Subject: Convolution, IDL & Numerical Recipes
Posted by aceves on Thu, 31 Oct 2002 20:50:29 GMT
View Forum Message <> Reply to Message

Hello..

I am using IDL for some of my research and have a particular problem with convolution of two arrays. I have used IDL's CONVOL procedure and subroutine CONVLV given in NUMERICAL RECEIPES..both give different results. I hope some one can shed light on what the reason might be.

Thank you. Hector

****
Problem:
****

At the IDL prompt I entered and obtained:

```
IDL> a=[0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0]  ;  signal!
IDL> k=[1,0,0,0,0,0,0,0,0]  ; kernel!
IDL> z=convol(a,k)
IDL> print, z
      0       0       0       0       0       0       0       0
      0       1       1       1       0       0       0       0
```

With Numerical Receipes (Example Book in Fortran, Program XCONVLV, Chap.13)

The Signal file, or Data is (ibin,DATA)

```
 1  0.
 2  0.
 3  0.
 4  0.
 5  0.
 6  1.
 7  1.
 8  1.
 9  1.
 10  1.
 11  0.
 12  0.
 13  0.
 14  0.
```

15 0.
16 0.

The Kernel or Response Function is (jbin,RESPNS=KERNEL) ..an identity filter

1 1.
2 0.
3 0.
4 0.
5 0.
6 0.
7 0.
8 0.
9 0.

The Fortran program gives (ibin,Convolution,Expected value):

I	CONVLV	Expected
1	0.000000	0.000000
2	0.000000	0.000000
3	0.000000	0.000000
4	0.000000	0.000000
5	0.000000	0.000000
6	1.000000	1.000000
7	1.000000	1.000000
8	1.000000	1.000000
9	1.000000	1.000000
10	1.000000	1.000000
11	0.000000	0.000000
12	0.000000	0.000000
13	0.000000	0.000000
14	0.000000	0.000000
15	0.000000	0.000000
16	0.000000	0.000000

As shown, the results given by the numerical subroutine from NR
gives the expected results and differ from the one by IDL's CONVOL.

Subject: Re: Convolution, IDL & Numerical Recipes
Posted by R.G. Stockwell on Fri, 01 Nov 2002 18:44:51 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> R.G. Stockwell (sorry@noemail.now) writes:
>

>
>> Perhaps you want to use the following keywords:
>> Check out the help file to see the effects the keywords
>> have on how the arrays line up to be convolved.
>> (Note: you must explicitly set center=0, or else it defaults
>> to 1)
>
>
> Alright, now, can you give me the layman's definition
> of the difference between spacial filtering (CENTER=1)
> and convolution "in the strict mathematical sense"
> (CENTER=0).
...

> Cheers,
>
> David

If I may answer quickly off the top of my head without thinking about
it or looking at the help files, then I'd say, .. uh.... hmmm.... oh I
better look it up.

Ok, convol is just about the most messed up piece of code that IDL has.
(Don't get me started about people using the letter "l" as a variable,
which to me is indistinguishable from the number "1").

The difference is quite profound between the two.

IF center = 0 EXPLICITLY, then
you have the sum of A[t-m/2+i] K[i]
NOTE that the index of A is a constant +i, this is a correlation.
The kernel shifts along, and the time series shifts along in the same direction.

IF center = 1 OR is ommitted, then
you have the sum of A[t-i] K[i]
NOTE that the index of A is now a constant - i, this is a convolution.
The kernel shifts along, BUT the time series is shifting backwards (in the opposite direction).

Also, the offsets move around too.

> Which would I use if I'm trying to make a pretty image? :-)

I suggest running all possible permutations of the keywords, and
selecting the one that matches the textbook examples :)

```
; test code

a = indgen(10)
k = [1,2,3]

print,'____________________'
print,strcompress(string(convol(a,k,center=1)))
print
print,strcompress(string(convol(a,k,center=0)))

<results>

  0  8  14  20  26  32  38  44  50  0

  0  0  4  10  16  22  28  34  40  46
```

Note the way the results are not even similar! YAY!

Subject: Re: Convolution, IDL & Numerical Recipes
Posted by David Fanning on Fri, 01 Nov 2002 19:12:28 GMT
View Forum Message <> Reply to Message

R.G. Stockwell (sorry@noemail.now) writes:

>> Which would I use if I'm trying to make a pretty image? :-)
>
> I suggest running all possible permutations of the keywords, and
> selecting the one that matches the textbook examples :)

Right. That's what I'm doing now, but those SOBs
are tricky! I get the same result, but only after
a tiny histogram stretch that they fail to mention
to make the final image have the same gray-scale
range as the original. Fortunately, I've written
a book too, so the ol' enhance-the-image-so-it-looks-
better-in-the-book trick is not new to me. :-)

Cheers,

David

P.S. Let's just say that my original plan to write
a nice, short image processing recipe book for IDL
users looks like a harder project than I imagined.

But then, again, I could have started with HISTOGRAM
instead of CONVOL. :-)

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Convolution, IDL & Numerical Recipes
Posted by David Fanning on Fri, 01 Nov 2002 20:14:47 GMT
View Forum Message <> Reply to Message

R.G. Stockwell (sorry@noemail.now) writes:

>> Which would I use if I'm trying to make a pretty image? :-)
>
> I suggest running all possible permutations of the keywords, and
> selecting the one that matches the textbook examples :)

Well, I can assure you that if you want to sharpen an image,
that setting the CENTER keyword correctly is critical. :-)

Here are some images and the code I used to make them,
so you can see for yourselves. You definitely want a
convolution, not a correlation, which is just the opposite
of what MatLab wants, or.... Oh, hell, I'm confused again.
Where are those aspirin!?

http://www.dfanning.com/test/center_1.jpg
http://www.dfanning.com/test/center_1.jpg
http://www.dfanning.com/test/sharpen.pro

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Convolution, IDL & Numerical Recipes

Posted by R.G. Stockwell on Tue, 05 Nov 2002 13:34:42 GMT
View Forum Message <> Reply to Message

Hector Aceves wrote:
> "R.G. Stockwell" <sorry@noemail.now> wrote in message
news:<3DC28954.7060605@noemail.now>...
>
>> Perhaps you want to use the following keywords:
>> Check out the help file to see the effects the keywords
>> have on how the arrays line up to be convolved.
>> (Note: you must explicitly set center=0, or else it defaults
>> to 1)
>>
>> z=convol(a,k,center=0,edge_wrap=1)
>>
>> a 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0
>> k 1 0 0 0 0 0 0 0 0
>>
>> z 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0
>>
>>
>> Cheers,
>> bob stockwell
>
>
> Dear Bob...
>
> It works well with the kernel [1,0,...]
> But when I tried the actual examples of Numerical Recipes it did not
> give me the same results:
>
> a=[0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0]
> k=[0,0,1,1,1,1,0,0,0]
>
> z=convol(a,k,center=0,edge_wrap=1)
> IDL> print,z
> 0 0 0 0 0 0 0 1 2
> 3 4 4 3 2 1 0
> IDL>
>
> With Numerical Recipes gives..
>
> 0 1 1 1 1 1 0 1 2 3 3 3 2 1 0 0
>
> which seems ok!

If by "ok" you mean "completely wrong" then I agree with you. :)

Correllating two "boxcars" gives you a "triangle".

Perhaps you typed in the wrong "k" in your numrec code?

a=[0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0]
k=[1,1,1,0,0,0,0,0,1]

z=convol(a,k,center=1,edge_wrap=0,edge_trunc=1)

0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0
1 1 1 0 0 0 0 0 1

0 1 1 1 1 1 0 1 2 3 3 3 2 1 0 0

Also, keep in mind, as J.D. mentioned, that IDL convol is
a correlation with center=0, and a convolution with center = 1
(among other things).

You'd probably be better off to write your own 10 line piece of
code to perform the exact operation you want.
Actually, I might even do that, but I have a lot of other work to do,
so it's gonna be a while.

I'd use an fft to do it, and if you want no edge wrap, just zeropad.

Cheers,
bob

Subject: Re: Convolution, IDL & Numerical Recipes
Posted by JD Smith on Tue, 05 Nov 2002 22:42:59 GMT
View Forum Message <> Reply to Message

On Tue, 05 Nov 2002 06:34:42 -0700, R.G. Stockwell wrote:

> Hector Aceves wrote:
>> "R.G. Stockwell" <sorry@noemail.now> wrote in message
>> news:<3DC28954.7060605@noemail.now>...
>>
>>> Perhaps you want to use the following keywords: Check out the help file
>>> to see the effects the keywords have on how the arrays line up to be
>>> convolved. (Note: you must explicitly set center=0, or else it defaults
>>> to 1)
>>>
>>> z=convol(a,k,center=0,edge_wrap=1)
>>>
>>> a 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 k 1 0 0 0 0 0 0
>>> 0 0

>>>
>>> z 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0
>>>
>>>
>>> Cheers,
>>> bob stockwell
>>
>>
>> Dear Bob...
>>
>> It works well with the kernel [1,0,...] But when I tried the actual
>> examples of Numerical Recipes it did not give me the same results:
>>
>>   a=[0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0]
>>   k=[0,0,1,1,1,1,0,0,0]
>>
>>   z=convol(a,k,center=0,edge_wrap=1)
>> IDL> print,z
>>   0   0   0   0   0   0   0   1 2 3
>>   4   4   3   2   1   0
>> IDL>
>>
>> With Numerical Recipes gives..
>>
>>   0 1 1 1 1 1 0 1 2 3 3 3 2 1 0 0
>>
>> which seems ok!
>
> If by "ok" you mean "completely wrong" then I agree with you. :)
>
> Correllating two "boxcars" gives you a "triangle". Perhaps you typed in
> the wrong "k" in your numrec code?
>
> a=[0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0]
> k=[1,1,1,0,0,0,0,0,1]
>
> z=convol(a,k,center=1,edge_wrap=0,edge_trunc=1)
>
> 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1
>
> 0 1 1 1 1 1 0 1 2 3 3 3 2 1 0 0
>
> Also, keep in mind, as J.D. mentioned, that IDL convol is a correlation
> with center=0, and a convolution with center = 1 (among other things).
>
> You'd probably be better off to write your own 10 line piece of code to
> perform the exact operation you want. Actually, I might even do that,
> but I have a lot of other work to do, so it's gonna be a while.

>
> I'd use an fft to do it, and if you want no edge wrap, just zeropad.

Have a look at the NASA-library's CONVOLVE, which explicitly takes all
these IDL-native "features" into account, uses FFT when appropriate, and
may save you the trouble of writing one yourself.

JD

Subject: Re: Convolution, IDL & Numerical Recipes
Posted by aceves on Thu, 07 Nov 2002 01:02:59 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@as.arizona.edu> wrote in message
news:<pan.2002.11.05.22.42.57.734458.26650@as.arizona.edu>...
> On Tue, 05 Nov 2002 06:34:42 -0700, R.G. Stockwell wrote:
>
>> Hector Aceves wrote:
>>> "R.G. Stockwell" <sorry@noemail.now> wrote in message
>>> news:<3DC28954.7060605@noemail.now>...
>>>
>>>> Perhaps you want to use the following keywords: Check out the help file
>>>> to see the effects the keywords have on how the arrays line up to be
>>>> convolved. (Note: you must explicitly set center=0, or else it defaults
>>>> to 1)
>>>>
>>>> z=convol(a,k,center=0,edge_wrap=1)
>>>>
>>>> a 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 k 1 0 0 0 0 0 0
>>>> 0 0
>>>>
>>>> z 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0
>>>>
>>>>
>>>> Cheers,
>>>> bob stockwell
>>>
>>>
>>> Dear Bob...
>>>
>>> It works well with the kernel [1,0,...] But when I tried the actual
>>> examples of Numerical Recipes it did not give me the same results:
>>>
>>> a=[0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0]
>>> k=[0,0,1,1,1,1,0,0,0]
>>>
>>> z=convol(a,k,center=0,edge_wrap=1)

>>> IDL> print,z
>>> 0 0 0 0 0 0 0 1 2 3
>>> 4 4 3 2 1 0
>>> IDL>
>>>
>>> With Numerical Recipes gives..
>>>
>>> 0 1 1 1 1 1 0 1 2 3 3 3 2 1 0 0
>>>
>>> which seems ok!
>>
>> If by "ok" you mean "completely wrong" then I agree with you. :)
>>
>> Correllating two "boxcars" gives you a "triangle". Perhaps you typed in
>> the wrong "k" in your numrec code?
>>
>> a=[0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0]
>> k=[1,1,1,0,0,0,0,0,1]
>>
>> z=convol(a,k,center=1,edge_wrap=0,edge_trunc=1)
>>
>> 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1
>>
>> 0 1 1 1 1 1 0 1 2 3 3 3 2 1 0 0
>>
>> Also, keep in mind, as J.D. mentioned, that IDL convol is a correlation
>> with center=0, and a convolution with center = 1 (among other things).
>>
>> You'd probably be better off to write your own 10 line piece of code to
>> perform the exact operation you want. Actually, I might even do that,
>> but I have a lot of other work to do, so it's gonna be a while.
>>
>> I'd use an fft to do it, and if you want no edge wrap, just zeropad.
>
> Have a look at the NASA-library's CONVOLVE, which explicitly takes all
> these IDL-native "features" into account, uses FFT when appropriate, and
> may save you the trouble of writing one yourself.
>
> JD

Thanks to everyone... I think I got it now.

Hector

Subject: Re: Convolution, IDL & Numerical Recipes
Posted by muzic on Tue, 12 Nov 2002 14:10:32 GMT

The definition of discrete convolution is
c[n] = sum_over_i (a[i] * b[n-i])
where c equals a convolved with b.

Correlation differs in that sign inside the [] of b
d[n] = sum_over_i (a[i] * b[n+i])
where d equals correlation of a with b

In both cases, sum_over_i can mean sum as i goes
from negative infinity to positive infinity. Since
computers have finite speed and memory, this does
not make implementation practical ... unless, of
course, a and b have only a finite number of non-zero
values so that multiplication and summing need
only be done on non-zeros.

In a IDL, C, MATLAB or whatever implementation,
the index of the first element of an array is typically
zero or one but not negative. This means how the results are stored
in an array representing c[] or d[] may require an offset since
someone might like to calculate c[-4]. If we want to
put c[-4] in the first element of an IDL array, then
one would have to apply an offset of four. E.g. let cc be a shifted
(or offset) version of c, so that cc[0] = c[-4] and in
general cc[n] = c[n+4].

Regarding, "center" and a sharpening filter kernel, ...
it is the same principle but the issue is the offset of a or b.
Let a be the siganl to filter and b be the coefficients of a
filter kernel. Then, center should(!) is an offset for
the indexing on b. If you follow the details of the
math, changing the center should(!) simply shift the
result. If IDL implementtion does not behave in this
way, (or at least in a manner consistent with its
documentation) I'd say it is a bug.

Ray Muzic
Associate Professor, Radiology and Biomedical Engineering
Case Western Reserve University

David Fanning <david@dfanning.com> wrote in message

news:<MPG.182c80089cb24eff9899fb@news.frii.com>...
> R.G. Stockwell (sorry@noemail.now) writes:
>
>>> Which would I use if I'm trying to make a pretty image? :-)
>>
>> I suggest running all possible permutations of the keywords, and
>> selecting the one that matches the textbook examples :)
>
> Right. That's what I'm doing now, but those SOBs
> are tricky! I get the same result, but only after
> a tiny histogram stretch that they fail to mention
> to make the final image have the same gray-scale
> range as the original. Fortunately, I've written
> a book too, so the ol' enhance-the-image-so-it-looks-
> better-in-the-book trick is not new to me. :-)
>
> Cheers,
>
> David
>
> P.S. Let's just say that my original plan to write
> a nice, short image processing recipe book for IDL
> users looks like a harder project than I imagined.
> But then, again, I could have started with HISTOGRAM
> instead of CONVOL. :-)

Subject: Re: Convolution, IDL & Numerical Recipes
Posted by muzic on Tue, 12 Nov 2002 14:10:33 GMT
View Forum Message <> Reply to Message

The definition of discrete convolution is
c[n] = sum_over_i (a[i] * b[n-i])
where c equals a convolved with b.

Correlation differs in that sign inside the [] of b
d[n] = sum_over_i (a[i] * b[n+i])
where d equals correlation of a with b

In both cases, sum_over_i can mean sum as i goes
from negative infinity to positive infinity. Since
computers have finite speed and memory, this does
not make implementation practical ... unless, of
course, a and b have only a finite number of non-zero
values so that multiplication and summing need
only be done on non-zeros.

In a IDL, C, MATLAB or whatever implementation,

the index of the first element of an array is typically
zero or one but not negative. This means how the results are stored
in an array representing c[] or d[] may require an offset since
someone might like to calculate c[-4]. If we want to
put c[-4] in the first element of an IDL array, then
one would have to apply an offset of four. E.g. let cc be a shifted
(or offset) version of c, so that cc[0] = c[-4] and in
general cc[n] = c[n+4].

Regarding, "center" and a sharpening filter kernel, ...
it is the same principle but the issue is the offset of a or b.
Let a be the siganl to filter and b be the coefficients of a
filter kernel. Then, center should(!) is an offset for
the indexing on b. If you follow the details of the
math, changing the center should(!) simply shift the
result. If IDL implementtion does not behave in this
way, (or at least in a manner consistent with its
documentation) I'd say it is a bug.

Ray Muzic
Associate Professor, Radiology and Biomedical Engineering
Case Western Reserve University

David Fanning <david@dfanning.com> wrote in message
news:<MPG.182c80089cb24eff9899fb@news.frii.com>...
> R.G. Stockwell (sorry@noemail.now) writes:
>
>>> Which would I use if I'm trying to make a pretty image? :-)
>>
>> I suggest running all possible permutations of the keywords, and
>> selecting the one that matches the textbook examples :)
>
> Right. That's what I'm doing now, but those SOBs
> are tricky! I get the same result, but only after
> a tiny histogram stretch that they fail to mention
> to make the final image have the same gray-scale
> range as the original. Fortunately, I've written
> a book too, so the ol' enhance-the-image-so-it-looks-
> better-in-the-book trick is not new to me. :-)
>
> Cheers,
>

> David
>
> P.S. Let's just say that my original plan to write
> a nice, short image processing recipe book for IDL
> users looks like a harder project than I imagined.
> But then, again, I could have started with HISTOGRAM
> instead of CONVOL. :-)
