Subject: Re: Regression fit and random noise
Posted by Phillip Bitzer on Thu, 28 Mar 2013 21:51:36 GMT
View Forum Message <> Reply to Message

Not quite sure what you're asking here - we need a little more info. What routine are you using to do the fit? When you say noise/signal ratio, do you mean signal to noise ratio (SNR)? Do you have some sort of example data?

Regardless, consider the following "simple" linear regression, adapted from the IDL help:

```
PRO test_model

npts = 100
x = FINDGEN(npts)

noiseFactor = 0.

y = x + noiseFactor*(RANDOMU(seed, npts)-0.5)

plot, x, y, psym=2

result = REGRESS(X, Y, SIGMA=sigma, CONST=const, corr = r2, $
  MEASURE_ERRORS=measure_errors)

PRINT, 'Constant: ', const
PRINT, 'Coefficients: ', result[*]
PRINT, 'Standard errors: ', sigma
PRINT, 'Correl Coeff: ', r2

END
```

Notice as you increase the noise factor, the correlation coefficient gets worse. This is entirely expected and is not a IDL-only thing. Basically, the "signal" gets swamped out by the "noise". You should get your hands on a good statistics book (e.g., Data Reduction and Error Analysis in the Physical Sciences by Bevington, Statistical Methods in the Atmospheric Sciences by Wilks) to better interpet what's going on "under the hood". For instance, according to regress.pro, the fit is done via chi squared minimization.

Good luck!

Subject: Re: Regression fit and random noise
Posted by kisCA on Thu, 28 Mar 2013 22:26:01 GMT
View Forum Message <> Reply to Message

Thank you Phillip for your answer and the book reference !

What I am trying to know is if by increasing the noise ratio in my data, the model will still find a

good fit.

I increase the noise with:
new_sig = original_sig + noise_ratio*randomu(seed,n_elements(original_sig))

I increase slowly the value of noise_ratio. So first I obtain almost the same value of R2 as if no noise was there. R2 is getting lower as the value of noise_ratio increase. After a certain value of noise_ratio is reached, R2 values don't get lower than 0.3.

Hope it is clearer now

Subject: Re: Regression fit and random noise
Posted by Phillip Bitzer on Thu, 28 Mar 2013 23:18:50 GMT
View Forum Message <> Reply to Message

The (linear) correlation coefficient, r, is a measure how well the independent/dependent variables are correlated. For perfectly correlated data, r = 1, and the data plots as a straight line with positive slope. Perfectly anit-correlated data has a r=-1, and the data plots as a straight line with negative slope. Uncorrelated data has r=0; in this case, the best fit line has a slope of zero (imagine data points that are scattered with no perceptible trend). (You're dealing with the multiple correlation coefficient, but the concept is similar. There's a nice discussion in Bevington, among other places. BTW, the multiple correlation coefficient can be shown to be a linear combination of the linear correlation coefficients for each variable x_i. Further, the linear correlation coefficient can be used to assess the usefulness of a predictor in the model.)

In your case, setting noise ratio = 0 should provide the same value as if no noise was present because no (artificial) noise is present! As you increase noise_ratio, you're essentially "destroying" the correlation, in a manner of speaking. I bet if you crank up noise_ratio far enough you can get essentially uncorrelated data.

Be careful when you speak of a "good fit" - there ways to qualify what is a good fit (for example, using the chi squared value to test the null hypothesis). Depending on the SNR, the model will still be a "good fit" to the (noisy) data.

Ultimately, the answer to your question lies in the underlying statistics - there isn't (shouldn't be?) anything wonky going on in IDL.

Hope this helps!

Subject: Re: Regression fit and random noise
Posted by on Thu, 28 Mar 2013 23:26:30 GMT
View Forum Message <> Reply to Message

Den torsdagen den 28:e mars 2013 kl. 22:26:01 UTC skrev kisCA:
> Thank you Phillip for your answer and the book reference !

>
> What I am trying to know is if by increasing the noise ratio in my data, the model will still find a good fit.
>
> I increase the noise with:
>
> new_sig = original_sig + noise_ratio*randomu(seed,n_elements(original_sig))
>
> I increase slowly the value of noise_ratio. So first I obtain almost the same value of R2 as if no noise was there. R2 is getting lower as the value of noise_ratio increase. After a certain value of noise_ratio is reached, R2 values don't get lower than 0.3.

Maybe I misunderstand what you are trying to do but... Are you aware that randomu has a uniform distribution between 0 and 1? So you are adding on the average something like 0.5*noise_ratio to your original signal. So maybe you want to add noise_ratio*(randomu(...)-0.5) instead. Or, since randomn is normal distributed with zero mean, simply noise_ratio*randomn(...).

Subject: Re: Regression fit and random noise
Posted by kisCA on Thu, 28 Mar 2013 23:37:34 GMT
View Forum Message <> Reply to Message

Again thanks to both of you.

First, Mats, I tried both and the results are the same.

Phillip:
> "In your case, setting noise ratio = 0 should provide the same value as if no noise was present because no (artificial) noise is present! "
Yes it does !
> "I bet if you crank up noise_ratio far enough you can get essentially uncorrelated data. "
I understand the process of "destroying" the correlation. What I don't get is why does the coefficient of determination (R2) reach a plateau value (0.3) and doesn't get closer to zero as I raise the noise_ratio a lot (like a hundred)...
> "underlying statistics"
Is there any useful quantity I can calculate which would help me in this case ?

Again, thank you for your time and your explanations !

Subject: Re: Regression fit and random noise
Posted by Phillip Bitzer on Thu, 28 Mar 2013 23:40:49 GMT
View Forum Message <> Reply to Message

This is actually a good point, and likely explains your asymptotic value of the coefficient. Check out the sample code I posted to see an example of how to add uniform noise distributed about 0

----> noiseFactor*(RANDOMU(seed, npts)-0.5)

Depending on the data, and what you're trying to do, you could instead add Gaussian-distributed noise as mentioned. You'll probably want to modify the width of the Gaussian noise to *really* test the model as well.

See the help for more:
http://www.exelisvis.com/docs/RANDOMU.html
http://www.exelisvis.com/docs/RANDOMN.html

On Thursday, March 28, 2013 6:26:30 PM UTC-5, Mats Löfdahl wrote:
>
>
> Maybe I misunderstand what you are trying to do but... Are you aware that randomu has a uniform distribution between 0 and 1? So you are adding on the average something like 0.5*noise_ratio to your original signal. So maybe you want to add noise_ratio*(randomu(...)-0.5) instead. Or, since randomn is normal distributed with zero mean, simply noise_ratio*randomn(...).

Subject: Re: Regression fit and random noise
Posted by Craig Markwardt on Fri, 29 Mar 2013 01:16:03 GMT
View Forum Message <> Reply to Message

On Thursday, March 28, 2013 7:37:34 PM UTC-4, kisCA wrote:

> I understand the process of "destroying" the correlation. What I don't get is why does the coefficient of determination (R2) reach a plateau value (0.3) and doesn't get closer to zero as I raise the noise_ratio a lot (like a hundred)...
>
>> "underlying statistics"

The little cut and paste example below should show that the correlation factors do indeed converge to zero as the noise value is increased. Of course an individual sample of random scale factors may not make a R^2 value that goes to zero. R^2 itself has sample variance.

Craig

```
x = randomu(seed,100)  ;; Random X positions
ym = 0.3 - 0.7 * x     ;; Pure Y model (no noise)
ye = 0.01              ;; Initial scatter
;; Sampled y value
ys = ym + randomn(seed,100)*ye
print, r_correlate(x, ys)

;; NOISE_FACTOR multiples
noise_factors = [1, 10, 100, 1000, 10000]
```

```
;; Try different noise factors
for i = 0, n_elements(noise_factors)-1 do begin & ys1 = ym + randomn(seed,100)*ye *
noise_factors(i) & print, r_correlate(x, ys1)


;; EXAMPLE RUN:
;;       i   NOISE    SPEARMAN     SIGNIF
;;       0       1    -0.998248      0.00000
;;       1      10    -0.904374  5.05695e-38
;;       2     100    -0.175770    0.0802491
;;       3    1000     0.0576417     0.568917
;;       4   10000    -0.0107651     0.915328
```

Subject: Re: Regression fit and random noise
Posted by kisCA on Fri, 29 Mar 2013 17:24:32 GMT
View Forum Message <> Reply to Message

> Philip and Craig, thank you for your example.

I still don't understand the asymptotic value I reach. My point is, if you "drown" your signal in noise, even if it's between 0 and 1, the R^2 should tend to zero.

Craig, what do you mean by: "R^2 itself has sample variance."

Thank you again for your help, things start clearing up in my mind

---

Subject: Re: Regression fit and random noise
Posted by Craig Markwardt on Fri, 29 Mar 2013 23:39:00 GMT
View Forum Message <> Reply to Message

On Friday, March 29, 2013 1:24:32 PM UTC-4, kisCA wrote:
>> Philip and Craig, thank you for your example.
>
>
>
> I still don't understand the asymptotic value I reach. My point is, if you "drown" your signal in noise, even if it's between 0 and 1, the R^2 should tend to zero.

Yes, it does. I gave you an example.

> Craig, what do you mean by: "R^2 itself has sample variance."

Given a random sample of data, the computed R^2 value will not exactly equal its expected theoretical value. There is variance about the expected mean value. Only in the limit of averaging over many samples does it convert to the expectation value.