Comparison of Gregory-Loredo to Bayesian Blocks

Michael Nowak, July 8, 2005

Abstract

The Bayesian Blocks algorithm, applied to the ASCII lightcurve files generated by Arnold Rots for ObsId 635, is compared to the variability results found from Arnold's application of the Gregory-Loredo algorithm to the same lightcurves. The Bayesian Blocks algorithm seems to be slightly more sensitive, and has the advantage of being able to optimally divide the lightcurve into segments of varying length (i.e., find a mixture of very long and very short data segments). However, it is not recommended as a good 'Level 3' substitute for the Gregory-Loredo algorithm since it is only marginally more sensitive, but requires nearly 100 times the run time. (The latter figure is dominated by the run on a single source, 0003, which has >24,000 counts.) It does, however, serve as a good validation of the sensitivity of the Gregory-Loredo algorithm, and can be very useful for users to 'post-process' Level 3 lightcurves in search of finer features.

Introduction

The Bayesian Blocks algorithm of Scargle (unpublished - it's been in the works for a while now) is in a number of ways related and similar to the Gregory-Loredo algorithm (1992, ApJ **398**, 146). The primary difference is that, as opposed to the GL algorithm where the lightcurve is parsed into a successive number of histograms with evenly divided time bins, BB can divide the lightcurve into any number of bins of arbitrary length. GL will find the 'optimal' uniform partitioning of the lightcurve, whereas BB will find the 'optimal' partitioning overall.

The drawback is that BB is an N^2 scheme (which, to be fair, is not unreasonable given that it is finding the optimal solution among 2^N possibilities). The other important caveat about the BB routine is that there are issues with what is the best 'prior' to use in its implementation, and it has a 'penalty factor' for deciding whether to accept a partitioning. This latter factor is something akin to a significance level, but 'significance level' is anathema to Bayesians, and this is not an exact interpretation, as I explain further below. But given proper choices for both, BB can be a fairly robust routine.

Implementation

There's a lot of debate for various Bayesian schemes that deal with Poisson statistics as to what the best prior to use is. Some marginalize over count rate, taken as uniform from 0 to infinity (or instead, an arbitrarily chosen upper bound). Scargle suggested marginalizing over Poisson probability, p, for finding one or more events in an interval as being superior, since p only varies between 0 and 1. (Scargle chooses it to be uniform between 0 and 1.) I think both of these latter priors unfairly penalize large, positive fluctuations as not being 'special' enough. Nor do they incorporate any information from the lightcurve itself. In the tt s-lang code that I have written, I allow for a variety of choices. For my implementation of BB presented here, I marginalize with a prior that is $\propto \exp(-\Lambda/\Lambda_0)$, where Λ is the (presumed constant) rate for the lightcurve, and Λ_0 is the actual mean rate calculated for the lightcurve.

Aside from not unfairly penalizing large fluctuations and taking into account the mean rate, I like this prior in part because in the tests that I have run it seems to produce results for a given 'penalty factor' that are in good accord with its naive, frequentist interpretation as a 'significance level'. (Jeff Scargle doesn't find this a sufficient aesthetic justification for a prior, whereas not being a pure Bayesian, I rather like this justification.)

The 'penalty factor' is basically a prior on the probability for the number of blocks, taken as $\gamma^{N_{blocks}}$. In practice, it is put in as a parameter, `ncp_prior`, such that $\texttt{ncp\_prior} = -\log(\gamma)$. In a frequentist interpretation, this parameter yields the significance level as $1 - \exp(-\texttt{ncp\_prior})$ (or the p value as $\exp(-\texttt{ncp\_prior})$. This is another difference between the BB and GL algorithms. In the latter, an odds ratio (as well as an optimal decomposition) are automatically output. In BB, a 'significance level' must be chosen ahead of time, and the optimal decomposition *for that significance level* is output. (Thus, the

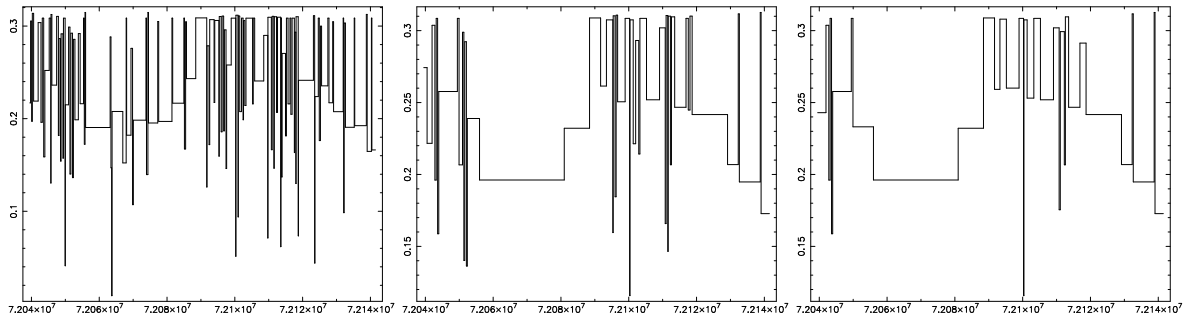

Figure 1: Source 0003, at 'p values' of 3×10^{-4}, 10^{-6}, and 10^{-7}. The peak levels are at 1 count/frame, and probably are piled up. In this particular case, Bayesian Blocks seems to be detecting more variability than Gregory-Loredo.

algorithm must be run multiple times if one wishes to explore different 'significance levels'.) Here I have run tests with `ncp_prior` = 3 and 4.6 (roughly 95% and 99% significance levels).

The implementation of the algorithm comes from my `s-lang` scripts, found at:
`http://space.mit.edu/CXC/analysis/SITAR`
Running it on the ASCII files is pretty straightforward. The `s-lang` script I used is as follows:

```
variable nfile = 118;
variable file,fp,t_event,cell,results;
fp = fopen("bb_results_4.6","w");
() = fclose(fp);
variable ctype=2;
variable clump=0.5;
variable frame=3.24104;
variable ncp_prior=4.6;
variable tmin = 7.2039524e7, tmax = 7.2141507e7;
variable i=1;
loop(nfile)
{
   file = sprintf("%04d",i);
   t_event = readcol("./635/"+file+"_src_time.lis");
   cell = sitar_make_data_cells(t_event,ctype,clump,frame,tmin,tmax);
   results = sitar_global_optimum(cell,ncp_prior,ctype,,0,1);
   fp = fopen("bb_results_4.6","a");
   () = fprintf(fp,file+"  %7i  %5i \n",length(t_event),length(results.cpt));
   () = fclose(fp);
   i++;
}
```

Results

Running at 95% significance level, the BB algorithm detects all of the sources that Arnold listed as 'significantly variable' in his memo, with the exception of Source 0084. Additionally, one that was listed as a 'high background' source (0024), as well as eight other sources, are detected. None of the sources that Arnold listed as 'false detections' are found to be variable. Increasing the threshold significance level to 99%, source 0116 is no longer detected as variable, and the additional detected sources (beyond 0024) decreases from eight to three. Each run takes about 1550 seconds.

A few other interesting results pop out. Observation 003, which takes up more than half the run time in applying the BB algorithm, has excellent statistics. Arnold showed this in his report on GL, and the GL

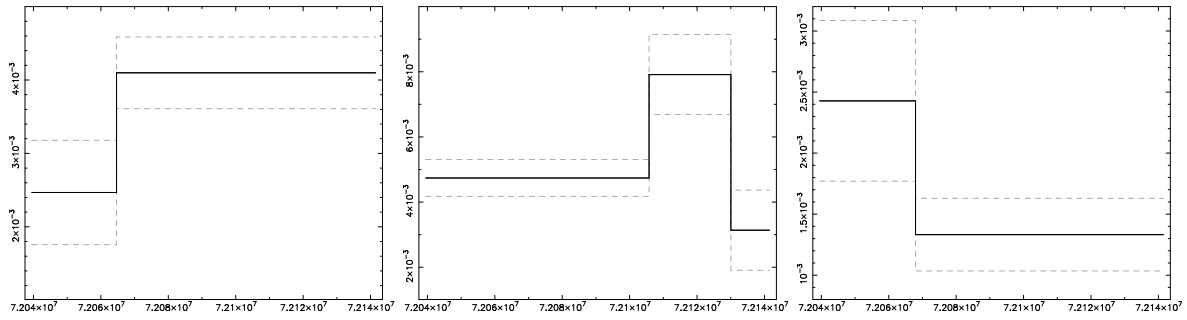

Figure 2: Sources 0018, 0024, and 0051, at 'p values' of 0.01. Error bars are $\pm 2\sigma$ (assuming Gherels statistics - sorry). At this significance level, these are the only three sources detected by Bayesian Blocks, and not by the K-S or G-L tests run by Arnold Rots. I would argue that these lightcurves are consistent with real variability (at least within the chosen significance level).

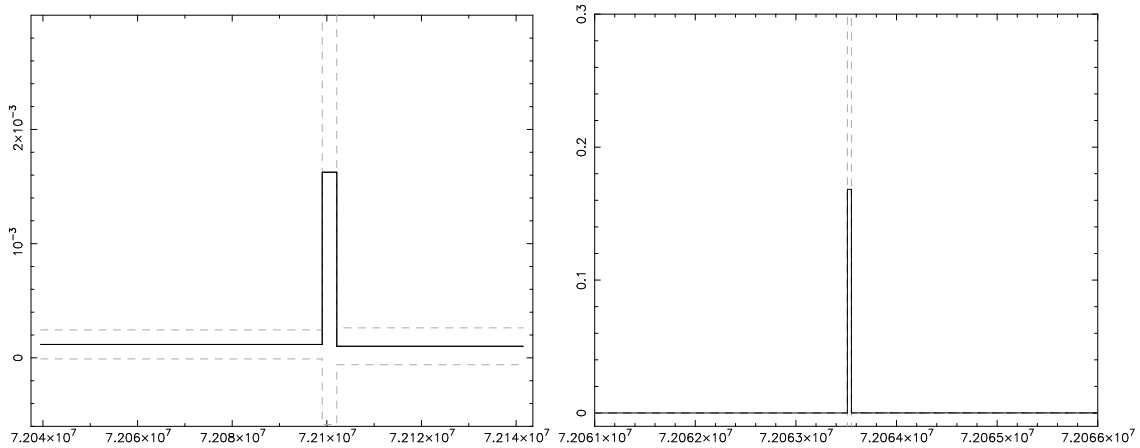

Figure 3: Source 0084 at 78% significance level, and source 0116 at 95% significance level. (Both of these sources were detected by G-L.) The former only pops up at fairly low significance levels, whereas the latter disappears somewhere between 95% and 99% significance. (Again, I apologize for the Gherels statistics.) I'm not sure if either of those results is unrealistic.

decomposition had relatively low temporal resolution. Bayesian blocks finds lots of very significant short time scale variability. (For instance, there is one 114 sec interval with a single photon, when the mean rate would have predicted 23!). GL, misses, or in this case averages over, such behavior. As shown in Fig. 1, this behavior persists to very high significance level. I suspect what is going on in this source is some very serious flaring. The clearly seen plateaus are at a level of one count/frame - the maximum expected for a piled source. The deep dips near these plateaus are probably even stronger flares where pileup is wiping out detected counts (either through grade migration and/or exceeding the on-board spacecraft energy threshold) and leaving a hole. At any rate, I believe that this short term variability is probably real. This observation also shows some of the nastiness that can occur if pileup is an issue.

As for the sources that Bayesian Blocks finds as variable, which Arnold's runs of the GL and KS tests miss, I think these are also truly variable sources. Running BB at the 99% significance level, the three extra detected variable sources are shown in Fig. 2. I think that they get missed by GL since they are decidely asymmetric. The initial, even breaks into two or three pieces that the GL algorithm performs is likely missing the variability, while a larger number of breaks isn't statistically warranted. I don't believe that any of these lightcurves are wildly significant, but I think that a detection algorithm run at a nominal 99% significance level could call these 'variable'.

In contrast, I see why the BB algorithm misses Source 0084 altogether, and then loses Source 0116 somewhere between the 95% and 99% significance levels. As shown in Fig. 3, these sources are on the

hairy edge of what I'd consider variable. They are reasonably consistent with random fluctuations, so again, I can't quibble with the results of BB.

Recommendations

Despite the fact that I believe BB is doing a better job of both detecting and characterizing the variability, I don't think I would recommend it over the GL algorithm. There are two primary reasons for this. First, it is definitely slower. I'm running on a 1.2 GHz AMD processor, and a full run at a single confidence level takes 1550 seconds of CPU time (running consecutively two confidence levels takes 2960 seconds of CPU - there is some overhead in setting up the lightcurve that is saved on the second confidence level run). GL is on the order of 100 times faster. Second, BB is less flexible in that a value of `ncp_prior` must be chosen and then the algorithm run. GL can just be run, the odds ratio printed out, and the most optimal lightcurve decomposition can be stored, regardless of the odds ratio. We can set a flag based on an agreed upon odds ratio, but a user could potentially filter on whatever value they like. With BB, they are stuck with what we give them.

The BB algorithm run at a 99% significance threshold (`ncp_prior`=4.6); however, I think serves as a good validation of the GL results at an odds ratio of 2. The two alogorithms agree on 43 sources. BB finds 4 that GL does not, while GL finds one that BB does not. Essentially, they are in 90% agreement.

My one concern is learning how to identify situations like shown in Fig. 1, where the source is likely piling up. (Although, in general, dealing with pileup in the L3 pipeline is not a fully resolved issue.) But at some level I think the L3 pipeline is to point out behavior for further study, not necessarily provide its definitive description. GL certainly flags Source 0003 as interesting, and a user could use BB to study it further.

Table

Here I provide a list of the variable sources for ObsId 635, as detected by BB run at a threshold of 95% (`ncp_prior`=3.0) and 99% (`ncp_prior`=4.6) significance levels. N_event is the number of photon events in the ASCII file. N_cpt is the number of 'change points' (= the number of blocks the lightcurve has been decomposed into). I'm only printing those results where one of the runs produced more than one block. You can see in general that increasing the value of `ncp_prior` smooths out the lightcurve decomposition.

Source	N_event	N_cpt 0.95	N_cpt 0.99
0001	8674	20	8
0002	17450	16	8
0003	24093	443	288
0004	6775	29	8
0005	2992	14	11
0006	2831	10	6
0007	1115	4	4
0008	4445	110	35
0009	1106	13	8
0010	1102	50	31
0011	8697	16	9
0012	1156	3	1
0013	1401	13	11
0014	1323	5	5
0015	4847	15	2
0016	484	3	3

0017	785	7	6
0018	377	2	2
0019	3247	8	6
0020	1025	13	8
0021	659	3	3
0022	651	8	5
0023	220	2	2
0024	542	3	3
0025	170	2	1
0026	199	4	4
0027	2710	6	4
0028	336	2	2
0031	287	2	1
0032	291	3	3
0033	171	3	3
0034	104	3	1
0035	129	3	2
0037	100	2	2
0039	76	2	1
0041	2040	2	2
0043	149	3	3
0044	162	3	1
0049	67	3	2
0051	167	2	2
0052	98	2	2
0053	57	4	3
0054	138	3	3
0055	69	3	3
0056	68	3	3
0058	530	2	2
0060	58	2	2
0062	38	3	2
0074	47	2	2
0078	328	6	5
0080	135	2	2
0095	25	2	1
0096	93	4	3
0116	14	3	1