Subject: inverse gradient
Posted by erano on Wed, 26 Nov 2008 10:10:40 GMT
View Forum Message <> Reply to Message

Hello all,

I'm looking for a IDL function doing inverse gradient for 2D array.
inputs should be :
2D array gradient for X
  2D array gradient for Y
output: the inverse gradient array
Can anyone help me?

Eran

Subject: Re: inverse gradient
Posted by Jeremy Bailin on Thu, 27 Nov 2008 14:24:24 GMT
View Forum Message <> Reply to Message

On Nov 27, 3:27 am, erano <eran.o...@gmail.com> wrote:
>> "inverse" meaning what for a vector-field?
>
>> Paolo
>
> "inverse" is the opposite operation for gradient.
> The inputs are 2D gradient images (dX and dY), where high values are
> large changes in the "inverse gradient" image, and zeros are stable
> (no changes) in the "inverse gradient".
>
> Eran

I would never use this in production code, but here's a hack that will
give you something to look at:

scalarfield = total(dX, /cumulative, 1) + total(dY, /cumulative, 2)

The real solution is to replace those totals with actual integrals.

-Jeremy.

Subject: Re: inverse gradient
Posted by erano on Fri, 28 Nov 2008 09:31:24 GMT
View Forum Message <> Reply to Message

On Nov 27, 5:24 pm, Jeremy Bailin <astroco...@gmail.com> wrote:

> On Nov 27, 3:27 am, erano <eran.o...@gmail.com> wrote:
>
>>> "inverse" meaning what for a vector-field?
>
>>> Paolo
>
>> "inverse" is the opposite operation for gradient.
>> The inputs are 2D gradient images (dX and dY), where high values are
>> large changes in the "inverse gradient" image, and zeros are stable
>> (no changes) in the "inverse gradient".
>
>> Eran
>
> I would never use this in production code, but here's a hack that will
> give you something to look at:
>
> scalarfield = total(dX, /cumulative, 1) + total(dY, /cumulative, 2)
>
> The real solution is to replace those totals with actual integrals.
>
> -Jeremy.

Thanks,

It's the simplest way. it's basicly works but with errors.
I found few articlies about "inverse gradient" on the web, and the problem is very complex and the simple MATLAB function (from matlab exchange) is expensive (memory).
The main problem is to solve A*F=V where A is M*N matrix, V is vector and we look for F.
I used LA_LEAST_SQUARES (with all possible methods) for solve this but it is slow and have memory problem. please note that most of A matrix are zeros...

Any ideas?

Subject: Re: inverse gradient
Posted by Jeremy Bailin on Fri, 28 Nov 2008 13:27:46 GMT
View Forum Message <> Reply to Message

On Nov 28, 4:31 am, erano <eran.o...@gmail.com> wrote:
> On Nov 27, 5:24 pm, Jeremy Bailin <astroco...@gmail.com> wrote:
>
>
>
>> On Nov 27, 3:27 am, erano <eran.o...@gmail.com> wrote:

>
>>>> "inverse" meaning what for a vector-field?
>
>>>> Paolo
>
>>> "inverse" is the opposite operation for gradient.
>>> The inputs are 2D gradient images (dX and dY), where high values are
>>> large changes in the "inverse gradient" image, and zeros are stable
>>> (no changes) in the "inverse gradient".
>
>>> Eran
>
>> I would never use this in production code, but here's a hack that will
>> give you something to look at:
>
>> scalarfield = total(dX, /cumulative, 1) + total(dY, /cumulative, 2)
>
>> The real solution is to replace those totals with actual integrals.
>
>> -Jeremy.
>
> Thanks,
>
> It's the simplest way. it's basicly works but with errors.
> I found few articlies about "inverse gradient" on the web, and the
> problem is very complex and the simple MATLAB function (from matlab
> exchange) is expensive (memory).
> The main problem is to solve A*F=V where A is M*N matrix, V is vector
> and we look for F.
> I used LA_LEAST_SQUARES (with all possible methods) for solve this
> but it is slow and have memory problem. please note that most of A
> matrix are zeros...
>
> Any ideas?

Oddly enough, that's the second time sparse arrays have come up in one
week!

You want LINBCG, which takes as input a sparse matrix created using
SPRSIN. The help pages on them are pretty decent - give them a read.

-Jeremy.

Subject: Re: inverse gradient
Posted by pgrigis on Fri, 28 Nov 2008 16:10:30 GMT
View Forum Message <> Reply to Message

erano wrote:

>>
>>>> "inverse" meaning what for a vector-field?
>>
>>>> Paolo
>>
>>> "inverse" is the opposite operation for gradient.
>>> The inputs are 2D gradient images (dX and dY), where high values are
>>> large changes in the "inverse gradient" image, and zeros are stable
>>> (no changes) in the "inverse gradient".
>>
>>> Eran
>>
>> I would never use this in production code, but here's a hack that will
>> give you something to look at:
>>
>> scalarfield = total(dX, /cumulative, 1) + total(dY, /cumulative, 2)
>>
>> The real solution is to replace those totals with actual integrals.
>>
>> -Jeremy.
>
>
> Thanks,
>
> It's the simplest way. it's basicly works but with errors.
> I found few articlies about "inverse gradient" on the web, and the
> problem is very complex and the simple MATLAB function (from matlab
> exchange) is expensive (memory).
> The main problem is to solve A*F=V where A is M*N matrix, V is vector
> and we look for F.

While that is true in general, the gradient operatiom (which is a
linear
and therefore can be represented as a matrix operation) is a matrix
that
consist only of diagonal and one-row above diagonal elements.

Therefore, there is no need to solve the full matrix equation, but one
can
use simpler methods (i.e. in IDL trisol).

Paolo

> I used LA_LEAST_SQUARES (with all possible methods) for solve this
> but it is slow and have memory problem. please note that most of A
> matrix are zeros...
>
> Any ideas?

Subject: Re: inverse gradient
Posted by pgrigis on Fri, 28 Nov 2008 16:14:02 GMT
View Forum Message <> Reply to Message

If the original poster has a gradient image A and wants to compute
the image B whose gradient is A, then it seems odd to me that a
sparse approach would work.

Ciao,
Paolo

Jeremy Bailin wrote:

>>
>>
>>

>>
>>>> > "inverse" meaning what for a vector-field?
>>
>>>> > Paolo
>>
>>>> "inverse" is the opposite operation for gradient.
>>>> The inputs are 2D gradient images (dX and dY), where high values are
>>>> large changes in the "inverse gradient" image, and zeros are stable
>>>> (no changes) in the "inverse gradient".
>>
>>>> Eran
>>
>>> I would never use this in production code, but here's a hack that will
>>> give you something to look at:
>>
>>> scalarfield = total(dX, /cumulative, 1) + total(dY, /cumulative, 2)
>>
>>> The real solution is to replace those totals with actual integrals.
>>
>>> -Jeremy.
>>
>> Thanks,
>>

>> It's the simplest way. it's basicly works but with errors.
>> I found few articlies about "inverse gradient" on the web, and the
>> problem is very complex and the simple MATLAB function (from matlab
>> exchange) is expensive (memory).
>> The main problem is to solve A*F=V where A is M*N matrix, V is vector
>> and we look for F.

>> but it is slow and have memory problem. please note that most of A
>> matrix are zeros...
>>
>> Any ideas?
>
> Oddly enough, that's the second time sparse arrays have come up in one
> week!
>
> You want LINBCG, which takes as input a sparse matrix created using
> SPRSIN. The help pages on them are pretty decent - give them a read.
>
> -Jeremy.

Subject: Re: inverse gradient
Posted by pgrigis on Fri, 28 Nov 2008 16:18:53 GMT
View Forum Message <> Reply to Message

erano wrote:
>>
>> "inverse" meaning what for a vector-field?
>>
>> Paolo
>>
>
> "inverse" is the opposite operation for gradient.
> The inputs are 2D gradient images (dX and dY), where high values are
> large changes in the "inverse gradient" image, and zeros are stable
> (no changes) in the "inverse gradient".

I guess that I would have called it the "scalar potential" myself
instead of the "inverse gradient" ... but I see what you mean.

Ciao,
Paolo

>
>
> Eran

Subject: Re: inverse gradient
Posted by pgrigis on Fri, 28 Nov 2008 16:31:12 GMT
View Forum Message <> Reply to Message

Jeremy Bailin wrote:

>>> "inverse" meaning what for a vector-field?
>>
>>> Paolo
>>
>> "inverse" is the opposite operation for gradient.
>> The inputs are 2D gradient images (dX and dY), where high values are
>> large changes in the "inverse gradient" image, and zeros are stable
>> (no changes) in the "inverse gradient".
>>
>> Eran
>
> I would never use this in production code, but here's a hack that will
> give you something to look at:
>
> scalarfield = total(dX, /cumulative, 1) + total(dY, /cumulative, 2)

From the standpoint of discrete operations, that's perfectly fine.
As an approximation for the real potential, that depends on
how the gradient was measured I gues...

Paolo

>
> The real solution is to replace those totals with actual integrals.
>
> -Jeremy.

Subject: Re: inverse gradient
Posted by erano on Tue, 02 Dec 2008 09:27:32 GMT
View Forum Message <> Reply to Message

>
> Oddly enough, that's the second time sparse arrays have come up in one
> week!
>
> You want LINBCG, which takes as input asparsematrix created using
> SPRSIN. The help pages on them are pretty decent - give them a read.
>
> -Jeremy.

YES, we are at the right direction.

BUT my matrix iz M*N (where M=2*N). when I add zeros to make it M*M,
and thrn use the SPRSIN to make it sparse, the solution from LINBCG is
noot good.
When I work with very small array, using the LA_LEAST_SQUARES on the
original array give clean and good result.

Subject: Re: inverse gradient
Posted by pgrigis on Tue, 02 Dec 2008 14:45:52 GMT
View Forum Message <> Reply to Message

erano wrote:
>>
>> Oddly enough, that's the second time sparse arrays have come up in one
>> week!
>>
>> You want LINBCG, which takes as input asparsematrix created using
>> SPRSIN. The help pages on them are pretty decent - give them a read.
>>
>> -Jeremy.
>
> YES, we are at the right direction.
> BUT my matrix iz M*N (where M=2*N). when I add zeros to make it M*M,
> and thrn use the SPRSIN to make it sparse, the solution from LINBCG is
> noot good.

How much is M and N?

Paolo

> When I work with very small array, using the LA_LEAST_SQUARES on the
> original array give clean and good result.

Subject: Re: inverse gradient
Posted by erano on Tue, 02 Dec 2008 15:12:44 GMT
View Forum Message <> Reply to Message

On Dec 2, 4:45 pm, Paolo <pgri...@gmail.com> wrote:
> erano wrote:
>
>>> Oddly enough, that's the second time sparse arrays have come up in one
>>> week!
>
>>> You want LINBCG, which takes as input asparsematrix created using
>>> SPRSIN. The help pages on them are pretty decent - give them a read.
>

>>> -Jeremy.
>
>> YES, we are at the right direction.
>> BUT my matrix iz M*N (where M=2*N). when I add zeros to make it M*M,
>> and thrn use the SPRSIN to make it sparse, the solution from LINBCG is
>> noot good.
>
> How much is M and N?
>
> Paolo
>
>

For the large array: N is between 10,000 to 800,000 and M=N*2.
(actually, N is dim_x * dim_y of an image)

Thanks again

Eran

Subject: Re: inverse gradient
Posted by pgrigis on Tue, 02 Dec 2008 15:45:56 GMT
View Forum Message <> Reply to Message

erano wrote:

>> erano wrote:
>>
>>>> Oddly enough, that's the second time sparse arrays have come up in one
>>>> week!
>>
>>>> You want LINBCG, which takes as input asparsematrix created using
>>>> SPRSIN. The help pages on them are pretty decent - give them a read.
>>
>>>> -Jeremy.
>>
>>> YES, we are at the right direction.
>>> BUT my matrix iz M*N (where M=2*N). when I add zeros to make it M*M,
>>> and thrn use the SPRSIN to make it sparse, the solution from LINBCG is
>>> noot good.
>>
>> How much is M and N?
>>
>> Paolo
>>
>>
>

> For the large array: N is between 10,000 to 800,000 and M=N*2.
> (actually, N is dim_x * dim_y of an image)

My suggestion is to operate on each line of the array
separately and loop over lines and columns.

This way you only need to deal with dim_x by dim_y
arrays, a much simpler problem...

Ciao,
Paolo

>
> Thanks again
>
> Eran

Subject: Re: inverse gradient
Posted by erano on Tue, 02 Dec 2008 17:04:43 GMT
View Forum Message <> Reply to Message

> My suggestion is to operate on each line of the array
> separately and loop over lines and columns.
>
> This way you only need to deal with dim_x by dim_y
> arrays, a much simpler problem...
>
> Ciao,
> Paolo
>
>
This idea is not clear to me. Do you mean to work on "each line of the
array" from the orignal array (dim_x * dim_y) or from the M * N
matrix?

Eran

Eran

Subject: Re: inverse gradient
Posted by pgrigis on Tue, 02 Dec 2008 17:20:34 GMT
View Forum Message <> Reply to Message

erano wrote:
>> My suggestion is to operate on each line of the array

>> separately and loop over lines and columns.
>>
>> This way you only need to deal with dim_x by dim_y
>> arrays, a much simpler problem...
>>
>> Ciao,
>> Paolo
>>
>>
> This idea is not clear to me. Do you mean to work on "each line of the
> array" from the orignal array (dim_x * dim_y) or from the M * N
> matrix?

I meant:

From dx compute the first row of the array (one dim_x vector).
Use dy to go from the first to the second row.
and so on.

But I do realize that this approach is not the best one if you
have experimental data and the divergence of (dx,dy)
is not zero.

But you haven't really described how you got dX and dY and
what is the potential....

Ciao,
Paolo

>
> Eran
>
> Eran

Subject: Re: inverse gradient
Posted by erano on Tue, 02 Dec 2008 18:50:55 GMT
View Forum Message <> Reply to Message

>
> But you haven't really described how you got dX and dY and
> what is the potential....
>

Well, dX and dY are based on other parameters gradient. I based on

MATLAB code. In MATLAB, we can use "\" for doing Ax=Y: x=A\Y, where A
is M*N matrix. I can attach the code.

Eran

Subject: Re: inverse gradient
Posted by pgrigis on Tue, 02 Dec 2008 18:59:48 GMT
View Forum Message <> Reply to Message

erano wrote:
>>
>> But you haven't really described how you got dX and dY and
>> what is the potential....
>>
>
> Well, dX and dY are based on other parameters gradient. I based on
> MATLAB code. In MATLAB, we can use "\" for doing Ax=Y: x=A\Y, where A
> is M*N matrix. I can attach the code.

My question was what are dX and dY? What is the potential?

In IDL you can do x=A#invert(Y) that I guess is similar
to what matlab does (modulo transposition of the arrays).

But if A is large and sparse, then use the sparse methods suggested.

Paolo

>
> Eran

Subject: Re: inverse gradient
Posted by erano on Tue, 02 Dec 2008 19:42:37 GMT
View Forum Message <> Reply to Message

On Dec 2, 8:59 pm, Paolo <pgri...@gmail.com> wrote:
> erano wrote:
>
>>> But you haven't really described how you got dX and dY and
>>> what is the potential....
>
>> Well, dX and dY are based on other parameters gradient. I based on
>> MATLAB code. In MATLAB, we can use "\" for doing Ax=Y: x=A\Y, where A
>> is M*N matrix. I can attach the code.

>
> My question was what are dX and dY? What is the potential?
>
> In IDL you can do x=A#invert(Y) that I guess is similar
> to what matlab does (modulo transposition of the arrays).
>
> But if A is large and sparse, then use the sparse methods suggested.
>
> Paolo
>
>
>
>
>
>> Eran- Hide quoted text -
>
> - Show quoted text -

The potential ia not relevant. The dX and dY can be based on any 2D function. And yes, A is (very) large and sparse, but the sparse methods are only for N*N matrix...while A is M*N.

Eran

Subject: Re: inverse gradient
Posted by pgrigis on Tue, 02 Dec 2008 22:44:57 GMT
View Forum Message <> Reply to Message

On Dec 2, 2:42 pm, erano <eran.o...@gmail.com> wrote:
> On Dec 2, 8:59 pm, Paolo <pgri...@gmail.com> wrote:
>
>
>
>> erano wrote:
>
>>>> But you haven't really described how you got dX and dY and
>>>> what is the potential....
>
>>> Well, dX and dY are based on other parameters gradient. I based on
>>> MATLAB code. In MATLAB, we can use "\" for doing Ax=Y: x=A\Y, where A
>>> is M*N matrix. I can attach the code.
>
>> My question was what are dX and dY? What is the potential?
>
>> In IDL you can do x=A#invert(Y) that I guess is similar
>> to what matlab does (modulo transposition of the arrays).
>

>> But if A is large and sparse, then use the sparse methods suggested.
>
>> Paolo
>
>>> Eran- Hide quoted text -
>
>> - Show quoted text -
>
> The potential ia not relevant. The dX and dY can be based on any 2D
> function.

I am still not convinced that total wouldn't work in that case...
maybe increasing the sampling by a factor 2 or 3 if memory allows...
again it depends what your final goal is, of which we haven't a clue.

> And yes, A is (very) large and sparse, but the sparse
> methods are only for N*N matrix...while A is M*N.

OK, you are totally right that IDL sparse-array
function are somewhat... sparse ;-)

I think you can buy an add-on for this kind of things,
but I never used it.

Maybe somebody out there has written an SVD program that
accept non-square, sparse arrays?

Ciao,
Paolo

>
> Eran

Subject: Re: inverse gradient
Posted by erano on Tue, 09 Dec 2008 08:36:07 GMT
View Forum Message <> Reply to Message

Just a short update :
(I talked with IDL - ITT Technical Support)
For solve A*x=b
When I A is M*N array and i wish to make it sparse I must to add zeros
to make it M*M (if M>N).
If I wish to use LINBCG I must to have not zero values at diagonal.
for all the new elements the diag shoult be set to 1.0, and I must to
sort the old array rows (and the vector b as well) so the diag is
full. It's working.

Subject: Re: inverse gradient
Posted by erano on Wed, 24 Dec 2008 09:55:34 GMT
View Forum Message <> Reply to Message

On Dec 9, 10:36 am, erano <eran.o...@gmail.com> wrote:
> Just a short update :
> (I talked with IDL - ITT Technical Support)
> For solve A*x=b
> When I A is M*N array and i wish to make it sparse I must to add zeros
> to make it M*M (if M>N).
> If I wish to use LINBCG I must to have not zero values at diagonal.
> for all the new elements the diag shoult be set to 1.0, and I must to
> sort the old array rows (and the vector b as well) so the diag is
> full. It's working.

UPDATE: working only for small array, and not need to set to 1.0 the
diag