (BP=backprojection, FP=forward projection):
>
> ========Pseudo-code========
> sino1=sinogram with all 1's
> tomo1=BP(sino1)
>
> tomo=tomogram with all 1's
> for i=0,niter-1 do begin
> estsino=FP(tomo)
> tomo = tomo * BP(sino_measured/estsino)/tomo1
> endfor
> ========================
>
> You see that "estsino" and "tomo1" can't have zeroed pixels. Off
> course, I tried replacing the zeroed pixels by 1, max(rest),
> min(rest), etc... But this gives some artifacts in the resulting
> tomogram. Any ideas?

Hi

I cannot comment your effort to implement an iterative reconstruction algorithm. Simply comparing the two sinograms, i claim that your estsinogram is not the full sinogram of a square. You first have to make sure to start with the sinogram of a square !

HTH

Peter

Subject: Re: Radon forward projection problem
Posted by mmiller3 on Wed, 23 Apr 2008 16:53:05 GMT
View Forum Message <> Reply to Message

>>>> > "Wox" == Wox <nomail@hotmail.com> writes:

> The algorithm I was talking about (OSEM, although I think
> it's really called MLEM, I'm not sure) goes like this
> (BP=backprojection, FP=forward projection):

Be careful that you are sure of what you are implementing. OSEM and MLEM differ - especially in computational speed. The citations that I've got for the algorithms are:

"Accelerated Image Reconstruction Using Ordered Subsets of

Projection Data," IEEE Trans Med Img, 13, 601-609, 1994.

"Maximum likelihood reconstruction for emission tomography,"
IEEE Trans Med Img, MI-2, 113-122, 1982

Mike

--
Michael A. Miller mmiller3@iupui.edu
Imaging Sciences, Department of Radiology, IU School of Medicine

---

Subject: Re: Radon forward projection problem
Posted by Wox on Thu, 24 Apr 2008 11:14:57 GMT
View Forum Message <> Reply to Message

On Wed, 23 Apr 2008 12:53:05 -0400, mmiller3@iupui.edu (Michael A.
Miller) wrote:

>>>> >> "Wox" == Wox <nomail@hotmail.com> writes:
>
>> The algorithm I was talking about (OSEM, although I think
>> it's really called MLEM, I'm not sure) goes like this
>> (BP=backprojection, FP=forward projection):
>
> Be careful that you are sure of what you are implementing. OSEM
> and MLEM differ - especially in computational speed. The
> citations that I've got for the algorithms are:
>
> "Accelerated Image Reconstruction Using Ordered Subsets of
> Projection Data," IEEE Trans Med Img, 13, 601-609, 1994.
>
> "Maximum likelihood reconstruction for emission tomography,"
> IEEE Trans Med Img, MI-2, 113-122, 1982
>
>
> Mike

I have been reading those articles, but I couldn't understand the
difference. I would really appreciate a professional opinion on this.
As far as I understand, for each iteration in MLEM:
v'=v * BP(s0/FP(v))/BP(s1)

s0: experimental sinogram
s1: sinogram with all 1's
v: tomogram of previous iteration (this is a uniform image with 1's
for the first iteration)

v': the tomogram calculated for this iteration
BP: backprojection
FP: forward projection

Now what is OSEM doing?

Btw, is it correct that SIRT is doing this:
v'=v - b*BP(s0-FP(v))
where b a relaxation factor.

Subject: Re: Radon forward projection problem
Posted by mmiller3 on Thu, 24 Apr 2008 16:22:44 GMT
View Forum Message <> Reply to Message

>>>> > "Wox" == Wox <nomail@hotmail.com> writes:

> Now what is OSEM doing?

OSEM uses a different subset of the data for each iteration. For example, if you were running with 8 subsets, you'd use data from angles 0, 7, 15, ... for the first iteration, the data from angles 1, 8, 16, ... for the second iteration, 2, 9, 17, ... for the third and so on in order - hence the name ordered subsets EM. Each subset is handled using regular EM. Note that each subset must be a reasonably complete measurement by itself. If too many subset are used, the signal-to-noise in each subset will approach zero and the method won't do any thing useful.

OSEM has the advantage of making each iteration take less time then using the full data set each time, so it is computationally feasible compared to EM. I noted that in your original posting, you were asking about interpolating your sinograms before back projecting. If you interpolate subsets to fill in larger sinograms, you will loose some of the speed advantages of OSEM.

When run to "convergence," the OSEM result will sort of oscillate between the subsets. Usually OSEM is stopped before that happens, but the stopping point has to be determined empirically - typically when the images look "good." Since the algorithm hasn't fully converged at that point, there are issues with using OSEM images for quantitative work. The same issues arise with EM, since it will converge on the data, including the noise. Since OSEM uses subsets, it will not even converge on the same result as EM, if both where run to convergence.

In the medical world, OSEM is very commonly used for diagnostic work, though. OSEM takes longer than filtered back projection,

but produces nicer/smoother images far more quickly than
quantitatively accurate methods such as MAP.

Mike

P.S. Note the use of 0-based indices for choosing angles - still
trying to stay on topic for IDL ;-)

--
Michael A. Miller mmiller3@iupui.edu
Imaging Sciences, Department of Radiology, IU School of Medicine

---

Subject: Re: Radon forward projection problem
Posted by Wox on Sun, 27 Apr 2008 11:16:58 GMT
View Forum Message <> Reply to Message

On Thu, 24 Apr 2008 12:22:44 -0400, mmiller3@iupui.edu (Michael A.
Miller) wrote:

>>>> >> "Wox" == Wox <nomail@hotmail.com> writes:
>
>> Now what is OSEM doing?
>
> OSEM uses a different subset of the data for each iteration. For
> example, if you were running with 8 subsets, you'd use data from
> angles 0, 7, 15, ... for the first iteration, the data from
> angles 1, 8, 16, ... for the second iteration, 2, 9, 17, ... for
> the third and so on in order - hence the name ordered subsets EM.
> Each subset is handled using regular EM. Note that each subset
> must be a reasonably complete measurement by itself. If too many
> subset are used, the signal-to-noise in each subset will approach
> zero and the method won't do any thing useful.

Ah, so I'm using MLEM :-). However, my initial problem still stands.
What should one do when FP(v) is zero in some pixels, that is in
formula
v'=v * BP(s0/FP(v))/BP(s1)

Subject: Re: Radon forward projection problem
Posted by mmiller3 on Mon, 28 Apr 2008 13:33:22 GMT
View Forum Message <> Reply to Message

>>>> > "Wox" == Wox <nomail@hotmail.com> writes:

> Ah, so I'm using MLEM :-). However, my initial problem

> still stands. What should one do when FP(v) is zero in
> some pixels, that is in formula

> v'=v * BP(s0/FP(v))/BP(s1)

This is a common problem in numerical codes. When dividing, I usually pick a small threshold value, epsilon, and ensure that my value is greater than epsilon before dividing:

```
IDL> x=findgen(50)
IDL> y = x
IDL> epsilon = 10
IDL> y[where(x le epsilon)] = epsilon
IDL> plot, 1/x
% Program caused arithmetic error: Floating divide by 0
IDL> plot, 1/y
```

Or something along those lines.

Mike

--
Michael A. Miller mmiller3@iupui.edu
Imaging Sciences, Department of Radiology, IU School of Medicine