
Subject: Least squares fit of a model to a skeleton consisting out of 3D points.
Posted by [Johan](#) on Mon, 24 Nov 2008 13:33:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have the following problem to solve and was wondering whether the mpfit routines of Craig Markwardt will do the job?

Do have the following model:

Let $g(X,Y,Z)=1$ be a quadratic function in the coordinate system (O,Z,Y,Z) defined by the long, horizontal and vertical axes (ellipsoid). Write the equation of this quadratic function in matrix notation as follows:

$$g(X,Y,Z) = [X, Y, Z] * [[A1,A4,A5],[A4,A2,A6],[A5,A6,A3]] * [[X],[Y],[Z]] + [X, Y, Z] * [[A7],[A8],[A9]]$$

Need to fit this model to a 3D skeleton of N points by using least squares by calculating the coefficients A_i .

This is achieved by minimizing the total squared error between the exact position of the points (X_i, Y_i, Z_i) on the quadratic surface and their real position in the coordinate system (O, X, Y, Z). The minimizing is performed from the derivative of the equation below with respect to $A_1 \dots A_9$:

$$J(A_1 \dots A_9) = \sum_{i=0}^N (1 - (X_i, Y_i, Z_i))^2$$

This equation yields a linear system of nine equations in which the values of coefficients $A_1 \dots A_9$ are unknown.

Anyone that can help?
Johan Marais

Subject: Re: Least squares fit of a model to a skeleton consisting out of 3D points.
Posted by [Wout De Nolf](#) on Mon, 24 Nov 2008 16:22:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Mon, 24 Nov 2008 05:33:18 -0800 (PST), Johan <johan@jmarais.com> wrote:

> I have the following problem to solve and was wondering whether the
> mpfit routines of Craig Markwardt will do the job?

>

> Do have the following model:

> Let $g(X,Y,Z)=1$ be a quadratic function in the coordinate system
> (O,Z,Y,Z) defined by the long, horizontal and vertical axes
> (ellipsoid). Write the equation of this quadratic function in matrix

> notation as follows:

>

> $g(X,Y,Z) = [X, Y, Z] * [[A1,A4,A5],[A4,A2,A6],[A5,A6,A3]] * [[X],[Y],[Z]]$

> $+ [X, Y, Z] * [[A7],[A8],[A9]]$

>

> Need to fit this model to a 3D skeleton of N points by using least

> squares by calculating the coefficients A_i .

>

> This is achieved by minimizing the total squared error between the

> exact position of the points (X_i, Y_i, Z_i) on the quadratic surface and

> their real position in the coordinate system (O, X, Y, Z) . The

> minimizing is performed from the derivative of the equation below with

> respect to $A_1 \dots A_9$:

>

> $J(A_1 \dots A_9) = \sum_{i=0}^N (1/2) (X_i, Y_i, Z_i)^2$

>

> This equation yields a linear system of nine equations in which the

> values of coefficients $A_1 \dots A_9$ are unknown.

>

> Anyone that can help?

Search the IDL Code Library on the ITTVIS website for Ronn Kling's "Ellipse and Ellipsoid fitting routine" (krellipsoidfit.pro) I think this uses orthogonal distance regression, not non-linear least squares fitting.

If x, y and z are measured, orthogonal distance regression is the way to go. If you insist on using NLLSQ, you could use mpfit by setting $X=[X,Y,Z]$; (you need to extract the separate X, Y and Z in your user routine)

$Y=\text{replicate}(1, n_elements(X))$

assuming $g(x,y,z)=1$

I'm not sure whether that works (e.g. what are the weights for Y? I'd guess all 1...). Someone else will probably be able to tell you why to use ODR instead of NLLSQ.

Subject: Re: Least squares fit of a model to a skeleton consisting out of 3D points.

Posted by [Wout De Nolf](#) on Mon, 24 Nov 2008 16:35:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Mon, 24 Nov 2008 17:22:53 +0100, Wox <spam@nomail.com> wrote:

> $X=[X,Y,Z]$; (you need to extract the separate X, Y and Z in your user routine)

> $Y=\text{replicate}(1, n_elements(X))$

Woops, redefined X :-). I mean Y=replicate(1,n3Dpoints).

Subject: Re: Least squares fit of a model to a skeleton consisting out of 3D points.
Posted by [Johan](#) on Wed, 26 Nov 2008 08:40:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Nov 24, 4:35 pm, Wox <s...@nomail.com> wrote:

> On Mon, 24 Nov 2008 17:22:53 +0100, Wox <s...@nomail.com> wrote:

>> X=[X,Y,Z] ; (you need to extract the seperate X, Y and Z in your user
>> routine)

>> Y=replicate(1,n_elements(X))

>

> Woops, redefined X :-). I mean Y=replicate(1,n3Dpoints).

Thank you, it seems that krellipsoidfit.pro works rather well. I do have another question regarding this and will appreciate if can advise me.

I need to get the 3 angles and axis lengths and use the following code to get it from the given eigenvalues (evals) and eigenvectors (evec):

```
semia = sqrt(evals[0]) * 2.0
```

```
semib = sqrt(evals[1]) * 2.0
```

```
semic = sqrt(evals[2]) * 2.0
```

```
a = semia * 2.0
```

```
b = semib * 2.0
```

```
c = semic * 2.0
```

```
semiAxes = [semia, semib, semic]
```

```
axes = [a, b, c]
```

```
eigenvector = evec[* ,0]
```

```
eigenvector2 = evec[* ,1]
```

```
eigenvector3 = evec[* ,2]
```

```
orientation1 = atan(eigenvector1[1], eigenvector1[0])*!RADEG
```

```
orientation2 = atan(eigenvector2[1], eigenvector2[0])*!RADEG
```

```
orientation3 = atan(eigenvector3[1], eigenvector3[0])*!RADEG
```

```
angles = [orientation1, orientation2, orientation3]
```

Is this correct or do I need made some adjustments, especially to the orientation?

Thanks

Johan Marais

Subject: Re: Least squares fit of a model to a skeleton consisting out of 3D points.
Posted by [Wout De Nolf](#) on Wed, 26 Nov 2008 17:18:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 26 Nov 2008 00:40:38 -0800 (PST), Johan <johan@jmarais.com> wrote:

> Is this correct or do I need made some adjustments, especially to the
> orientation?

I don't know but use JLlipsoidFit to simulate your own ellipsoid
(use makeEllipsoid and rotate + scale + shift it) and check your
calculations.

Subject: Re: Least squares fit of a model to a skeleton consisting out of 3D points.
Posted by [Jeremy Bailin](#) on Thu, 27 Nov 2008 13:53:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Nov 26, 3:40 am, Johan <jo...@jmarais.com> wrote:

> On Nov 24, 4:35 pm, Wox <s...@nomail.com> wrote:

>

>> On Mon, 24 Nov 2008 17:22:53 +0100, Wox <s...@nomail.com> wrote:

>>> X=[X,Y,Z] ; (you need to extract the seperate X, Y and Z in your user
>>> routine)

>>> Y=replicate(1,n_elements(X))

>

>> Woops, redefined X :-). I mean Y=replicate(1,n3Dpoints).

>

> Thank you, it seems that krellipsoidfit.pro works rather well. I do
> have another question regarding this and will appreciate if can advise
> me.

>

> I need to get the 3 angles and axis lengths and use the following code
> to get it from the given eigenvalues (evals) and eigenvectors (evec):

>

> semia = sqrt(evals[0]) * 2.0

> semib = sqrt(evals[1]) * 2.0

> semic = sqrt(evals[2]) * 2.0

>

> a = semia * 2.0

> b = semib * 2.0

> c = semic * 2.0

> semiAxes = [semia, semib, semic]

> axes = [a, b, c]

>

> eigenvector = evec[* ,0]

> eigenvector2 = evec[* ,1]

```
> eigenvector3 = evec[* ,2]
>
> orientation1 = atan(eigenvector1[1], eigenvector1[0])*!RADEG
> orientation2 = atan(eigenvector2[1], eigenvector2[0])*!RADEG
> orientation3 = atan(eigenvector3[1], eigenvector3[0])*!RADEG
> angles = [orientation1, orientation2, orientation3]
>
> Is this correct or do I need made some adjustments, especially to the
> orientation?
>
> Thanks
> Johan Marais
```

That does indeed give you 3 angles, but it doesn't fully specify the orientation. Which angles are you looking for?

Incidentally, I'm not quite sure why you have that factor of 2 in the definition of semia etc., but I suppose it depends what went into the matrix you're diagonalizing...

-Jeremy.

Subject: Re: Least squares fit of a model to a skeleton consisting out of 3D points.
Posted by [Johan](#) on Tue, 02 Dec 2008 15:50:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
On Nov 27, 1:53 pm, Jeremy Bailin <astroco...@gmail.com> wrote:
> On Nov 26, 3:40 am, Johan <jo...@jmarais.com> wrote:
>
>
>
>
>
>> On Nov 24, 4:35 pm, Wox <s...@nomail.com> wrote:
>
>>> On Mon, 24 Nov 2008 17:22:53 +0100, Wox <s...@nomail.com> wrote:
>>>> X=[X,Y,Z] ; (you need to extract the seperate X, Y and Z in your user
>>>> routine)
>>>> Y=replicate(1,n_elements(X))
>
>>> Woops, redefined X :-). I mean Y=replicate(1,n3Dpoints).
>
>> Thank you, it seems that krellipsoidfit.pro works rather well. I do
>> have another question regarding this and will appreciate if can advise
>> me.
>
>> I need to get the 3 angles and axis lengths and use the following code
```

```

>> to get it from the given eigenvalues (evals) and eigenvectors (evec):
>
>> semia = sqrt(evals[0]) * 2.0
>> semib = sqrt(evals[1]) * 2.0
>> semic = sqrt(evals[2]) * 2.0
>
>> a = semia * 2.0
>> b = semib * 2.0
>> c = semic * 2.0
>> semiAxes = [semia, semib, semic]
>> axes = [a, b, c]
>
>> eigenvector = evec[:,0]
>> eigenvector2 = evec[:,1]
>> eigenvector3 = evec[:,2]
>
>> orientation1 = atan(eigenvector1[1], eigenvector1[0])*!RADEG
>> orientation2 = atan(eigenvector2[1], eigenvector2[0])*!RADEG
>> orientation3 = atan(eigenvector3[1], eigenvector3[0])*!RADEG
>> angles = [orientation1, orientation2, orientation3]
>
>> Is this correct or do I need made some adjustments, especially to the
>> orientation?
>
>> Thanks
>> Johan Marais
>
> That does indeed give you 3 angles, but it doesn't fully specify the
> orientation. Which angles are you looking for?
>
> Incidentally, I'm not quite sure why you have that factor of 2 in the
> definition of semia etc., but I suppose it depends what went into the
> matrix you're diagonalizing...
>
> -Jeremy.- Hide quoted text -
>
> - Show quoted text -

```

I tried different ways of getting the angles but it seems I am still at a lost. The angles I am looking for is as follow:
If you have an orthogonal reference framework and the ellipsoid are tilted in it. I am looking for the angles that the 3 axes of the ellipsoid make with the xy-plane, the yz-plane and yz-plane of the reference framework. I assume that for each of them you need to use all 3 relevant eigenvectors for each axes of the ellipsoid, or it could be only 2?

Subject: Re: Least squares fit of a model to a skeleton consisting out of 3D points.
Posted by [Jeremy Bailin](#) on Wed, 03 Dec 2008 14:14:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Dec 2, 10:50 am, Johan <jo...@jmarais.com> wrote:
> On Nov 27, 1:53 pm, Jeremy Bailin <astroco...@gmail.com> wrote:
>
>
>
>> On Nov 26, 3:40 am, Johan <jo...@jmarais.com> wrote:
>
>>> On Nov 24, 4:35 pm, Wox <s...@nomail.com> wrote:
>
>>>> On Mon, 24 Nov 2008 17:22:53 +0100, Wox <s...@nomail.com> wrote:
>>>> >X=[X,Y,Z] ; (you need to extract the seperate X, Y and Z in your user
>>>> >routine)
>>>> >Y=replicate(1,n_elements(X))
>
>>>> Woops, redefined X :-). I mean Y=replicate(1,n3Dpoints).
>
>>> Thank you, it seems that krellipsoidfit.pro works rather well. I do
>>> have another question regarding this and will appreciate if can advise
>>> me.
>
>>> I need to get the 3 angles and axis lengths and use the following code
>>> to get it from the given eigenvalues (evals) and eigenvectors (evec):
>
>>> semia = sqrt(evals[0]) * 2.0
>>> semib = sqrt(evals[1]) * 2.0
>>> semic = sqrt(evals[2]) * 2.0
>
>>> a = semia * 2.0
>>> b = semib * 2.0
>>> c = semic * 2.0
>>> semiAxes = [semia, semib, semic]
>>> axes = [a, b, c]
>
>>> eigenvector = evec[* ,0]
>>> eigenvector2 = evec[* ,1]
>>> eigenvector3 = evec[* ,2]
>
>>> orientation1 = atan(eigenvector1[1], eigenvector1[0])*!RADEG
>>> orientation2 = atan(eigenvector2[1], eigenvector2[0])*!RADEG
>>> orientation3 = atan(eigenvector3[1], eigenvector3[0])*!RADEG
>>> angles = [orientation1, orientation2, orientation3]
>
>>> Is this correct or do I need made some adjustments, especially to the
>>> orientation?
>

>>> Thanks
>>> Johan Marais
>
>> That does indeed give you 3 angles, but it doesn't fully specify the
>> orientation. Which angles are you looking for?
>
>> Incidentally, I'm not quite sure why you have that factor of 2 in the
>> definition of semia etc., but I suppose it depends what went into the
>> matrix you're diagonalizing...
>
>> -Jeremy.- Hide quoted text -
>
>> - Show quoted text -
>
> I tried different ways of getting the angles but it seems I am still
> at a lost. The angles I am looking for is as follow:
> If you have an orthogonal reference framework and the ellipsoid are
> tilted in it. I am looking for the angles that the 3 axes of the
> ellipsoid make with the xy-plane, the yz-plane and yz-plane of the
> reference framework. I assume that for each of them you need to use
> all 3 relevant eigenvectors for each axes of the ellipsoid, or it
> could be only 2?

That's 9 angles, so I'm still not quite sure what you mean. Maybe the Euler angles would be useful?

-Jeremy.

Subject: Re: Least squares fit of a model to a skeleton consisting out of 3D points.
Posted by [Johan](#) on Wed, 03 Dec 2008 14:37:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Dec 3, 2:14 pm, Jeremy Bailin <astroco...@gmail.com> wrote:

> On Dec 2, 10:50 am, Johan <jo...@jmarais.com> wrote:

>
>
>
>
>
>

>> On Nov 27, 1:53 pm, Jeremy Bailin <astroco...@gmail.com> wrote:

>

>>> On Nov 26, 3:40 am, Johan <jo...@jmarais.com> wrote:

>

>>>> On Nov 24, 4:35 pm, Wox <s...@nomail.com> wrote:

>

>>>> > On Mon, 24 Nov 2008 17:22:53 +0100, Wox <s...@nomail.com> wrote:

>>>> > >X=[X,Y,Z] ; (you need to extract the seperate X, Y and Z in your user

```

>>>> > >routine)
>>>> > >Y=replicate(1,n_elements(X))
>
>>>> > Woops, redefined X :-). I mean Y=replicate(1,n3Dpoints).
>
>>>> Thank you, it seems that krellipsoidfit.pro works rather well. I do
>>>> have another question regarding this and will appreciate if can advise
>>>> me.
>
>>>> I need to get the 3 angles and axis lengths and use the following code
>>>> to get it from the given eigenvalues (evals) and eigenvectors (evec):
>
>>>>     semia = sqrt(evals[0]) * 2.0
>>>>     semib = sqrt(evals[1]) * 2.0
>>>>     semic = sqrt(evals[2]) * 2.0
>
>>>>     a = semia * 2.0
>>>>     b = semib * 2.0
>>>>     c = semic * 2.0
>>>>     semiAxes = [semia, semib, semic]
>>>>     axes = [a, b, c]
>
>>>>     eigenvector = evec[* ,0]
>>>>     eigenvector2 = evec[* ,1]
>>>>     eigenvector3 = evec[* ,2]
>
>>>>     orientation1 = atan(eigenvector1[1], eigenvector1[0])*!RADEG
>>>>     orientation2 = atan(eigenvector2[1], eigenvector2[0])*!RADEG
>>>>     orientation3 = atan(eigenvector3[1], eigenvector3[0])*!RADEG
>>>>     angles = [orientation1, orientation2, orientation3]
>
>>>> Is this correct or do I need made some adjustments, especially to the
>>>> orientation?
>
>>>> Thanks
>>>> Johan Marais
>
>>> That does indeed give you 3 angles, but it doesn't fully specify the
>>> orientation. Which angles are you looking for?
>
>>> Incidentally, I'm not quite sure why you have that factor of 2 in the
>>> definition of semia etc., but I suppose it depends what went into the
>>> matrix you're diagonalizing...
>
>>> -Jeremy.- Hide quoted text -
>
>>> - Show quoted text -
>

```

>> I tried different ways of getting the angles but it seems I am still
>> at a lost. The angles I am looking for is as follow:
>> If you have an orthogonal reference framework and the ellipsoid are
>> tilted in it. I am looking for the angles that the 3 axes of the
>> ellipsoid make with the xy-plane, the yz-plane and yz-plane of the
>> reference framework. I assume that for each of them you need to use
>> all 3 relevant eigenvectors for each axes of the ellipsoid, or it
>> could be only 2?

>
> That's 9 angles, so I'm still not quite sure what you mean. Maybe the
> Euler angles would be useful?

>
> -Jeremy.- Hide quoted text -

>
> - Show quoted text -

Yes, I believe the Euler angles described the 3 I am after.

Johan

Subject: Re: Least squares fit of a model to a skeleton consisting out of 3D points.
Posted by [pgrigis](#) on Wed, 03 Dec 2008 14:37:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Johan wrote:

>>
>>
>>
>>
>>

>>
>>>> On Mon, 24 Nov 2008 17:22:53 +0100, Wox <s...@nomail.com> wrote:
>>>> >X=[X,Y,Z] ; (you need to extract the seperate X, Y and Z in your user
>>>> >routine)

>>>> >Y=replicate(1,n_elements(X))
>>
>>>> Woops, redefined X :-). I mean Y=replicate(1,n3Dpoints).

>>
>>> Thank you, it seems that krellipsoidfit.pro works rather well. I do
>>> have another question regarding this and will appreciate if can advise
>>> me.

>>
>>> I need to get the 3 angles and axis lengths and use the following code
>>> to get it from the given eigenvalues (evals) and eigenvectors (evec):

>>

>>

>>

>>

>>

>>> Is this correct or do I need made some adjustments, especially to the
>>> orientation?

>>

>>> Thanks

>>> Johan Marais

>>

>> That does indeed give you 3 angles, but it doesn't fully specify the
>> orientation. Which angles are you looking for?

>>

>> Incidentally, I'm not quite sure why you have that factor of 2 in the
>> definition of semia etc., but I suppose it depends what went into the
>> matrix you're diagonalizing...

>>

>> -Jeremy.- Hide quoted text -

>>

>> - Show quoted text -

>

> I tried different ways of getting the angles but it seems I am still
> at a lost. The angles I am looking for is as follow:
> If you have an orthogonal reference framework and the ellipsoid are
> tilted in it. I am looking for the angles that the 3 axes of the
> ellipsoid make with the xy-plane, the yz-plane and yz-plane of the

The angle between vectors a and b in IDL is given by
 $\arccos(\text{total}(a*b) / \sqrt{ \text{total}(a*a)*\text{total}(b*b) })$

Paolo

- > reference framework. I assume that for each of them you need to use
 - > all 3 relevant eigenvectors for each axes of the ellipsoid, or it
 - > could be only 2?
-