Doing Right By Massive Data: Using Probability Modeling To Advance The Analysis Of Huge Astronomical Datasets

Alexander W Blocker

17 April, 2010

Outline

What is massive data?

- In short, it's data where our favorite methods stop working
- Orders of magnitude more observations than we are used to dealing with, often combined with high dimensionality (e.g. 40 million time series with thousands observations each)
- Such scale of data is increasingly common in fields such as astronomy, computational biology, ecology, etc.
- Need statistical methods that scale to these quantities of data
- However, need to tradeoff statistical rigor and computational efficiency

Machine Learning vs. Statistics, in broad strokes

Statistical Methods

- Heavy computational burden
- Highly customizable
- Can handle "messy" data
- Internal assessment of uncertainty

Machine Learning Methods

- Computationally efficient
- Generically applicable
- Need clean input
- External assessment of uncertainty

How can we get the best of both worlds?

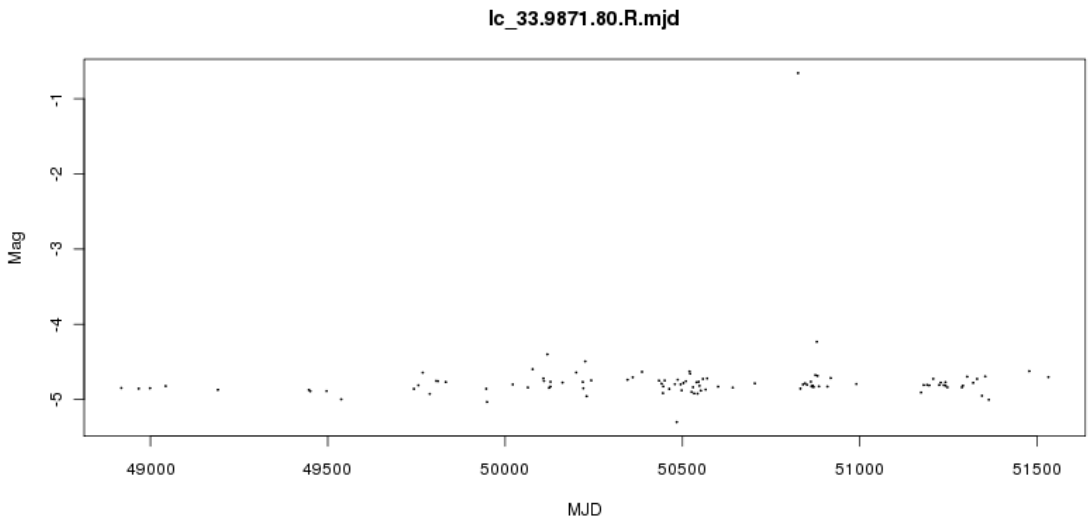
- Principled statistical methods are best for handling messy, complex data that we can effectively model, but scale poorly to massive datasets
- Machine learning methods handle clean data well, but choke on issues we often confront (outliers, nonlinear trends, irregular sampling, unusual dependence structures, etc.)
- Idea: Inject probability modeling into our analysis in the right places

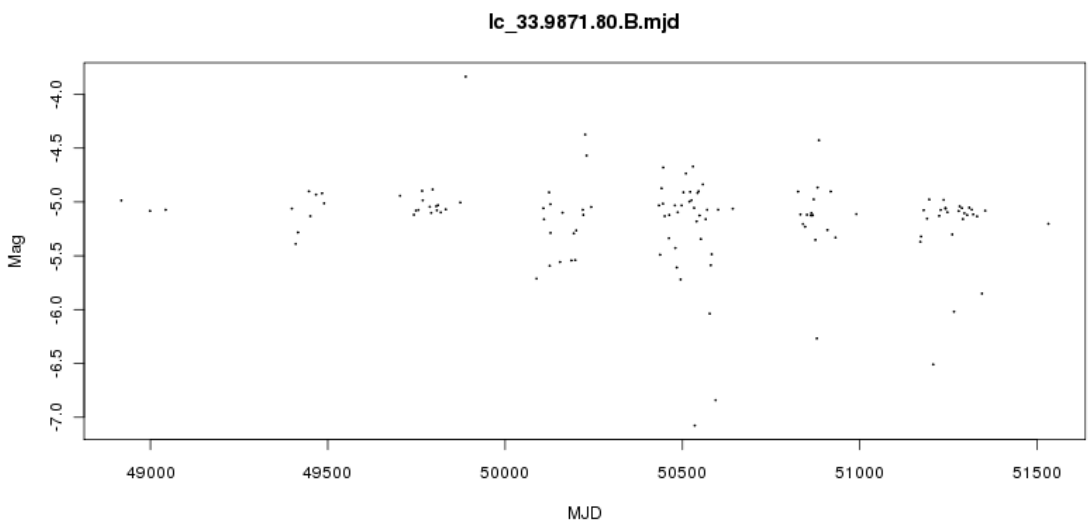
The Problem

- Massive database of time series (approximately 40 million) from the MACHO project (cover the LMC for several years)
- Goal is to identify and classify time series containing events
- How do we define an event?
  - Not interested in isolated outliers
  - Looking for groups of observations that differ significantly from those nearby (ie, "bumps" and "spikes")
  - Also attempting to distinguish periodic and quasi-periodic time series from isolated events

Exemplar time series from the MACHO project:

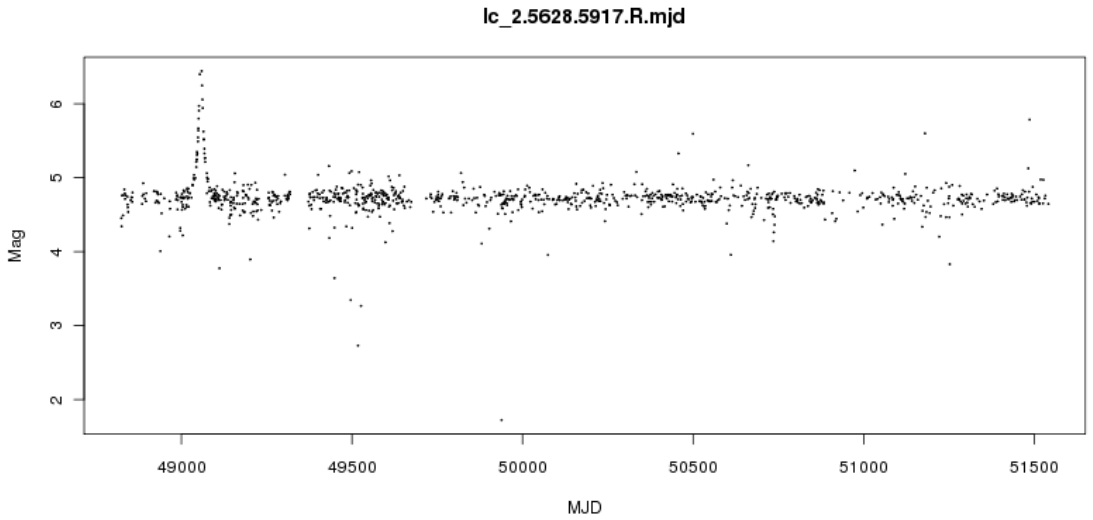
A null time series:

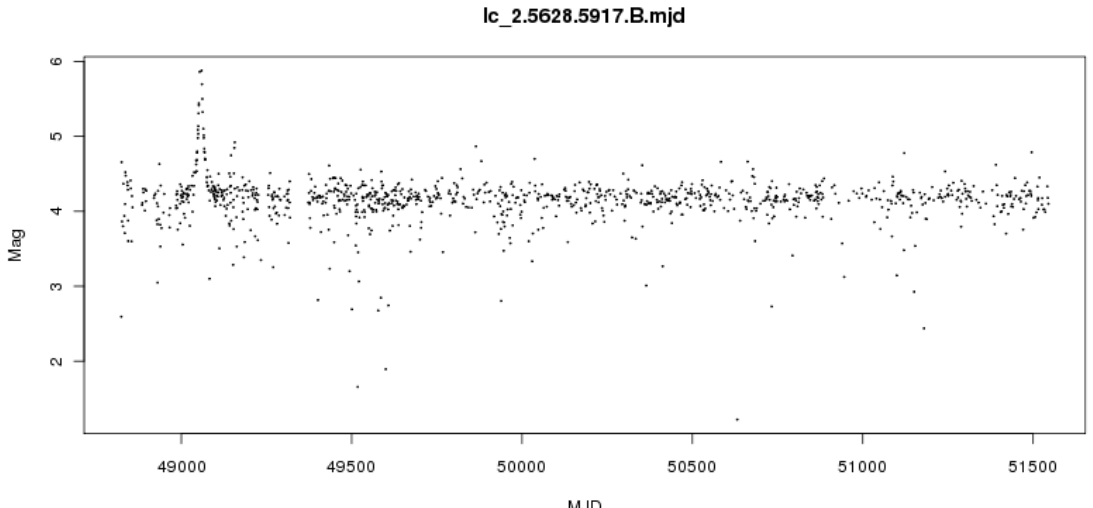




Exemplar time series from the MACHO project:

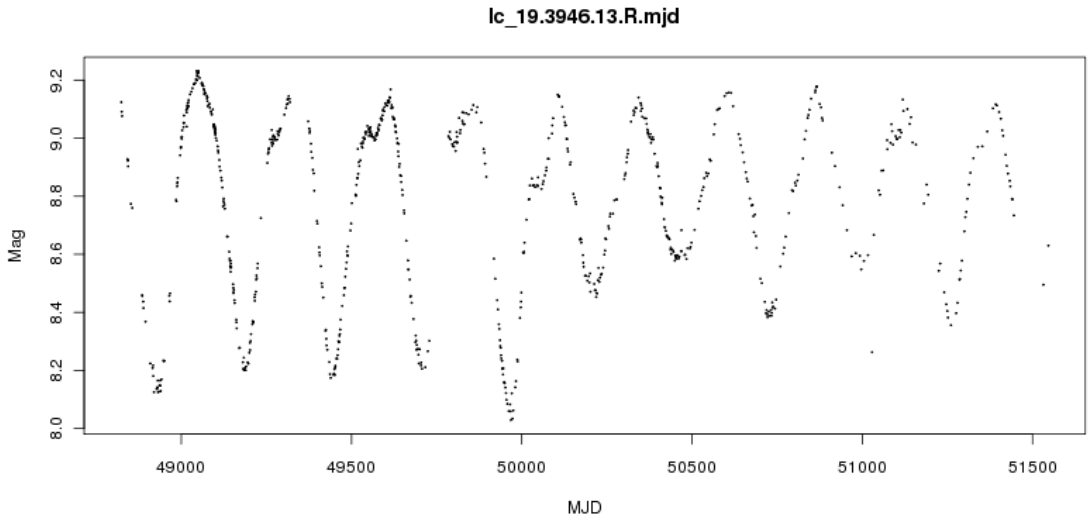
An isolated event (microlensing):

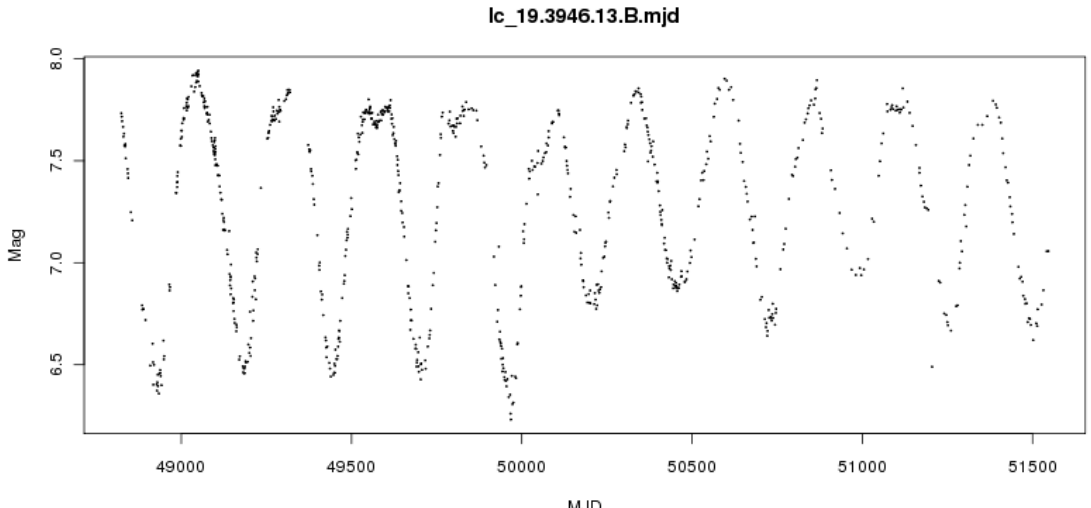




Exemplar time series from the MACHO project:

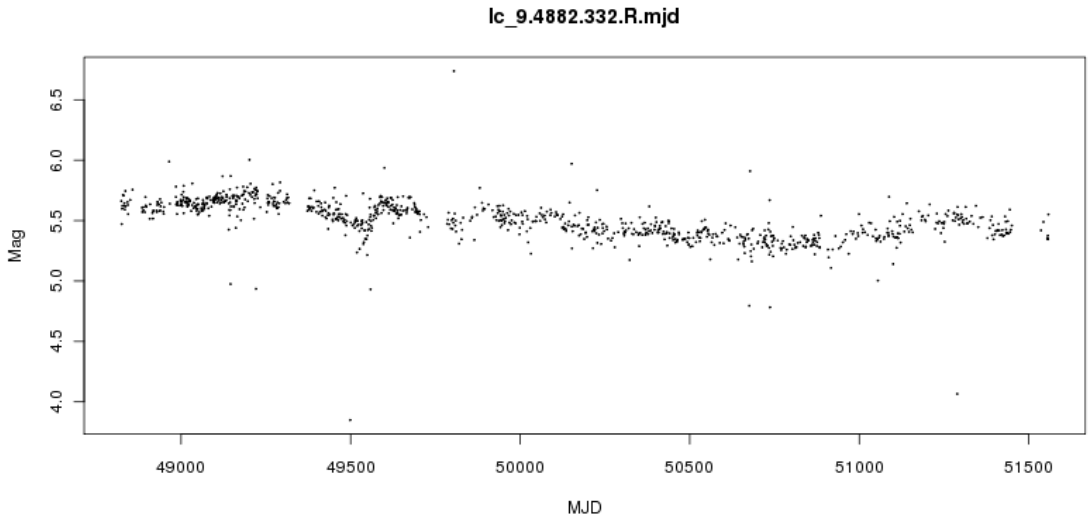
A quasi-periodic time series (LPV):

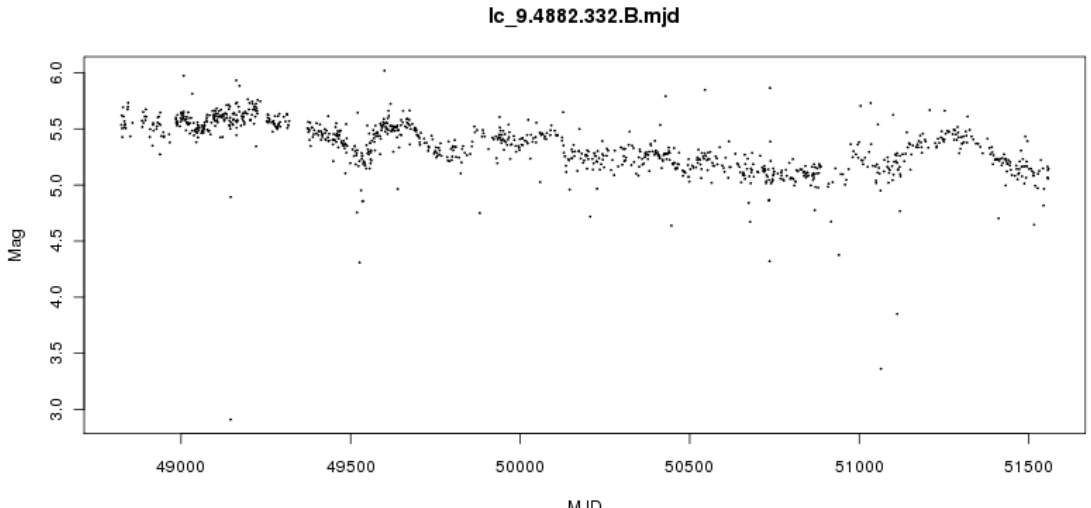




Exemplar time series from the MACHO project:

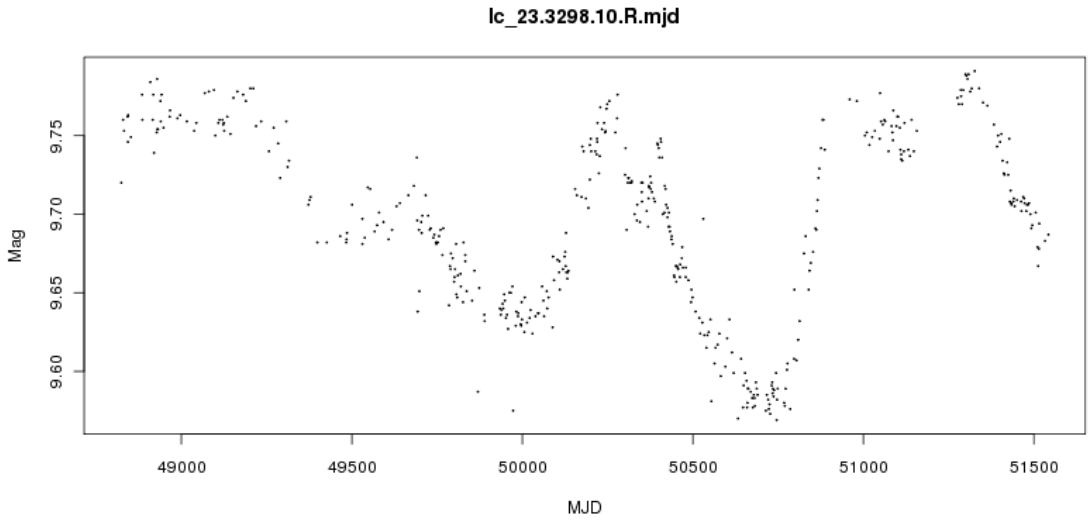
A variable time series (quasar):

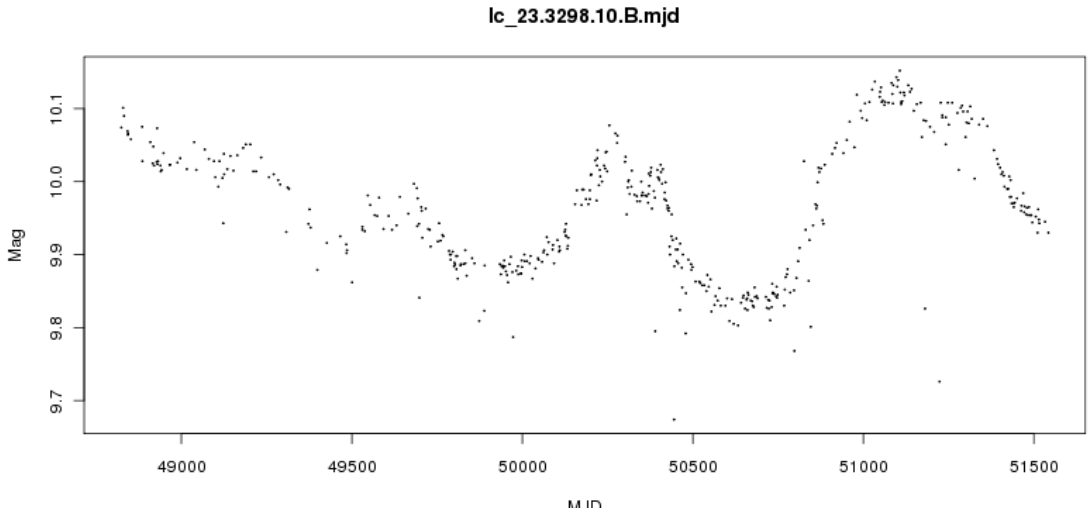




Exemplar time series from the MACHO project:

A variable time series (blue star):





Notable properties of this data

- Fat-tailed measurement errors
  - Common in astronomical data, especially from ground-based telescopes
  - Need more sophisticated models for the data than standard Gaussian approaches
- Quasi-periodic and other variable sources
  - Changes the problem from binary classification (null vs. event) to k-class
  - Need more complex test statistics and classification techniques
- Non-linear, low-frequency trends make less sophisticated approaches far less effective
- Irregular sampling can create artificial events in naïve analyses

Previous approaches to event detection

- Scan statistics are a common approach (Liang et al, 2004; Preston & Protopapas, 2009)
  - However, they often discard data by working with ranks and account for neither trends nor irregular sampling
- Equivalent width methods (a scan statistic based upon local deviations) are common in astrophysics
  - However, these rely upon Gaussian assumptions and crude multiple testing corrections
- Numerous other approaches have been proposed in the literature, but virtually all rely upon Gaussian distributional assumptions, stationarity, and (usually) regular sampling

Our approach

- Use a Bayesian probability model for both initial detection and to reduce the dimensionality of our data (by retaining posterior summaries)
- Using posterior summaries as features for machine learning classification technique to differentiate between events & variables
- Symbolically, let V be the set of all time series with variation at an interesting scale (ie, the range of lengths for events), and let E be the set of events
- For a given time series Y_i, we are interested in $P(Y_i \in E)$
- We decompose this probability as
$$P(Y_i \in E) = P(Y_i \in V) \cdot P(Y_i \in E | Y_i \in V)$$
via the above two steps

Probability model - specification

- Linear model for each time series with a split wavelet basis:

$$y(t) = \sum_{i=1}^{k_l} \beta_i \phi_i(t) + \sum_{j=k_l+1}^{M} \beta_j \phi_j(t) + u(t)$$

- Assume that our residuals $u(t)$ are distributed as iid $t_\nu(0, \sigma^2)$ random variables to account for extreme residuals ($\nu = 3$)
- Using a Symmlet 4 (aka Least Asymmetric Daubechies 4) wavelet basis
- $(\phi_1, \ldots, \phi_{k_l})$ contains the low-frequency components of a wavelet basis, and $(\phi_{k_l+1}, \ldots, \phi_M)$ contains the mid-frequency components
- For a basis on $(1, 2048)$, we set k_l to 8 and M to 128

Probability model - specification

$$y(t) = \sum_{i=1}^{k_l} \beta_i \phi_i(t) + \sum_{j=k_l+1}^{M} \beta_j \phi_j(t) + u(t)$$

- Idea: $(\phi_1, \ldots, \phi_{k_l})$ will model structure due to trends, and $(\phi_{k_l+1}, \ldots, \phi_M)$ will model structure at the scales of interest for events
- Explicitly accounting for irregular sampling in our time series through this basis formulation
- Placing independent Gaussian priors on all coefficients except for the intercept

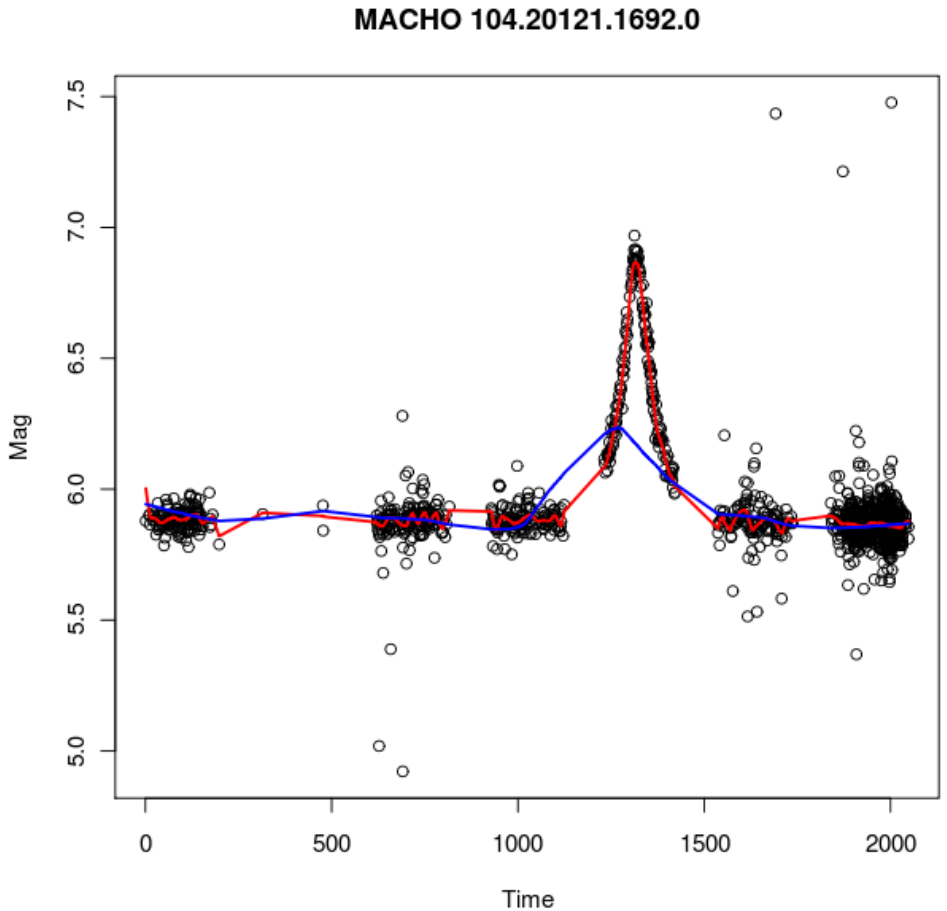
Probability model - estimation

$$y(t) = \sum_{i=1}^{k_l} \beta_i \phi_i(t) + \sum_{j=k_l+1}^{M} \beta_j \phi_j(t) + u(t)$$

- Using EM algorithm with optimal data augmentation scheme of Meng & Van Dyk (1997) to obtain MAP estimates of our parameters
- Using the `speedglm` package in R the weighted least-squares step of this algorithm
- Average time for a full estimation procedure is ≈ 0.4 seconds including file I/O on the Odyssey cluster

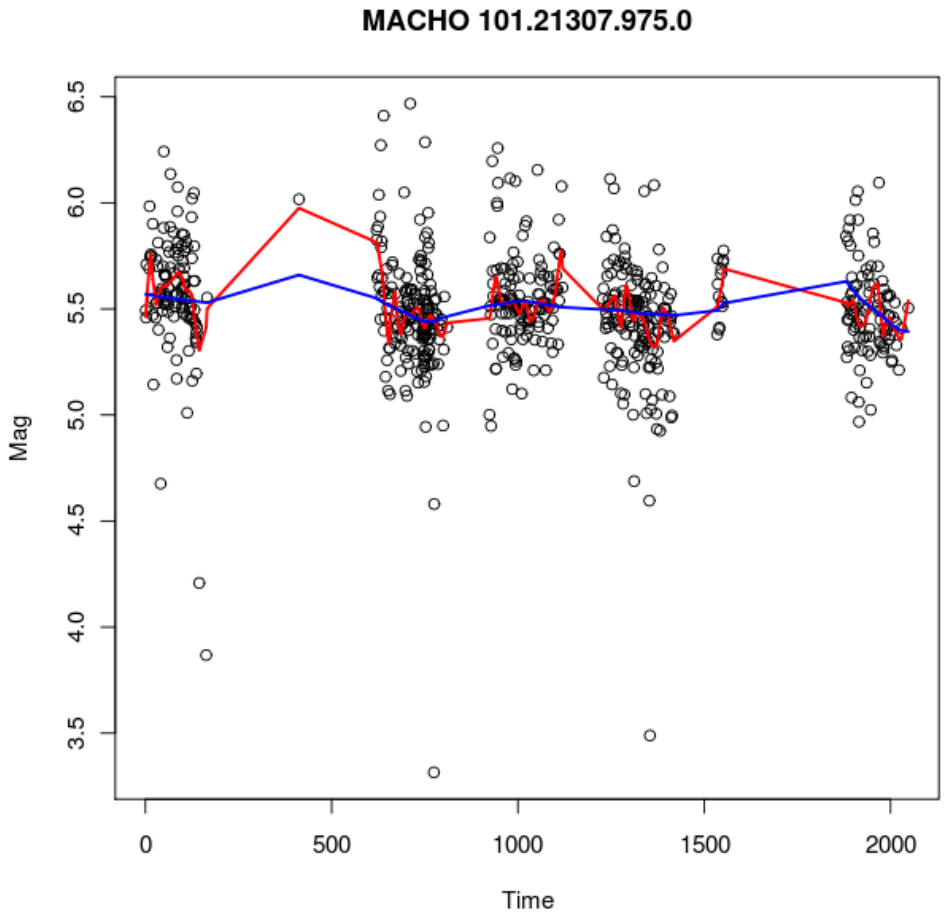
Examples of model fit

Idea is that, if there is an event at the scale of interest, there will be a large discrepancy between fits using $(\phi_1, \ldots, \phi_{k_l})$ vs. entire basis:



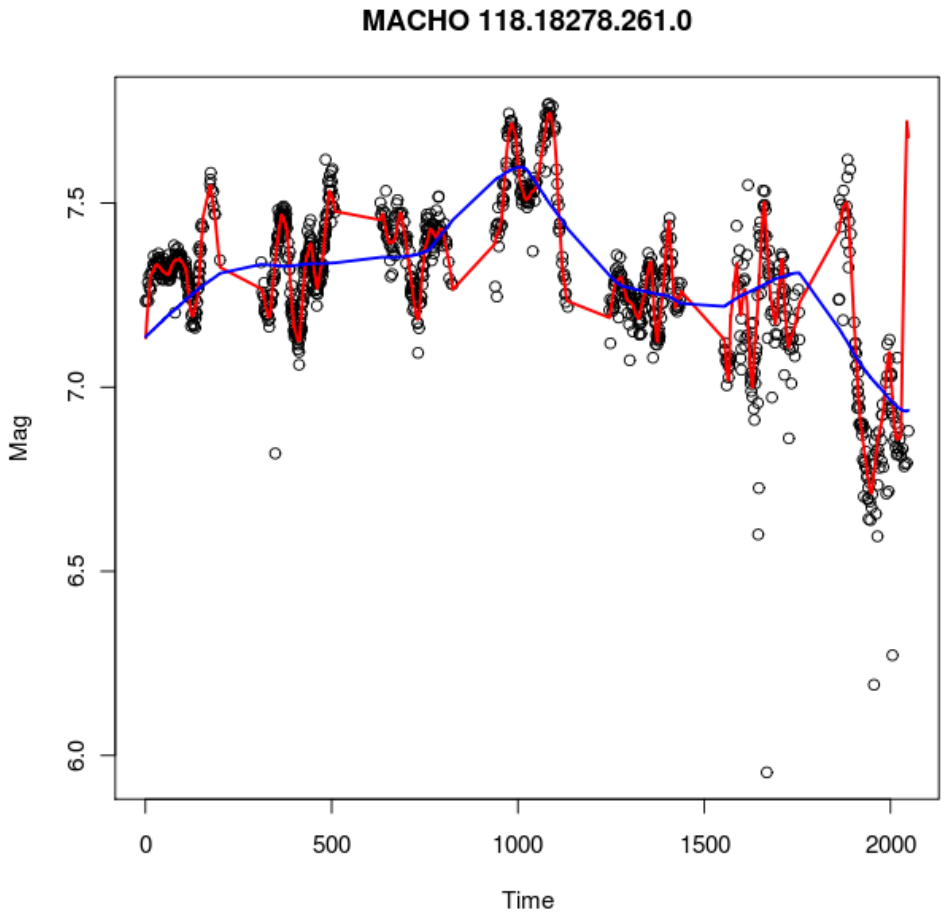
Example of model fit

For null time series, the discrepency will be small:



Example of model fit

And for quasi-periodic time series, the discrepency will be huge:



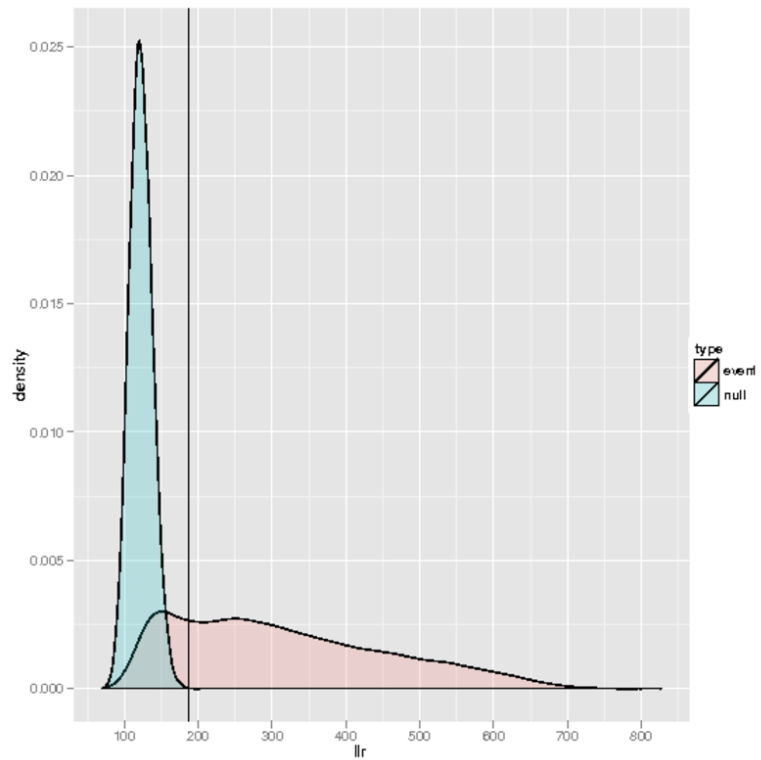
Probability model - testing

$$y(t) = \sum_{i=1}^{k_l} \beta_i \phi_i(t) + \sum_{j=k_l+1}^{M} \beta_j \phi_j(t) + u(t)$$

- We screen time series for further examination by testing $H_0 : \beta_{k_l+1} = \beta_{k_l+2} = \ldots = \beta_M = 0$
- Test statistic is $2(\hat{\ell}_1 - \hat{\ell}_0)$
- Using modified Benjamini-Hochberg FDR procedure with a maximum FDR of 10^{-4} to set the critical region for our test statistic

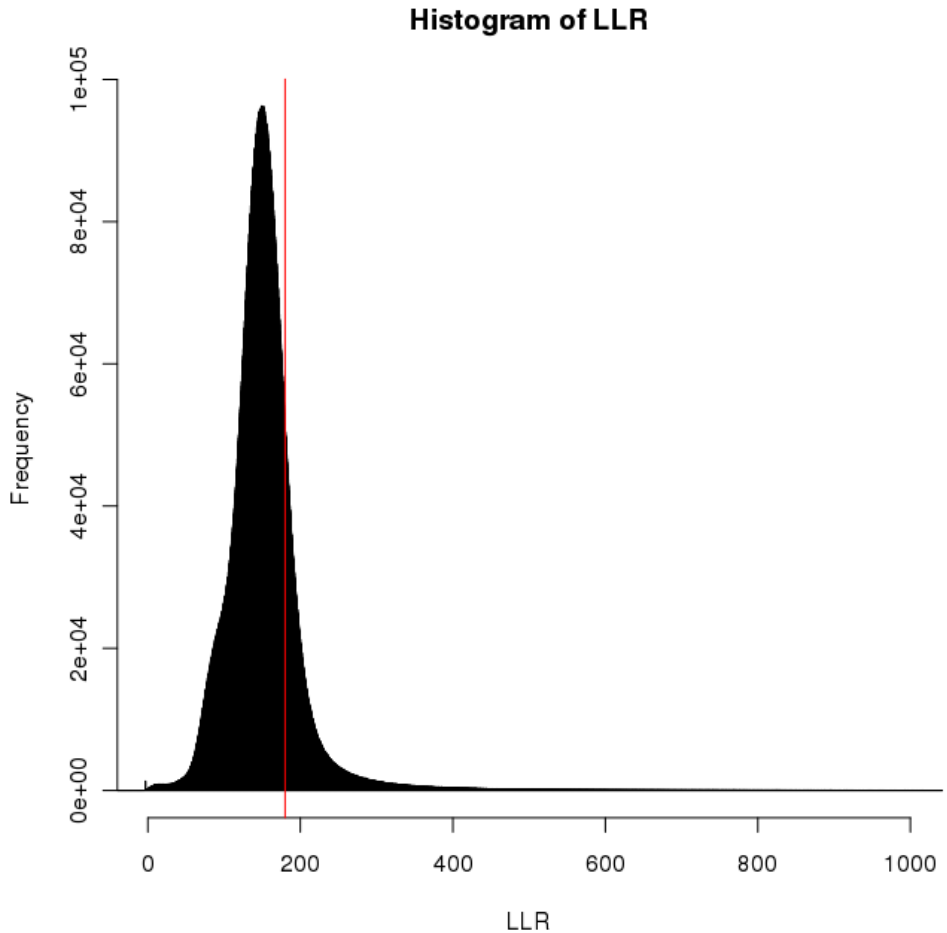
Distribution of LLR statistic

- To assess how well this statistic performs, we simulated $50,000$ events from a physics-based model and $50,000$ null time series
- We obtained approximate power of 80% with the stated FDR based on this simulated data



Results of likelihood ratio screening

- Reduced the set of data for further examination from 39.5 million time series to approximately 3.58 million



Feature Selection

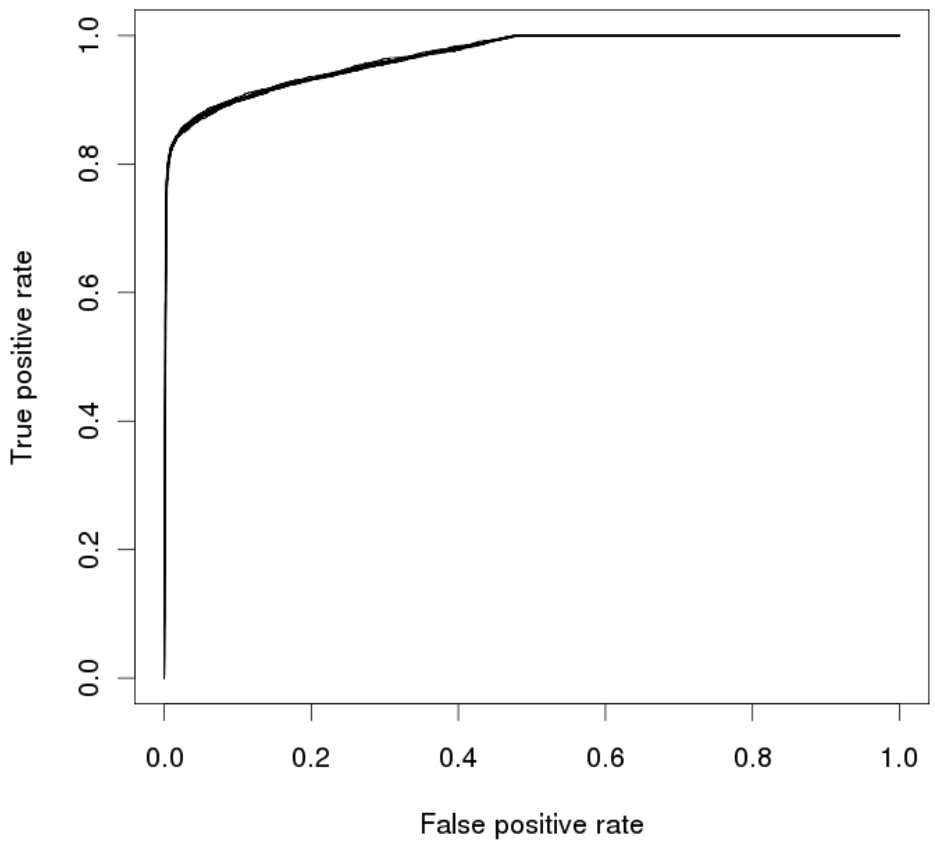
- Using estimated wavelet coefficients as base for features
- These provide a rich, clean representation of each time series, following detrending and denoising (from our MAP estimation)
- Using order statistics for each level of wavelet coefficients as feature
  - Invariant to location of event, but retain other available information

Methods

- Tested a wide variety of classifiers on our training data, including kNN, SVM (with radial and linear kernels), LDA, QDA, and others
- Regularized logisitic regression performs best
- Using weakly informative (Cauchy) prior for regularization

Training

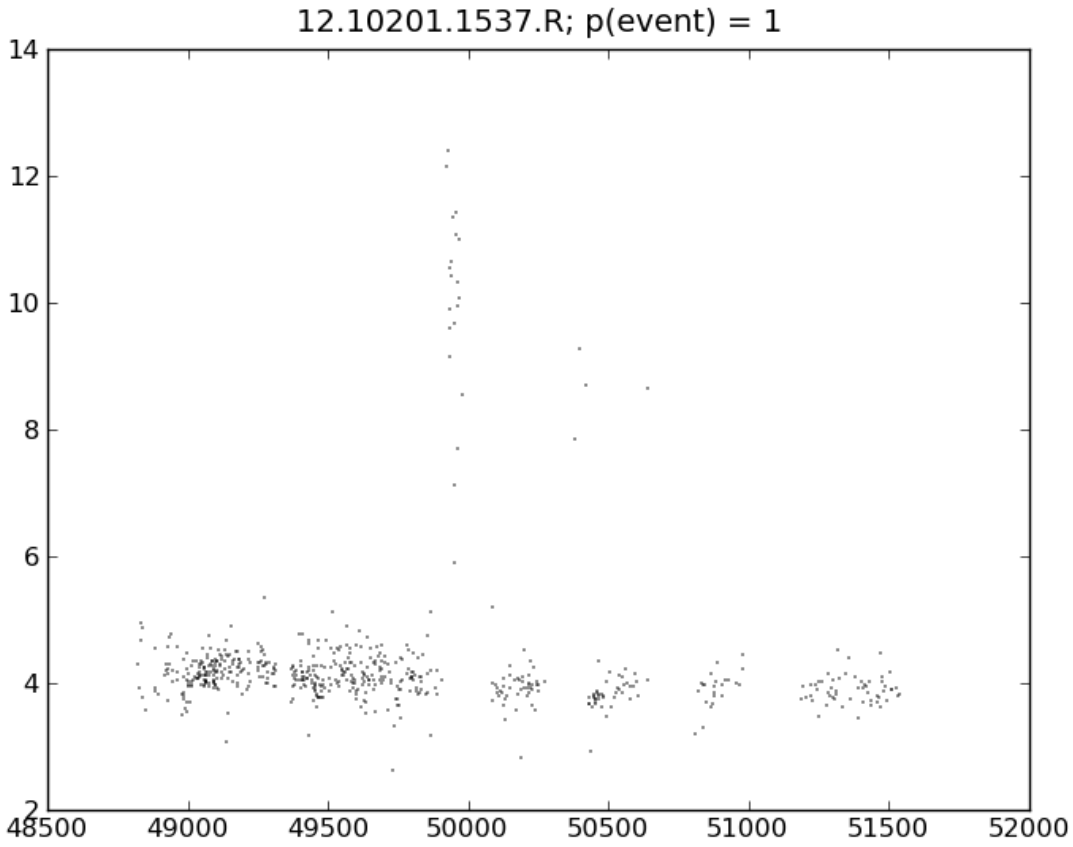
- Obtained excellent performance (cross-validated AUC of 0.97) on training data



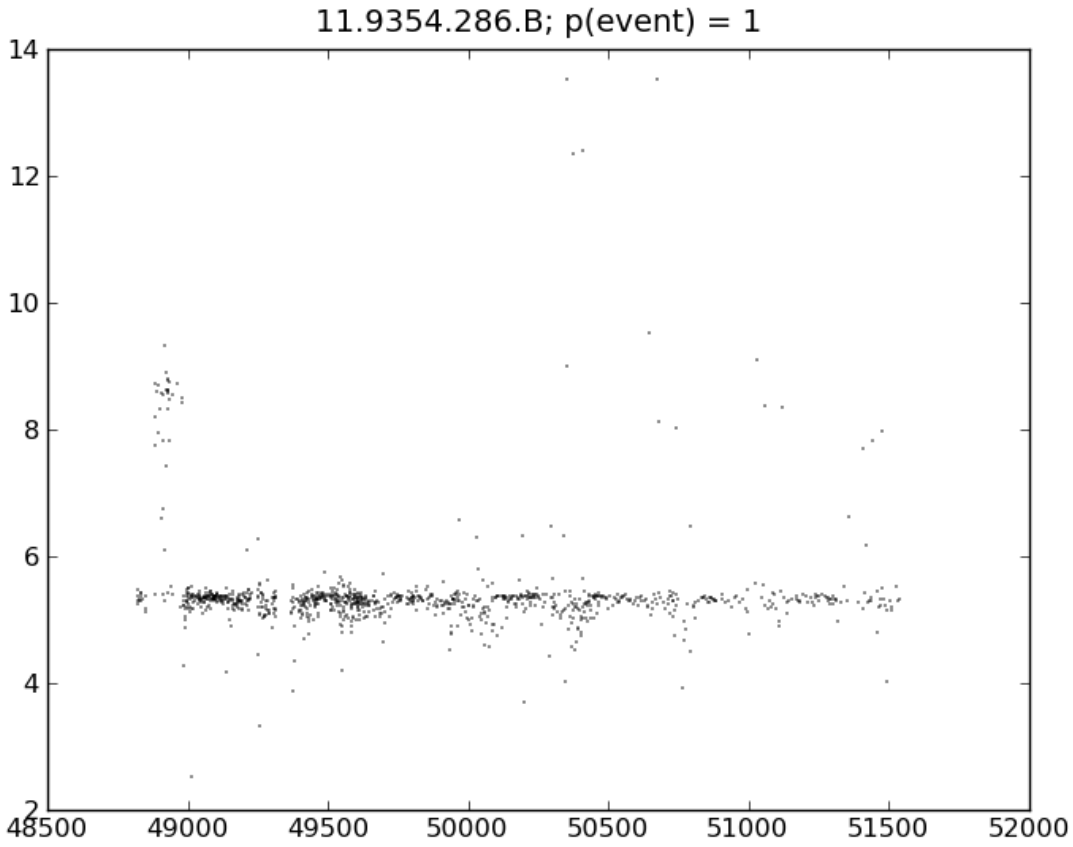
Results

- Identified a subset of less than $10,000$ candidates to examine
- Currently pursuing scientific follow-up on these with collaborators

Two examples:



Two examples:



Putting everything in its place: a mental meta-algorithm

- Understand your full (computationally infeasible) statistical model
- Preprocess to remove the "chaff", when possible
  - Be careful! Any prescreening must be extremely conservative to avoid significantly biasing your results
- Use approximations for the critical parts of your models (e.g. empirical Bayes as opposed to full hierarchical modeling) to maintain computational feasibility
  - Hyperparameters can be set based on scientific knowledge (if priors are sufficiently informative) or setup simply for mild regularization (if each observation is sufficiently rich)
  - Otherwise, a random subsample of the data can be used to obtain reasonable estimates

Putting everything in its place: a mental meta-algorithm

- Using estimates from your probability model as inputs, apply machine learning methods for computationally intractable tasks (e.g. for large scale classification or clustering)
  - This maintains computational efficiency and provides these methods with the cleaner input they need to perform well
- Use scale to your advantage when evaluating uncertainty
  - With prescreening, use known nulls
  - Without prescreening, use pseudoreplications or simulated data

Summary

- Massive data presents a new set of challenges to statisticians that many of our standard tools are not well-suited to address
- Machine learning has some valuable ideas and methods to offer, but we should not discard the power of probability modeling
- Conversely, reasonably sophisticated probability models can be incorporated into the analysis of massive datasets without destroying computational efficiency if appropriate approximations are used
- It is tremendously important to put each tool in its proper place for these types of analyses
- Our work on event detection for astronomical data shows the power of this approach by combining both rigorous probability models and standard machine learning approaches

Acknowledgements

- Many thanks to both Pavlos Protopapas and Xiao-Li Meng for their data and guidance on this project
- I would also like to thank Edo Airoldi for our discussions on this work and Dae-Won Kim for his incredible work in setting up the MACHO data

A sidenote: Why not use a Bayes factor?

- Given our use of Bayesian models, a Bayes factor would appear to be a natural approach for the given testing problem
- Unfortunately, these do not work well with "priors of convenience", such as our Gaussian prior on the wavelet coefficients
- Because of these issues, the Bayes factor was extremely conservative in this problem for almost any reasonable prior

Distribution of Bayes factor