Doing Right By Massive Data: How To Bring Probability Modeling To The Analysis Of Huge Datasets Without Taking Over The Datacenter

Alexander W Blocker Pavlos Protopapas Xiao-Li Meng

9 February, 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. 10 million time series of 1,000 observations each)
- Increasingly common in fields such as astronomy, computational biology, ecology, etc.
- Need statistical methods that scale to these quantities of data
- Question of statistical rigor vs. computational efficiency

Machine Learning methods: strengths & weaknesses

- Strengths:
  - Computationally efficient $\rightarrow$ scale well to large datasets
  - Relatively generic in their applicability
  - Often seem to "just work"
- Weaknesses:
  - Typically do not provide assessments of uncertainties
  - Lack of application-specific modeling $\rightarrow$ inefficient use of available data
  - Often statistically unprincipled

Statistical methods / Probability models: strengths & weaknesses

- Strengths:
  - Based on sound principles
  - Can build complex probability models appropriate to the particular application
  - Rigorous assessments of uncertainties
- Weaknesses:
  - Computation often scales very poorly with the size of the dataset ($O(n^2)$ or worse, especially for complex hierarchical models)
  - Modeling diverse, complex patterns in the data can require a very large amount of application- (and data-) specific modeling
  - Computation often does not parallelize well

How should we combine?

- Principled statistical methods are best for handling messy, complex data, but scale poorly
- Machine learning methods handle cleaner data well, but choke on issues we often confront (outliers, low counts, nonlinear trends, irregular sampling, etc.)
- Idea: Use probability modeling in the right places

Putting everything in its place

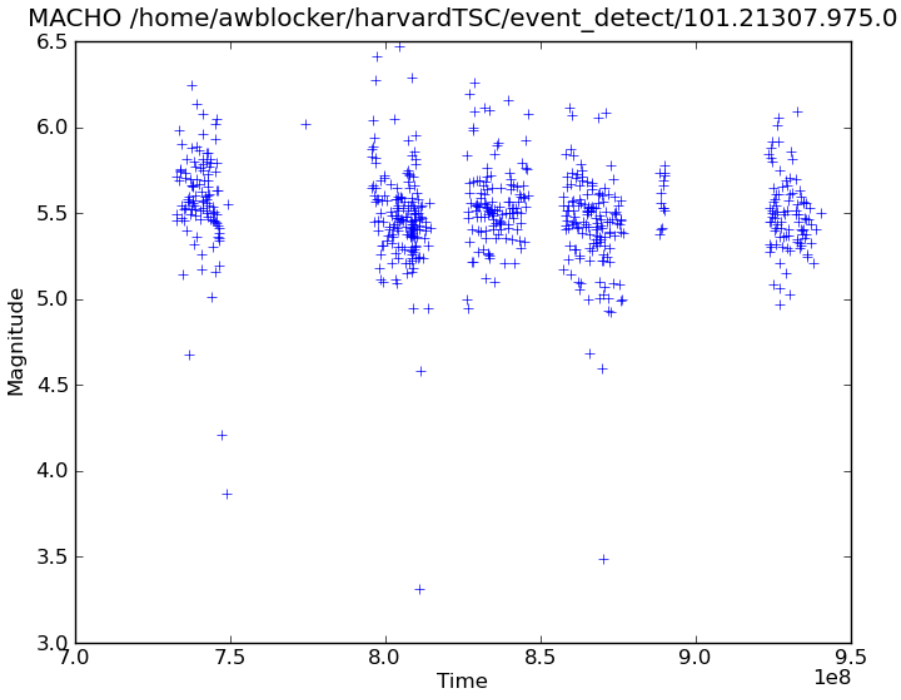
- Understand what your full (computationally infeasible) model is; this guides the rest of your decisions
- Preprocess to remove the "chaff", when possible
  - Be careful! Any prescreening must be extremely conservative to avoid significantly biasing your results
- Use approximations for the critical parts of your models (e.g. empirical Bayes as opposed to full hierarchical modeling) to maintain computational feasibility
- Apply machine learning methods as needed (e.g. for large scale classification) with the estimates from your probability model as inputs. This maintains computational efficiency and provides these method with the cleaner input they need.
- Use scale to your advantage when evaluating uncertainty
  - With prescreening, use known nulls
  - Without prescreening, use pseudoreplications
- How do we apply this?

The Problem

- Massive database of time series (approximately 10 million) from the MACHO project (many more are coming soon from PanSTARRs)
- Our goal is to identify and classify time series containing events
- How do we define an event?
  - We are not interested in isolated outliers. This differentiates our problem from traditional "anomaly detection" approaches.
  - We are looking for groups of observations that differ significantly from those nearby.
  - We are also attempting to distinguish periodic and quasi-periodic time series from isolated events, as they have very different scientific interpretations.

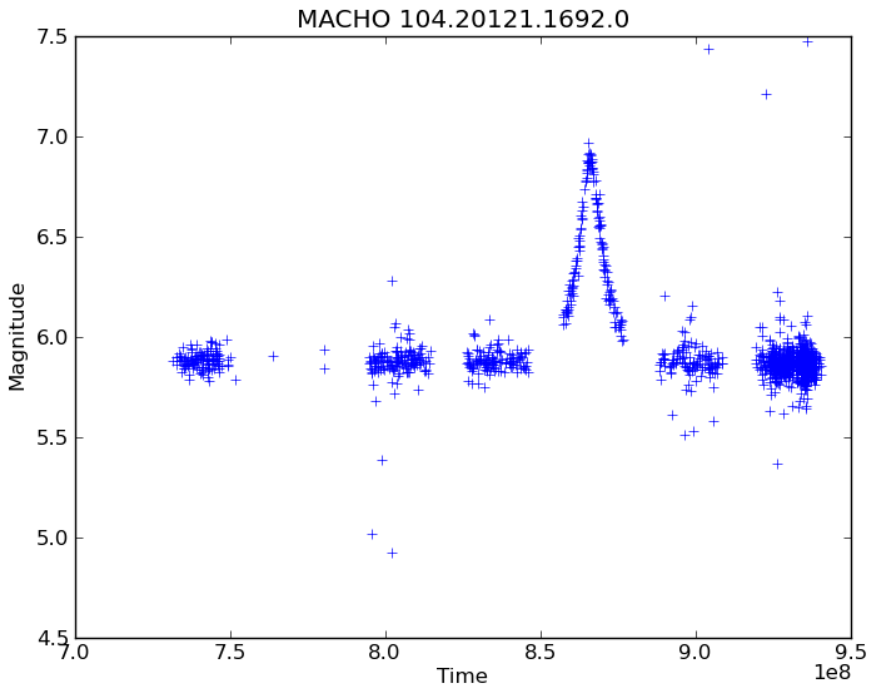
Exemplar time series from the MACHO project:

A null time series:



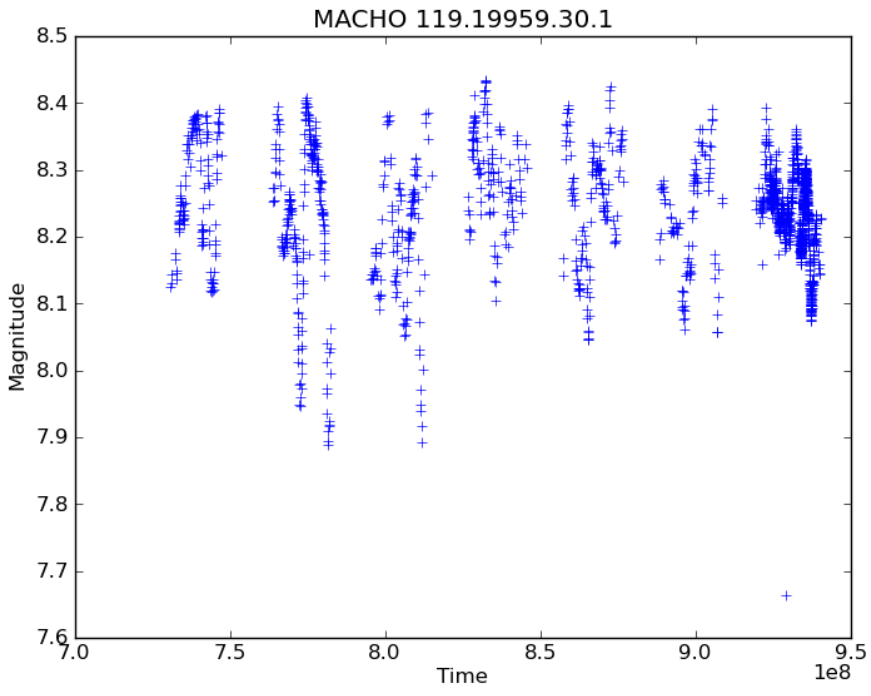
Exemplar time series from the MACHO project:

An isolated events:



Exemplar time series from the MACHO project:

A quasi-periodic time series:



Notable properties of this data

- Fat-tailed measurement errors
  - Common in astronomical data, especially from ground-based telescopes
  - Requires more sophisticated modeling of data than Gaussian approaches.
- Quasi-periodic sources
  - Changes problem from binary classification to k-class
  - Requires more complex test statistics
- Non-linear, low-frequency trends confound our analysis further and make less sophisticated approaches far less effective
- Irregular sampling, which can create artificial events if handled incorrectly
- Oh my!

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 do not account for trends
- Equivalent width methods are common in astrophysics
- However, these rely upon Gaussian assumptions and crude multiple testing corrections

Preprocessing: a modified CUSUM

- Following the framework given in the previous section, we begin with a preprocessing step.
- A CUSUM test is a simple and appropriate choice.
- These have a long history in change-point detection in industrial statistics and econometrics (e.g. Ploberger, 1992, Page 1954).
- The test statistic is the range of the cumulative sum of deviations from the mean (or from a fitted linear trend):

$$S_t = \frac{1}{\hat{\sigma}\sqrt{T}} \sum_{j=0}^{t} (Y_j - \hat{Y}_j)$$

$$R = \max_t(S_t) - \min_t(S_t)$$

Preprocessing: a modified CUSUM

- For T large, and assuming Gaussian residuals, the distribution of R can be approximated by the distribution of the range of a Brownian bridge.
- We use the range of our CUSUM series as our statistic (as opposed to its maximum or minimum) because we are not making an assumption as to the direction of any event in our time series.
- There is one caveat with the use of the CUSUM. If outliers are present, standard estimators of σ have a large upward bias, making the test far too anticonservative for our purposes. To correct this, we use a robust estimator of σ: the (rescaled) median absolute deviation.

How well does our preprocessing work?

- On our test sample of 515,136, we eliminate approximately 13.2% of our time series with a cut at $\alpha = 0.01$ for the modified CUSUM statistic
- We then use these series to build a null distribution for subsequent testing
- Visual inspection of time series near the threshold confirmed that our preprocessing was conservative

Probability model

- We assume a linear model for our observations:

$$Y = X_\ell \beta_\ell + X_m \beta_m + u$$

- We assume that our residuals u_t are distributed as iid $t_\nu(0, \sigma^2)$ random variables to account for extreme residuals (we set $\nu = 3$).
- X_ℓ contains the low-frequency components of a wavelet basis, and X_m contains the mid-frequency components
  - We use a symmlet 4 (aka Least Asymmetric Daubechies 4) wavelet basis; it's profile matches the events of interest quite well
  - For a basis of length 2048, we build X_ℓ to contain the first 16 coefficients; X_m contains the next 112
- Idea: X_ℓ will model structure due to trends, X_m will model structure at the scales of interest for events

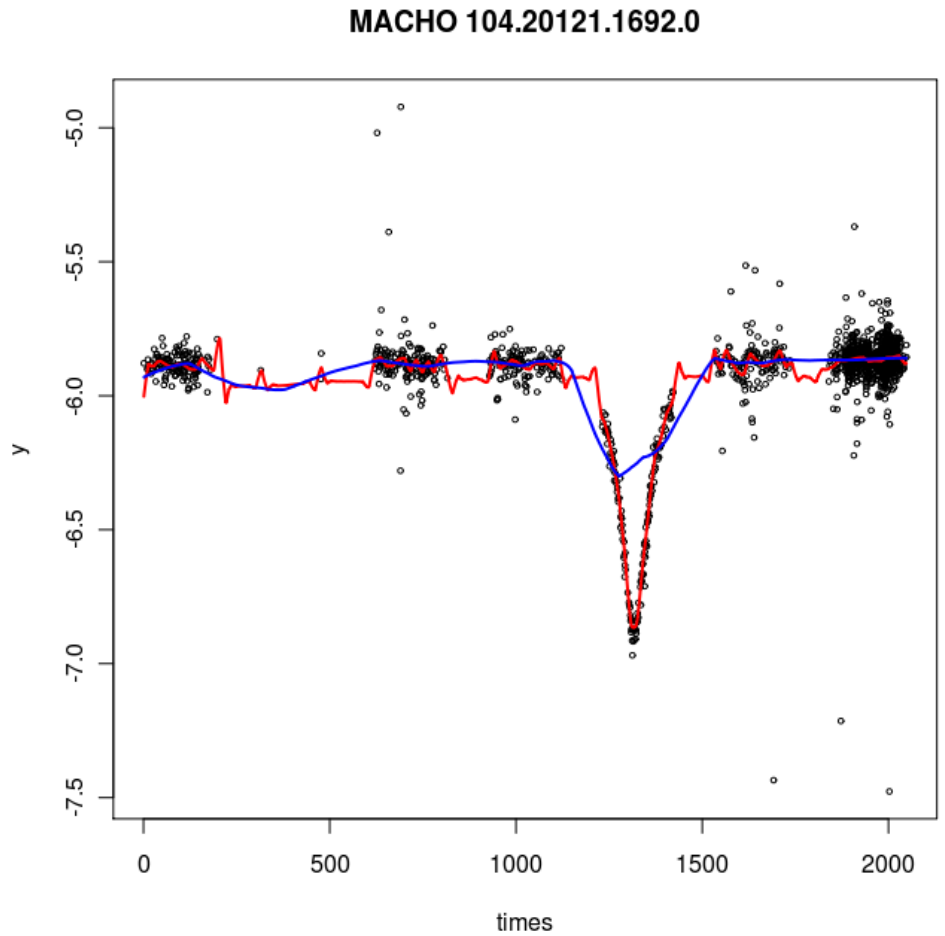
Probability model

$$Y = X_\ell \beta_\ell + X_m \beta_m + u$$

- We explicitly account for irregular sampling in our time series by interpolating our basis to the observation times of our data
- We place independent Gaussian priors on all coefficients except for the intercept to reflect prior knowledge and regularize estimates in undersampled regions
- We use the optimal data augmentation scheme of Meng & Van Dyk (1997) with the EM algorithm to fit our model (average time for a full estimation procedure is ≈ 0.2 seconds including file I/O, using the `speedglm` package in R)
- We use a modified LLR statistic to test for the presence of variation at the scales of interest (testing $\beta_m = 0$). Its null distribution is estimated from the data excluded as nulls in the preprocessing stage.

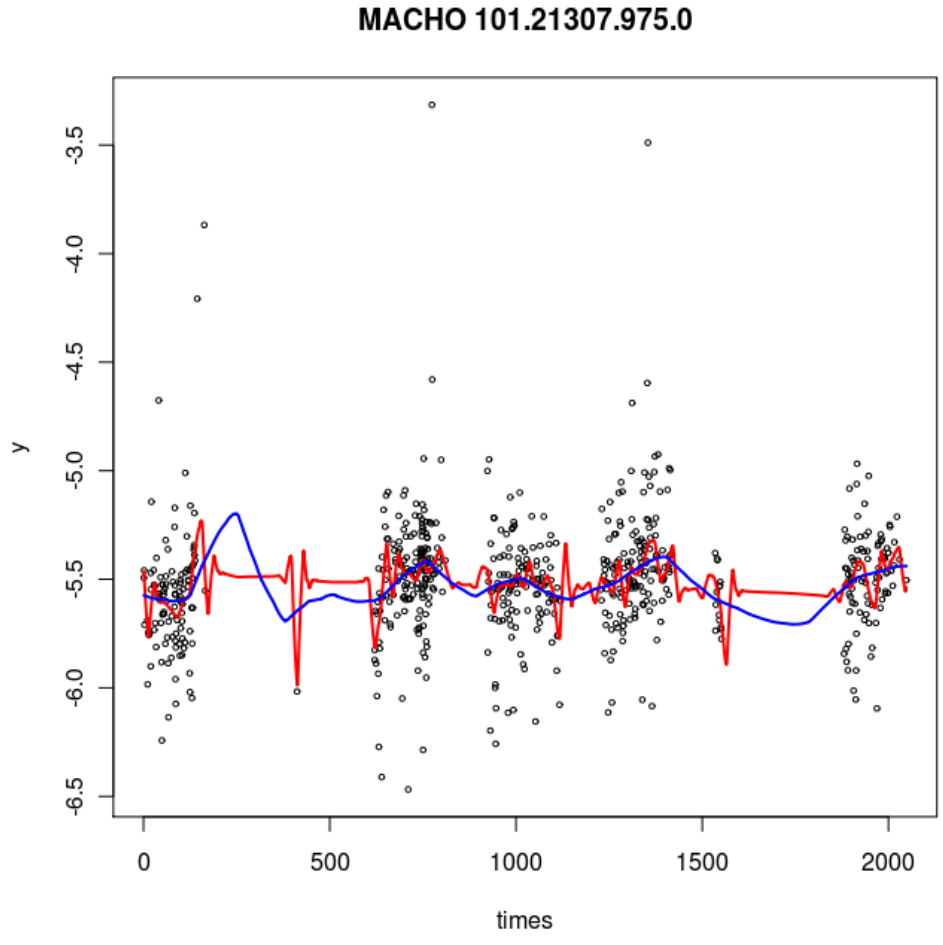
Examples of model fit

The idea is that, if there is an event at the scale of interest, there will be a large discrepancy between the residuals using X_m and X_ℓ:



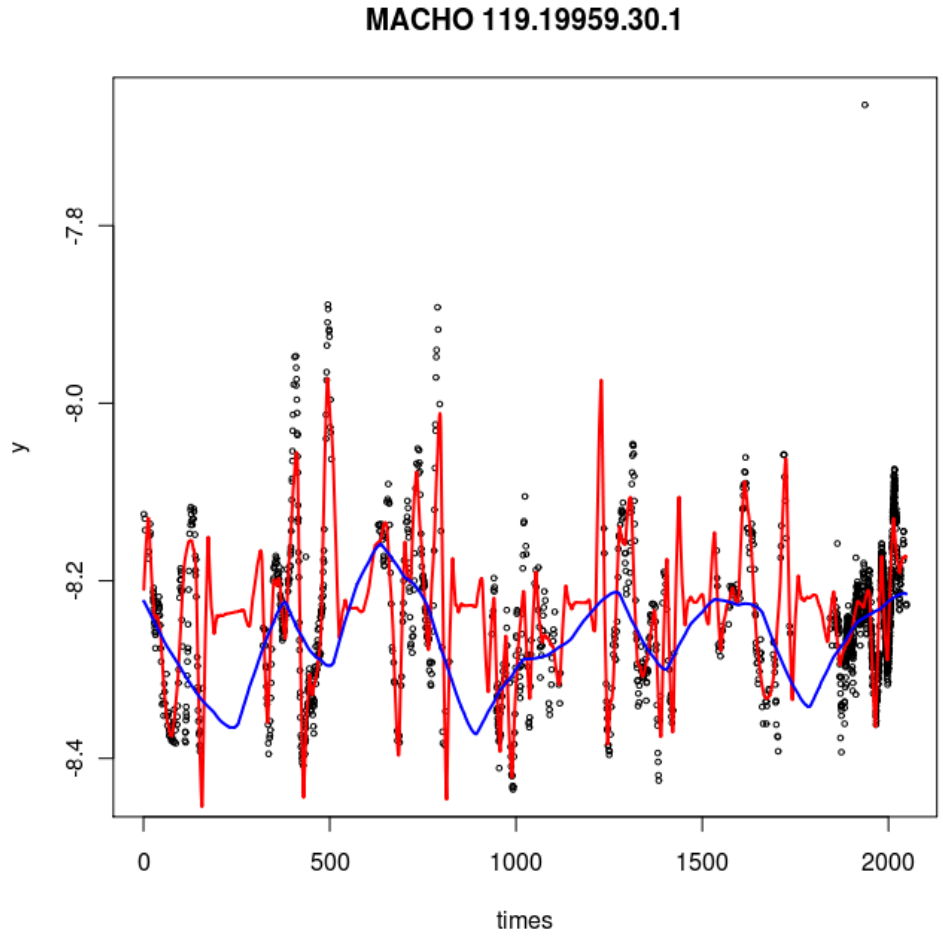
Example of model fit

For null time series, the discrepency will be small:



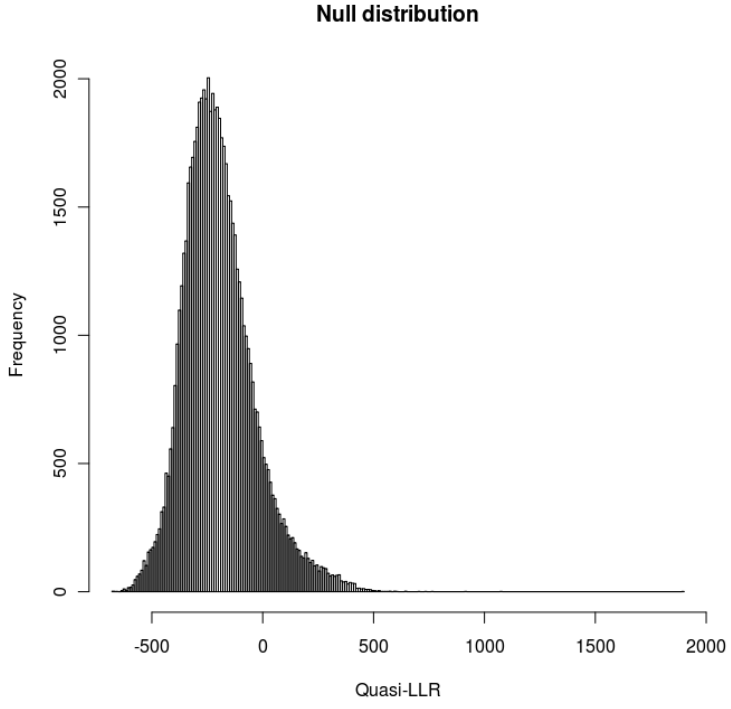
Example of model fit

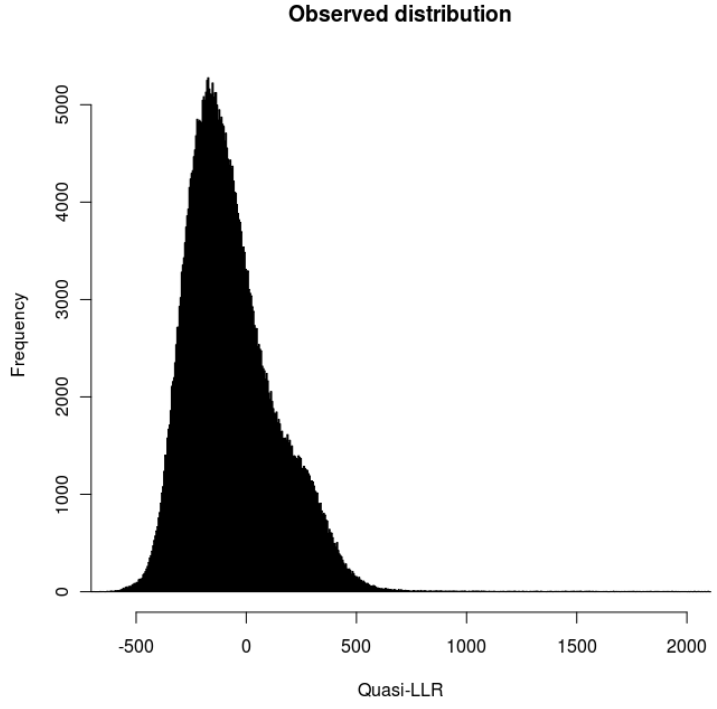
And for quasi-periodic time series, the discrepency will be huge:



Empirical distributions

- Fortunately, the empirical distributions of the quasi-LLR have some separation:





Testing and classification

- Using the Benjamini-Hochberg FDR procedure with an FDR of 0.1 and the empirical distribution of the null series, we identify 7542 nonnull time series based on our wavelet model.
- We then feed the estimated wavelet coefficients for the scales of interest ($\hat{\beta}_m$, normalized and transformed) into a KNN classifier to separate quasi-periodic and isolated events
- Another option that we are currently exploring is to treat null series as another category in the KNN classifier (using far more series)

Classification

- The features used by the KNN classifier are not the full wavelet coefficients. Instead, we use the sorted absolute values of $\hat{\beta}_m$.
- This helps to reduce issues of dimensionality without losing too much information
- We train the classifier on approximately 2000 exemplar time series from each category and set k via cross validation (we obtained $k = 10$ as optimal with our training data)

Classification

- We obtain the below confusion matrix for our classifier on the training set:

		Predicted	
		event	variable
Actual	event	1123	92
	variable	134	1912

- The corresponding error rates are:

		Predicted	
		event	variable
Actual	event	0.92	0.08
	variable	0.07	0.93

Results

- Using this approach we obtain 4,029 time series classified as isolated events and 3,513 time series classified as variable sources in our sample of 515,326 time series
- We are currently pursuing follow-up on these series
- We are also focusing on improving the testing component of this procedure

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
- 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
- There is a vast amount of future research to be done in this area (i.e. I have a lot of work ahead)

Thanks!

- Questions?
- Comments?
- And, of course, many thanks to both Pavlos Protopapas and Xiao-Li Meng for their advice and data