LMSD

Kiker, Thaddaeus <s024622@students.lmsd.org>

Data selection

9 messages

Steiner, James <james.steiner@cfa.harvard.edu> Tue, Jul 27, 2021 at 12:22 PM
To: "Kiker, Thaddaeus" <s024622@students.lmsd.org>

Hi Thaddaeus,

I wanted to check in with you on how things are going and to see what you are doing currently for the data quality / screening. Depending on where you are with getting the network running, we should make sure to tackle this.

-Jack

Kiker, Thaddaeus <s024622@students.lmsd.org> Tue, Jul 27, 2021 at 4:14 PM
To: "Steiner, James" <james.steiner@cfa.harvard.edu>

Hi Dr. Steiner!

Thanks for checking in!

I have been working the past couple days on a completely new processing routine which has a lot of cool features I'm looking forward to sharing with you! Could I send an in depth follow up email when I finish the fitting routine so I can share some analysis as well as describe the methods?

Thaddaeus

"Before I came here I was confused about this subject. Having listened to your lecture I am still confused. But on a higher level."
-Enrico Fermi

[Quoted text hidden]

Steiner, James <james.steiner@cfa.harvard.edu> Tue, Jul 27, 2021 at 4:28 PM
To: "Kiker, Thaddaeus" <s024622@students.lmsd.org>

Hi Thaddaeus,

Yes, of course! Looking forward to it.

-Jack

[Quoted text hidden]

Kiker, Thaddaeus <s024622@students.lmsd.org> Wed, Jul 28, 2021 at 2:44 PM
To: "Steiner, James" <james.steiner@cfa.harvard.edu>

Hi Dr. Steiner!

Okay! The routine was completed last night and I finalized all the wrangling routines today!

Short version:

Reprocessing has been completed...what quality checks do you think I should employ to cull values from the set I'll send to the neural network, and would you mind checking my hardness ratio routine?

Long version:

This routine was unique because I incorporated a lot of tcl code in it during XSPEC sessions, which streamlined a lot of things (e.g. only running error when the tcl code calculates that stat/dof is < 2) and made a lot of post processing / wrangling things after easier. Chief among these was the error results wrangling. In the past, I would log the entire error session to a log file, which made retrieving the final confidence intervals an intense ordeal. Now however, thanks to incorporation of tcl, the final error results and error tclout string are deposited into a pandas dataframe readable csv file for each observation's parameters that had confidence intervals calculated, which looks like this:

param_num,lower_bound,upper_bound,error_string
2,0,0.4258894025,FFFTFFTFF
3,55.18143562,5238.875993,FFFFFFFFF
4,1.683374841,2.159435738,FFFFFFFFF
9,0.006206526863,0.01517790938,FFFFFFFFF

There is a wealth of information ready to go for finalizing the sample based on this routine as I compiled all of these arrays into a dataframe:

```
dict = {
 'ids':np.array([]),
 'net_count_rates':np.array([]),
 'source_percents':np.array([]),
 'hardness_ratios':np.array([]),
 'mjds':np.array([]),
 'tins':np.array([]),
 'diskbb_norms':np.array([]),
 'gammas':np.array([]),
 'nthcomp_norms':np.array([]),
 'red_pgs':np.array([]),
 'tin_lowers':np.array([]),
 'tin_uppers':np.array([]),
 'tin_tclouts':np.array([]),
 'diskbb_norm_lowers':np.array([]),
 'diskbb_norm_uppers':np.array([]),
 'diskbb_norm_tclouts':np.array([]),
 'gamma_lowers':np.array([]),
 'gamma_uppers':np.array([]),
 'gamma_tclouts':np.array([]),
 'nthcomp_norm_lowers':np.array([]),
 'nthcomp_norm_uppers':np.array([]),
 'nthcomp_norm_tclouts':np.array([]),
 }
```

So the big question I wanted to get your advice on is how do you think I should "clean" the sample based on these parameters (e.g. source % cutoff, red pg. stat. range, whether or not all fitted parameters have confidence intervals ***not consistent*** with zero, etc.). On a side note, I noticed a lot of tin values got stuck at 0.2 and or had lower errors consistent with their hard limit / zero.

Also, would you mind taking a look over my hardness ratio calculating routine and letting me know if you see any errors in it?

```
# Get hardness ratio
 data_file = data_dir + '/' + obsid + '/jspipe/js_ni' + obsid + '_0mpu7_silver_GTI' + gti + '.jsgrp'
```

```
bg_file = data_file.replace('.jsgrp', '.bg')

orig_data_file = data_file
temp_data_file = data_file.replace('.fits', '(temp).fits')
temp_data_file = data_file.replace('.jsgrp', '(temp).fits')
shutil.copyfile(orig_data_file, temp_data_file)
data_hdul = fits.open(temp_data_file)

counts_array = np.array(data_hdul[1].data['COUNTS'])
exp_time = float(data_hdul[1].header['EXPOSURE'])
channels_array = np.array(data_hdul[1].data['CHANNEL'])

soft_mask = np.logical_and(channels_array>199, channels_array<400)
hard_mask = np.logical_and(channels_array>399, channels_array<1001)

hard_counts = np.sum(counts_array[hard_mask])
soft_counts = np.sum(counts_array[soft_mask])

mjd = float(data_hdul[1].header['MJDSTART'])
dict['mjds'] = np.append(dict['mjds'], mjd)

orig_bg = bg_file
temp_bg_file = bg_file.replace('.bg', '(temp).bg')
shutil.copyfile(orig_bg, temp_bg_file)
bg_hdul = fits.open(temp_bg_file)
bg_counts_array = bg_hdul[1].data['COUNTS']
bg_exp_time = float(bg_hdul[1].header['EXPOSURE'])

bg_hard_counts = np.sum(bg_counts_array[hard_mask])
bg_soft_counts = np.sum(bg_counts_array[soft_mask])

hardness_numerator = (hard_counts-(bg_hard_counts/bg_exp_time*exp_time))
hardness_denom = (soft_counts-(bg_soft_counts/bg_exp_time*exp_time))

hardness_ratio = hardness_numerator / hardness_denom

dict['hardness_ratios'] = np.append(dict['hardness_ratios'], hardness_ratio)

os.remove(temp_data_file)
os.remove(temp_bg_file)
```

Thanks so much!
Thaddaeus

"Before I came here I was confused about this subject. Having listened to your lecture I am still confused. But on a higher level."
-Enrico Fermi

[Quoted text hidden]

results.csv
112K

Steiner, James <james.steiner@cfa.harvard.edu> Thu, Jul 29, 2021 at 7:07 AM
To: "Kiker, Thaddaeus" <s024622@students.lmsd.org>

Hi Thaddaeus,

Looks great! Using the tclout features is a very handy way to go - nice work! Your hardness ratio looks algorithm looks fine to me with a very minor criticism that for consistency sake I would use either the inclusive range of channels 200-399,400-999 OR 201-400, 401-1000.

Only running errors for reduced stat <2 is okay *unless* this is missing the brightest data sets. If that's the case, then I suggest upping the "maxchi" threshold (or whatever it's called).

One additional item: I see a lot of the fits have pegged at Gamma=3. Soft states can go up to 4. I suggest putting in a soft upper limit of 3.5 and a hard limit of 4 to see what happens to those data sets.

>So the big question I wanted to get your advice on is how do you think I should "clean" the sample based on these parameters (e.g. source % cutoff, red pg. stat. range, whether or not all fitted parameters have confidence intervals ***not consistent*** with zero, etc.). On a side note, I noticed a lot of tin values got stuck at 0.2 and or had lower errors consistent with their hard limit / zero.

I think we want to have a few factors to use when assessing the full data in place. It looks like you have many of these handy but not all: (1) total background amplitude; (2) source / background ratio; (3) exposure time; (4) total counts; (5) reduced pgstat ; (6) parameters & their uncertainties.

Here's my baseline guidance on each:
(1) Let's eliminate anything with background count rate from 0.5-10 keV of >5 c/s.
(2) Can you please put together a scatter-plot of the ((source+background) / background) ratio versus either count-rate or flux and with logarithmic axes? Also, please remind me if you used a fixed energy range for the spectral fits or if you adjust depending on the background spectrum. (Probably restricting to observations with a ratio >~10 will be the right regime for a cut to cull the spectral fitting results.)
(3) We want to make sure the spectral data are matched to PDS; I believe this means t > 60s exposure-times (a header keyword of 'EXPOSURE' is a sufficiently close proxy to this - although it includes the small deadtime correction we discussed a while back).
(4) Anything with < ~2000 source counts won't produce a particularly useful spectral fit; those can be rejected. But since we also want good QPO data, a higher bar of 5,000 counts is a reasonable benchmark for being sensitive to ~1% QPO features. We also want to track total counts for the next item:
(5) If we had a perfect model and precise knowledge of all calibration uncertainties, the fit statistic would approach unity; unfortunately we have both an imperfect spectral model and imperfect knowledge of our detector. Both of those limitations are really only manifestly important when the data are brightest (often when they are also most interesting). It's useful to look at a scatterplot (log-scaling for counts) of the reduced fit statistic versus total source counts to get a sense for this landscape. This is in order to put in place a selection that allows the most signal-rich data in (i.e., not penalizing too harshly for cosmetic limitations of the calibration or model). To prevent a lot of additional confusion in the mix, how about you first apply the background screening so those rejected points aren't dominating the low-count-end of the distribution.

(6) Low-significance in terms of the departure from 0 can itself a very useful measure depending on the parameter in question. So by default I'd rather not screen based on parameter errorbars (unless the landscape of behavior shows obvious pathologies like those clearly bad temperatures when very faint in the first data set, or an error bar that is blown up an order of magnitude from everything else etc.). Instead I'd like to first implement the above screenings which are really about the data quality, but then have you make plots showing the evolution of these parameters over time (with their errors), and basic correlations versus flux and hardness, to see if anything stands out as an obvious issue.

As Mark Twain said, “If I had more time, I would have written a shorter letter.” I hope all is sufficiently clear although I know this note is a little jumbled.

Catch you at 2,

-Jack

[Quoted text hidden]

Kiker, Thaddaeus <s024622@students.lmsd.org> Thu, Jul 29, 2021 at 10:12 AM
To: "Steiner, James" <james.steiner@cfa.harvard.edu>

Hi Dr. Steiner!

Thanks so much for your email! This week was a little crazy so I'm a little behind from where I'd like to be, but I will definitely implement these checks in the coming days (next week should be a lot more regular).

See you soon!
[Quoted text hidden]
[Quoted text hidden]

Steiner, James <james.steiner@cfa.harvard.edu> Thu, Jul 29, 2021 at 10:20 AM
To: "Kiker, Thaddaeus" <s024622@students.lmsd.org>

Totally understood. That sounds great Thaddaeus; catch you shortly.

-Jack
[Quoted text hidden]

Kiker, Thaddaeus <s024622@students.lmsd.org> Tue, Aug 3, 2021 at 9:03 PM
To: "Steiner, James" <james.steiner@cfa.harvard.edu>

Hi!

> inclusive range of channels 200-399,400-999 OR 201-400, 401-1000

Thanks, I made this change!

> One additional item: I see a lot of the fits have pegged at Gamma=3. Soft states can go up to 4. I suggest putting in a soft upper limit of 3.5 and a hard limit of 4 to see what happens to those data sets.

Ah yes. I remember you mentioning this in our meeting and although I thought I put the hard limit at 3.5 the hard was at 3.0. I made this change, I will do a rerun based on this tonight.

1. This would just be sum(counts in the range)/bg exp time*spectrum exp time both deadtime adjusted?
2. Yes I will get this to you. I ignored **-0.5, 1.5-2.3, and 10.0-**. The middle ignore range was something we talked about a bit ago with the features with large residuals in that range I couldn't fit. I wonder if that's why tin was that informative because it didn't have that much data to get constrained on. Should I run another iteration of the fitting routine with that region noticed?
3. Ok. So should I stop fitting spectral ids corresponding to files with < 60 sec exposure?
4. Ok. I'll check how many files that brings us down to based on each criterion.
5. Ok I'll get this plot to you asap.
6. Sounds good. I think the best thing about how I worked with error this time around was that I logged the parameter values *after* running error --- after looking through the old code from December 2020 I realized that I was logging parameter values before running error and thus not receiving the "jump out of local minimums" or "new minimum found" benefit error provides.

Best!
[Quoted text hidden]
[Quoted text hidden]

Steiner, James <james.steiner@cfa.harvard.edu> Tue, Aug 3, 2021 at 9:41 PM
To: "Kiker, Thaddaeus" <s024622@students.lmsd.org>

Hi Thaddaeus,

Great, here's hoping the new run improves things a bit.

> 1. This would just be sum(counts in the range)/bg exp time*spectrum exp time both deadtime adjusted?

Here, the background count rate would be sum(bg counts from 0.5-10 keV) / bg exptime. (The exptime is in the header... no need to worry about whether or not deadtime is included, as the effect will be incredibly minor in this regime.)

> 2. Yes I will get this to you. I ignored **-0.5, 1.5-2.3, and 10.0-**. The middle ignore range was something we talked about a bit ago with the features with large residuals in that range I couldn't fit. I wonder if that's why tin was that informative because it didn't have that much data to get constrained on. Should I run another iteration of the fitting routine with that region noticed?

Hmm, interesting thought. There may be a few observations which get better constraints using the 1.5-2.3 keV data, but that should be a small minority and I think it's fine not to bother redoing with that. The option I had in mind (particularly if we need to bring in more faint data) is to identify the energy at which the background spectrum crosses over and exceeds the source spectrum, and to ignore data above that point. (The background if you do "setp back on" you'll see is quite flat whereas the source declines comparatively rapidly at high energy.. this crossover becomes important for faint or soft spectra as the data are less reliable when background is dominant and with a large systematic.)

> 3. Ok. So should I stop fitting spectral ids corresponding to files with < 60 sec exposure?

Yes.

> 4. Ok. I'll check how many files that brings us down to based on each criterion.
> 5. Ok I'll get this plot to you asap.
> 6. Sounds good. I think the best thing about how I worked with error this time around was that I logged the parameter values *after* running error --- after looking through the old code from December 2020 I realized that I was logging parameter values before running error and thus not receiving the "jump out of local minimums" or "new minimum found" benefit error provides.

Ah! Yes, that's important to capture. Probably often it won't change things greatly but best to nab the new and improved values.

[Quoted text hidden]