Abstract

The analysis of extremely large, complex datasets is becoming an increasingly important task in the analysis of scientific data. This trend is especially prevalent in astronomy, as large-scale surveys such as SDSS, Pan-STARRS, and the LSST deliver (or promise to deliver) terabytes of data per night. While both the statistics and machine-learning communities have offered approaches to these problems, neither has produced a completely satisfactory approach. Working in the context of event detection for the MACHO LMC data, I will present an approach that combines much of the power of Bayesian probability modeling with the efficiency and scalability typically associated with more ad-hoc machine learning approaches. This provides both rigorous assessments of uncertainty and improved statistical efficiency on a dataset containing approximately 20 million sources and 40 million individual time series. I will also discuss how this framework could be extended to related problems.
Doing Right By Massive Data:
Using Probability Modeling To Advance The Analysis Of Huge Astronomical Datasets

Alexander W Blocker

17 April, 2010
Outline

1. Challenges of Massive Data
2. Combining approaches
3. Application: Event Detection for Astronomical Data
   - Overview
   - Proposed method
     - Probability Model
     - Classification
   - Results
4. Conclusion
What is massive data?

In short, it's data where our favorite methods stop working. We have 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). This scale of data is increasingly common in fields such as astronomy, computational biology, ecology, etc.

There is an acute need for statistical methods that scale to these quantities of data. However, we are faced with a tradeoff between statistical rigor and computational efficiency.
What is massive data?

- In short, it’s data where our favorite methods stop working
What is massive data?

- In short, it’s data where our favorite methods stop working
- We have 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)
What is massive data?

- In short, it’s data where our favorite methods stop working
- We have 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)
- This scale of data is increasingly common in fields such as astronomy, computational biology, ecology, etc.
What is massive data?

- In short, it’s data where our favorite methods stop working
- We have 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)
- This scale of data is increasingly common in fields such as astronomy, computational biology, ecology, etc.
- There is an acute need statistical methods that scale to these quantities of data
What is massive data?

- In short, it’s data where our favorite methods stop working
- We have 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)
- This scale of data is increasingly common in fields such as astronomy, computational biology, ecology, etc.
- There is an acute need statistical methods that scale to these quantities of data
- However, we are faced with a tradeoff between statistical rigor and computational efficiency
Machine Learning methods: strengths & weaknesses, in broad strokes
Machine Learning methods: strengths & weaknesses, in broad strokes

**Strengths:**
- Such methods are typically very computationally efficient and scale well to large datasets.
- They are relatively generic in their applicability.
- Machine learning methods often “just work” (quite well) for tasks such as classification and prediction with clean data.
Machine Learning methods: strengths & weaknesses, in broad strokes

- **Strengths:**
  - Such methods are typically very computationally efficient and scale well to large datasets.
  - They are relatively generic in their applicability.
  - Machine learning methods often "just work" (quite well) for tasks such as classification and prediction with clean data.

- **Weaknesses:**
  - ML methods do not usually provide built-in assessments of uncertainties.
  - A lack of application-specific modeling often means that data is not used as efficiently as possible.
  - Machine learning methods are typically unprincipled from a statistical perspective.
Statistical methods / Probability models: strengths & weaknesses, in broader strokes
Statistical methods / Probability models: strengths & weaknesses, in broader strokes

- **Strengths:**
  - These methods are built upon on sound theoretical principles
  - We can build complex probability models appropriate to the particular application, incorporating detailed scientific knowledge
  - Statistical methods can provide rigorous, built-in 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)
  - While application-specific modeling can be a great strength of this approach, complex structure in the data can require an infeasibly large amount of case-specific modeling
  - Computation for these models often does not parallelize well (for example, MCMC methods are inherently sequential to a large extent)
Statistical methods / Probability models: strengths & weaknesses, in broader strokes

- **Strengths:**
  - These methods are built upon on sound theoretical principles
  - We can build complex probability models appropriate to the particular application, incorporating detailed scientific knowledge
  - Statistical methods can provide rigorous, built-in 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)
  - While application-specific modeling can be a great strength of this approach, complex structure in the data can require an infeasibly large amount of case-specific modeling
  - Computation for these models often does not parallelize well (for example, MCMC methods are inherently sequential to a large extent)
How can we get the best of both worlds?
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.
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.)
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.
Overview

The Problem

- We have a massive database of time series (approximately 40 million) from the MACHO project (these cover the LMC for several years)
Overview

The Problem

- We have a massive database of time series (approximately 40 million) from the MACHO project (these cover the LMC for several years)
- Our goal is to identify and classify time series containing events
Overview

The Problem

- We have a massive database of time series (approximately 40 million) from the MACHO project (these cover the LMC for several years)
- Our goal is to identify and classify time series containing events
- How do we define an event?
The Problem

- We have a massive database of time series (approximately 40 million) from the MACHO project (these cover the LMC for several years)
- 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 and require more refined approaches.
Overview

The Problem

- We have a massive database of time series (approximately 40 million) from the MACHO project (these cover the LMC for several years)
- 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 and require more refined approaches.
  - We are looking for groups of observations that differ significantly from those nearby (ie, "bumps" and "spikes")
The Problem

- We have a massive database of time series (approximately 40 million) from the MACHO project (these cover the LMC for several years)
- 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 and require more refined approaches.
  - We are looking for groups of observations that differ significantly from those nearby (ie, “bumps” and “spikes”)
  - We are also attempting to distinguish periodic and quasi-periodic time series from isolated events, as they have very different scientific interpretations
Overview

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):
Overview

Exemplar time series from the MACHO project:

A variable time series (blue star):
Notable properties of this data
Overview

Notable properties of this data

- Fat-tailed measurement errors
Overview

Notable properties of this data

- **Fat-tailed measurement errors**
  - These are common in astronomical data, especially from ground-based telescopes (atmospheric fluctuations are not kind to statisticians)
Overview

Notable properties of this data

- Fat-tailed measurement errors
  - These are common in astronomical data, especially from ground-based telescopes (atmospheric fluctuations are not kind to statisticians)
  - Thus, we need more sophisticated models for the data than standard Gaussian approaches
Overview

Notable properties of this data

- **Fat-tailed measurement errors**
  - These are common in astronomical data, especially from ground-based telescopes (atmospheric fluctuations are not kind to statisticians)
  - Thus, we need more sophisticated models for the data than standard Gaussian approaches
- **Quasi-periodic and other variable sources**
Overview

Notable properties of this data

- Fat-tailed measurement errors
  - These are common in astronomical data, especially from ground-based telescopes (atmospheric fluctuations are not kind to statisticians)
  - Thus, we need more sophisticated models for the data than standard Gaussian approaches
- Quasi-periodic and other variable sources
  - These changes the problem from binary classification (null vs. event) to $k$-class
Overview

Notable properties of this data

- Fat-tailed measurement errors
  - These are common in astronomical data, especially from ground-based telescopes (atmospheric fluctuations are not kind to statisticians)
  - Thus, we need more sophisticated models for the data than standard Gaussian approaches
- Quasi-periodic and other variable sources
  - These changes the problem from binary classification (null vs. event) to $k$-class
  - So, we need more complex test statistics and classification techniques
Notable properties of this data

- Fat-tailed measurement errors
  - These are common in astronomical data, especially from ground-based telescopes (atmospheric fluctuations are not kind to statisticians)
  - Thus, we need more sophisticated models for the data than standard Gaussian approaches
- Quasi-periodic and other variable sources
  - These changes the problem from binary classification (null vs. event) to $k$-class
  - So, we need more complex test statistics and classification techniques
- Non-linear, low-frequency trends confound our analysis further and make less sophisticated approaches (ie those without careful detrending) far less effective
Notable properties of this data

- **Fat-tailed measurement errors**
  - These are common in astronomical data, especially from ground-based telescopes (atmospheric fluctuations are not kind to statisticians)
  - Thus, we need more sophisticated models for the data than standard Gaussian approaches

- **Quasi-periodic and other variable sources**
  - These changes the problem from binary classification (null vs. event) to $k$-class
  - So, we need more complex test statistics and classification techniques

- **Non-linear, low-frequency trends** confound our analysis further and make less sophisticated approaches (ie those without careful detrending) far less effective

- **Irregular sampling** is the norm in this data. If handled incorrectly, this can create artificial events
Notable properties of this data

- Fat-tailed measurement errors
  - These are common in astronomical data, especially from ground-based telescopes (atmospheric fluctuations are not kind to statisticians)
  - Thus, we need more sophisticated models for the data than standard Gaussian approaches
- Quasi-periodic and other variable sources
  - These changes the problem from binary classification (null vs. event) to $k$-class
  - So, we need more complex test statistics and classification techniques
- Non-linear, low-frequency trends confound our analysis further and make less sophisticated approaches (ie those without careful detrending) far less effective
- Irregular sampling is the norm in this data. If handled incorrectly, this can create artificial events
- Oh my!
Overview

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.
Previous approaches to event detection

- Scan statistics are a common approach (Liang et al., 2004; Preston & Protopapas, 2009)
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
Overview

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
Overview

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
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
Overview

Our approach

We use a Bayesian probability model for both initial detection and to reduce the dimensionality of our data (by retaining posterior summaries). Using these posterior summaries as features, apply a ML classification technique to differentiate between events, variables, and null time series.

Symbolically, let $V$ be the set of all time series with variation at an interesting scale (i.e., 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 will decompose this probability (conceptually) as:

$$P( Y_i \in E ) = P( Y_i \in V ) \cdot P( Y_i \in E | Y_i \in V )$$

using the above two steps.
Overview

Our approach

- We use a Bayesian probability model for both initial detection and to reduce the dimensionality of our data (by retaining posterior summaries)
Our approach

- We use a Bayesian probability model for both initial detection and to reduce the dimensionality of our data (by retaining posterior summaries)
- Using these posterior summaries as features, apply a ML classification technique to differentiate between events, variables, and null time series
Overview

Our approach

- We use a Bayesian probability model for both initial detection and to reduce the dimensionality of our data (by retaining posterior summaries).
- Using these posterior summaries as features, apply a ML classification technique to differentiate between events, variables, and null time series.
- 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.
Overview

Our approach

- We use a Bayesian probability model for both initial detection and to reduce the dimensionality of our data (by retaining posterior summaries)
- Using these posterior summaries as features, apply a ML classification technique to differentiate between events, variables, and null time series
- 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 $P(Y_i \in E)$
Our approach

- We use a Bayesian probability model for both initial detection and to reduce the dimensionality of our data (by retaining posterior summaries)
- Using these posterior summaries as features, apply a ML classification technique to differentiate between events, variables, and null time series
- Symbolically, let $\mathcal{V}$ be the set of all time series with variation at an interesting scale (ie, the range of lengths for events), and let $\mathcal{E}$ be the set of events
- For a given time series $Y_i$, we are interested in $P(Y_i \in \mathcal{E})$
- We will decompose this probability (conceptually) as
  $$P(Y_i \in \mathcal{E}) = P(Y_i \in \mathcal{V}) \cdot P(Y_i \in \mathcal{E} | Y_i \in \mathcal{V})$$
  using the above two steps
Proposed method

Probability model

We assume a linear model for our observations:

\[ Y = X_\ell^T \beta_\ell + X_m^T \beta_m + u \]

We assume that our residuals \( u_t \) are distributed as iid \( t_\nu \) random variables to account for extreme residuals (we set \( \nu = 3 \)).

\( X_\ell \) 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 \( X_\ell \) to contain the first 8 coefficients; \( X_m \) contains the next 120.

Idea: \( X_\ell \) will model structure due to trends, and \( X_m \) will model structure at the scales of interest for events.
Proposed method

Probability model

- We assume a linear model for our observations:

\[ Y = X_\ell \beta_\ell + X_m \beta_m + u \]
Proposed method

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 \( \nu = 3 \)).
Proposed method

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 \( \nu = 3 \)).

- \( X_\ell \) contains the low-frequency components of a wavelet basis, and \( X_m \) contains the mid-frequency components.
Proposed method

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 \( \nu = 3 \)).

- \( X_\ell \) 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.
Proposed method

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 \( \nu = 3 \)).

- \( X_\ell \) 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 \( X_\ell \) to contain the first 8 coefficients; \( X_m \) contains the next 120

- Idea: \( X_\ell \) will model structure due to trends, and \( X_m \) will model structure at the scales of interest for events
Probability model

\[ Y = X_\ell \beta_\ell + X_m \beta_m + u \]
Proposed method

Probability model

\[ Y = X_l \beta_l + X_m \beta_m + u \]

- We explicitly account for irregular sampling in our time series by stretching our basis to total observation time of our data and
Probability model

\[ Y = X_\ell \beta_\ell + X_m \beta_m + u \]

- We explicitly account for irregular sampling in our time series by stretching our basis to total observation time of our data and
- We place independent Gaussian priors on all coefficients except for the intercept to reflect prior knowledge and regularize estimates in undersampled regions.
Proposed method

**Probability model**

\[ Y = X_\ell \beta_\ell + X_m \beta_m + u \]

- We explicitly account for irregular sampling in our time series by stretching our basis to total observation time of our data and
- 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 \( \approx 0.4 \) seconds including file I/O, using the speedglm package in R)
Proposed method

Probability model

\[ Y = X_\ell \beta_\ell + X_m \beta_m + u \]

- We explicitly account for irregular sampling in our time series by stretching our basis to total observation time of our data and
- 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 \( \approx 0.4 \) seconds including file I/O, using the speedglm package in R)
- We use a likelihood ratio statistic to test for the presence of variation at the scales of interest (testing \( \beta_m = 0 \)). We use a modified Benjamini-Hochberg FDR procedure to set the
Examples of model fit
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 $X_\ell$:
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 $X_\ell$:
Example of model fit
Proposed method

Example of model fit

For null time series, the discrepancy will be small:
For null time series, the discrepancy will be small:
Proposed method

Example of model fit

And for quasi-periodic time series, the discrepancy will be huge:
Example of model fit

And for quasi-periodic time series, the discrepancy will be huge:
Example of model fit

And for quasi-periodic time series, the discrepancy will be huge:
Proposed method

Results of likelihood ratio test via FDR

- Awaiting completion of computations
Proposed method

Distribution of likelihood ratio statistic
Proposed method

Distribution of likelihood ratio statistic

To assess how well this statistic performs, we simulated 50,000 events from a physics-based model and 50,000 null time series.
Proposed method

Distribution of likelihood ratio statistic
Distribution of likelihood ratio statistic

- We then added approximately 60,000 time series from known variable stars

- It should be noted that there is an extremely long right tail on the distribution of log-likelihood ratios for variable sources (extending out to approximately 8,000) that is not shown here; it is why additional steps are needed
A sidenote: Why not use a Bayes factor?
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.
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.
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.
Proposed method

Distribution of Bayes factor
Proposed method

Classification

We use the estimated wavelet coefficients \( \hat{\beta}_m \) (normalized by \( \sqrt{\hat{\tau}} \)) as features for classification. These provide a rich, clean representation of each time series, following detrending and denoising (from our MAP estimation). To simplify our classification and make our features invariant to the location of variation in our time series, we use as features the sorted absolute values of our normalized wavelet coefficients within each resolution level.
Proposed method

Classification

- We use the estimated wavelet coefficients $\hat{\beta}_m$ (normalized by $\sqrt{\hat{\tau}}$) as features for classification.
Proposed method

Classification

- We use the estimated wavelet coefficients $\hat{\beta}_m$ (normalized by $\sqrt{\hat{\tau}}$) as features for classification.

- These provide a rich, clean representation of each time series, following detrending and denoising (from our MAP estimation).
Proposed method

Classification

- We use the estimated wavelet coefficients $\hat{\beta}_m$ (normalized by $\sqrt{\hat{r}}$) as features for classification.
- These provide a rich, clean representation of each time series, following detrending and denoising (from our MAP estimation).
- To simplify our classification and make our features invariant to the location of variation in our time series, we use as features the sorted absolute values of our normalized wavelet coefficients within each resolution level.
Proposed method

Classification

We tested a wide variety of classifiers on our training data, including kNN, SVM, LDA, QDA, and others. In the end, regularized logistic regression appeared to be the best technique.

We obtained excellent performance (AUC = 0.98) on previous training data for the separation of ROC for logistic regression classifier (10-fold CV).
Classification

- We tested a wide variety of classifiers on our training data, including kNN, SVM, LDA, QDA, and others. In the end, regularized logistic regression appeared to be the best technique.
Proposed method

Classification

- We tested a wide variety of classifiers on our training data, including kNN, SVM, LDA, QDA, and others. In the end, regularized logistic regression appeared to be the best technique.

- We obtained excellent performance ($AUC = 0.98$) on previous training data for the separation of
Proposed method

Classification

- For the multiclass problem (null vs. event vs. variable), we are testing three approaches: partially ordered logistic regression, multinomial regression, and SVM
- Results are currently awaiting further computation
Computation has yet to complete, but the empirical distribution of our likelihood ratio statistics (with the 10% FDR threshold) is given below:
Putting everything in its place: a mental meta-algorithm
Putting everything in its place: a mental meta-algorithm

- Understand what your full (computationally infeasible) statistical model is; this should guides the rest of your decision
Putting everything in its place: a mental meta-algorithm

- Understand what your full (computationally infeasible) statistical model is; this should guide the rest of your decision
- Preprocess to remove the “chaff”, when possible
Putting everything in its place: a mental meta-algorithm

- Understand what your full (computationally infeasible) statistical model is; this should guide the rest of your decision
- Preprocess to remove the “chaff”, when possible
  - Be careful! Any prescreening must be extremely conservative to avoid significantly biasing your results
Putting everything in its place: a mental meta-algorithm

- Understand what your full (computationally infeasible) statistical model is; this should guide the rest of your decision
- 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
Putting everything in its place: a mental meta-algorithm

- Understand what your full (computationally infeasible) statistical model is; this should guide the rest of your decision
- 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 or for mild regularization if each observation is sufficiently rich or priors are sufficiently informative
Putting everything in its place: a mental meta-algorithm

- Understand what your full (computationally infeasible) statistical model is; this should guide the rest of your decision
- 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 or for mild regularization if each observation is sufficiently rich or priors are sufficiently informative
  - Otherwise, a random subsample of the data can be used to obtain reasonable estimates
Putting everything in its place: a mental meta-algorithm
Putting everything in its place: a mental meta-algorithm

- Using estimates from your probability model as inputs, apply machine learning methods as needed (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.
Putting everything in its place: a mental meta-algorithm

- Using estimates from your probability model as inputs, apply machine learning methods as needed (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.
Putting everything in its place: a mental meta-algorithm

- Using estimates from your probability model as inputs, apply machine learning methods as needed (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.
Putting everything in its place: a mental meta-algorithm

- Using estimates from your probability model as inputs, apply machine learning methods as needed (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
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. There is a vast amount of future research to be done in this areas.
Summary

- Massive data presents a new set of challenges to statisticians that many of our standard tools are not well-suited to address.
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.

There is a vast amount of future research to be done in this area.
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.
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.
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

There is a vast amount of future research to be done in these areas.
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.
- There is a vast amount of future research to be done in this area.
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