Subject: Re: Challenging question - array curve fitting
Posted by Qing on Tue, 03 Apr 2007 13:31:44 GMT
View Forum Message <> Reply to Message

On Mar 30, 1:07 am, Craig Markwardt
<craigm...@REMOVEcow.physics.wisc.edu> wrote:
> "Qing" <c...@bigpond.net.au> writes:
>> G'day folks,
>
>> I have a time series of images represented by a 3D array as Data(Nx,
>> Ny, Nt) or Data (Nt, Nx, Ny). I would like to apply a non-linear curve
>> fitting to the time dimension for every pixel respectively. I can loop
>> through every pixel using 1-D curve fitting procedure, but the process
>> is slow and it does not make efficient use of multiple CPUs.
>
>> Theoretically I would think it should be feasible to perform curve
>> fitting for all pixels simultaneously via matrix operation? However,
>> all the IDL's fitting routines only accept vectors for input
>> parameters to my knowledge. Does anyone know if there is any non-
>> linear fitting routines that accept array parameters. Or can anyone
>> comment on whether such a routine is feasible at all?
>
> Greetings, there should be nothing stopping you from grouping multiple
> time series into a single large vector, and fitting them
> simultaneously. You just need to make your model function smart
> enough to know what to do with the concatenated data set.
>
> However, there is a point of diminishing returns. Since the number of
> arithmetic operations required to perform the fit scales as the number
> of pixels *cubed*, there is really no advantage to grouping large
> numbers of pixels together, in fact there may be a disadvantage. This
> depends on the number of times (your Nt), but since we don't know that
> number, you will have to find the right balance yourself.
>
> Good luck,
> Craig
>
> --
> -- --------------
> Craig B. Markwardt, Ph.D.   EMAIL: craigm...@REMOVEcow.physics.wisc.edu
> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
> -- --------------

Hello Craig,

Thanks a lot for your comments and tips. It is intriguing for
"grouping multiple
time series into a single large vector...". I can manage to transform/

reform the
data array into a large vector, but my brain just can't think of a way
to model
the concatenated vector independently. For example, I am using a
Gaussian curve
model with 3 fitting parameters for each curve. Typically Nx=Ny=128
and the
number of time points Nt=60. The thing is that my computer has two
CPUs, and it
only uses about 50% total CPU when fitting the curve by looping
through each pixel.
I though usually array operation is more efficient than looping throug
all elements individually, but I was not sure if that is the case for
a non-linear
fitting task. Or at least, using array operation can get better use of
the CPUs
upto 100%. Do you thing using a large vector would be as efficient as
using array?
Why does "the number of arithmetic operations required to perform the
fit scales
as the number of pixels *cubed*"?, I thought it would be a linear
relation if using
array just like looping through all pixels one-by-one. Am I missing
something?

I would really appreciate any further elaboration.

Puzzled from Downunder
Qing