Summer 7-14-2018

Data-Driven Uncertainty Quantification Interpretation with High Density Regions

Matthew Gregor Peterson

Candidate: Matthew G. Peterson

Department: Computer Science

This dissertation is approved, and it is acceptable quality and form for publication:

Approved by the Dissertation Committee:

Trilce Estrada, Chairperson

Patrick Bridges

Shuang Luan

David Stracuzzi

Data-Driven Uncertainty Quantification Interpretation with High Density Regions

by

Matthew Gregor Peterson

B.S., Mathematics, Gonzaga University, 2012
B.S., Computer Science, Gonzaga University, 2012
M.S., Computer Science, University of New Mexico, 2015

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

July, 2018

Dedication

To my family

Acknowledgments

I would like to thank my advisor, Professor Trilce Estrada, for her support.

Data-Driven Uncertainty Quantification Interpretation with High Density Regions

by

Matthew Gregor Peterson

B.S., Mathematics, Gonzaga University, 2012

B.S., Computer Science, Gonzaga University, 2012
M.S., Computer Science, University of New Mexico, 2015

Ph.D., Computer Science, University of New Mexico, 2018

Abstract

In a time when data is being constantly generated by phones, vehicles, sensor networks, social media, etc. detecting anomalies with in the data can be very crucial. In cases where we know little prior knowledge about the data, it becomes difficult to extract uncertainty about our results. In this thesis, we will propose a framework in which we can extract uncertainty distributions from data-driven modeling problems. We will show some concrete examples of how to apply framework and provide some insight into what the uncertainty distributions are telling us using High Density Regions (HDRs).

Contents

List of Figures

Chapter 1

Introduction

One of the fastest growing research areas in recent years is data science. With the development of faster, cheaper computers, as well as cheaper memory storage devices, data has never been as easy to come by. So easy that in some cases analysts cannot keep up with the incoming data. By the time we finish analyzing, let alone verifying one data set, a new one is coming through the door. With the pressure to finish all of our analyses as quickly as possible we tend to over look the quality of our results. Instead of doing in depth uncertainty analyses we go with the simple approach of summary statistics, which can be very misleading.

Matejka and Fitzmaurice [42] made an argument that we cannot understand our data with calculations alone; we need to visualize our results with graphs. They came up with an example of multiple plots on a 2D plane that all can be described with the same summary statistics, but they all look vastly different (as shown in Fig. 1.1). With uncertainties we give a plus or minus value to our results creating a uniform error window around each result; when in reality this window of error should be a complex probability distribution around the result indicating where most likely result is for the true answer.

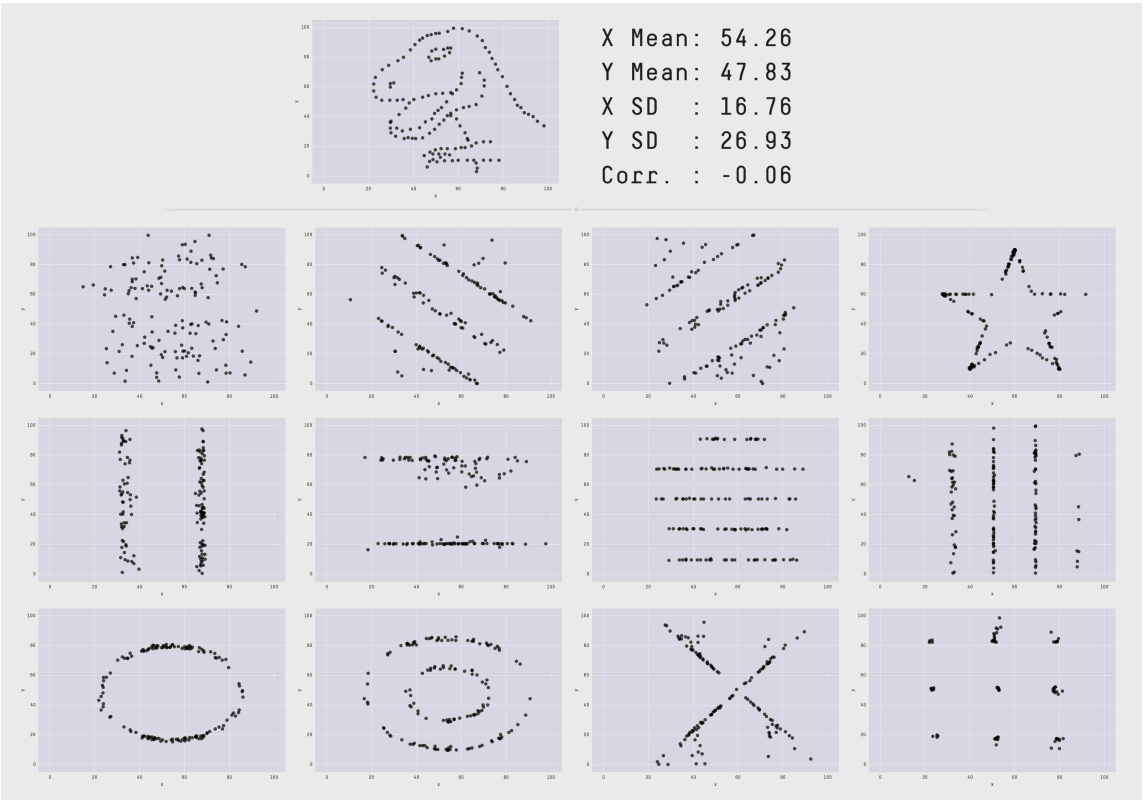


Figure 1.1: Multiple plots of points on a 2D plane all having the same summary statistics, making the point that we cannot trust calculations alone [42].

One area of study that depends on uncertainty is data-driven modeling for time series. This is an important area of study since its applications range from cleaning noise out of data to detecting adversarial attacks on a computer network. In real world applications knowing how certain our models are can be crucial to business operations.

We believe that uniform error is not enough and that we should be visualizing the probability distributions over the data. In this thesis we will provide a framework to extract uncertainty distributions from data-driven modeling methods and how to interpret the extracted distributions. This will be demonstrated using three different problems; Seismic Onset Detection, Arctic Tipping Points, and Protein Folding

Simulation Stability.

1.1 Thesis Statement

To better understand the confidence of our models, we need to interpret their uncertainty. We accomplish this through the use of High Density Region analyses over distributions of model outputs. In order to prove this thesis I propose a framework to assess that:

1. Uncertainty Quantification can provide a more detailed metric of the model's performance compared to simple summary statistics.
2. The mode and shape of the uncertainty distribution are important for interpretation.
    (a) Modes indicate the most likely solution, as well as multiple solutions.
    (b) Shape indicates a level of confidence in the solution.

1.2 My Contributions

The validity of my thesis will be proven by:

1. Presenting a framework that we can apply to any data-driven model for time series to extract uncertainty distributions.
2. Offer insight about how to interpret the uncertainty distributions with High Density Region analysis.
3. Applying the proposed methodology to multiple problems:

(a) Seismic Signal Onset Detection

(b) Arctic Tipping Points

(c) Protein Folding Simulation Stability

1.3 List of Publications

1. Charlie Vollmer, **Matt Peterson**, David Stracuzzi, and Maximillian Chen; *Using Data-Driven Uncertainty Quantification to Support Decision Making*; **Statistical Data Science 2017**.

2. David J. Stracuzzi, Michael C. Darling, Maximillian G. Chen, and **Matthew G. Peterson**; *Data-Driven Uncertainty Quantification for Multi-Sensor Analytics*; **SPIE 2018**

3. Maximillian Chen, Michael Darling, Charlie Vollmer, **Matthew Peterson**, and David Stracuzzi; *Using Uncertainty to Understand Machine Learning Models and Decisions*, **UQ-SciML 2018**

4. Timothy J. Draelos, **Matthew G. Peterson**, Hunter A. Knox, Benjamin J. Lawry, Kristin E. Phillips-Alonge, Abra E. Ziegler, Eric P. Chael, Christopher J. Young, and Aleksandra Faust; *Dynamic Tuning of Seismic Signal Detector Trigger Levels for Local Networks*; **BSSA 2018**

5. **Matt Peterson**, Hunter Knox, Eric Chael, Ben Lawry, Kristen Phillips-Alogne, Aleksandra Faust, Christopher Young, Timothy Draelos; *Self-Tuning Seismic Sensors: Real-Time Trigger Level Adjustments*; **ICASSP 2017**.

1.4 Thesis Organization

- Chapter 2 is an summary of the background knowledge for this dissertation.
- Chapter 3 presents a framework to extract uncertainty from inverse problems (such as machine learning models).
- Chapter 4 presents an example of how to quantify the uncertainty of detecting the onset time of a seismic signal within the presence of background noise.
- Chapter 5 presents an example of how we can use UQ to predict tipping events in the Arctic.
- Chapter 6 presents an example of how to determine if our protein folding simulation is in a stable configuration while the simulation is happening.
- Chapter 7 summarizes our thoughts and conclusions.

Chapter 2

Related Work

Uncertainty quantification (UQ) is the process of determining the likelihood of a result when the variables of the system are not completely defined. This kind of analysis is often needed in physics simulations where it may be costly to do real world experiments and we need to know how uncertain the results from the simulation are. These uncertainties stem from three distinct types of error described by Le Maître [35]; *model errors, numerical errors, and data errors.*

Model errors are a result of the mathematical equations describing the model not being able to completely capture all the aspects of the physical system. This is largely due to the fact that the mathematical equations are simplified to fit the problem space. There are even cases where parts of the physical system are completely ignored; these parts of the system usually have negligible contribution to the calculations. All in all, the equations that form the model are an approximation for the physical system which people deemed "good enough" through some validation techniques, such as variance analysis or comparing to the physical system's results.

Numerical errors are due to the resolution of the mathematical equations of the model. These equations often call for discretization techniques and algorithms which

makes the results an approximation of the true answer in the physical system. These kinds of errors can be reduced by using a finer discretization; for example, fine grain spatial meshes or using smaller time steps. This kind of error can also be reduced by imposing a harder convergence criteria for the model.

Data errors come from using data that does not fully capture all the qualities of the physical system. In a variety of cases there are limitations on how well the data describes the real world scenarios. These constraints can be due to a lack of experimental data available, weak boundary conditions, or the shear complexity of the physical system. For example, the input parameters to the system might be a coefficient of friction or a material strength, and the output of the system is the data we expect to observe. If we do not have accurate values for our input parameter, our observable output data will contain error.

All of these types of errors contribute to the uncertainty of the final results. Each of which can be reduced by imposing stricter conditions on the calculations or though extensive research of the physical system. At the end of the day our models are approximations of how the physical system works; and UQ can help us characterize the errors in our model in the form of a likelihood, often portrayed as a probability distribution.

2.1 Types of UQ Problems

There are two types of UQ problems, forward uncertainty propagation and inverse UQ. The forward problem can be visualized as a simulation. We input the parameters of the system into our model and we output the data,

$$Parameters \rightarrow Model \rightarrow Data.$$

While the inverse problem is the opposite. We have observed data as the input

to our model and the output is a prediction of the parameters of the system,

$$Parameters \leftarrow Model \leftarrow Data.$$

The inverse problem can be visualized as a machine learning problem, such as classification prediction. This is discussed more in Chapter 3.

In regards to UQ, we know more about the forward problem and how to propagate uncertainty, compared to the inverse problem [as in 36], and can provide great insight into the errors produced by the model. Those same insights can be extracted from the inverse problem, but there are some unresolved issues that arise. First of which is the dimensionality costs. The computational costs increase rapidly with the increased number of dimensions. Generating all of those samples and running all of them through a model can be time consuming; on a positive note, the process is embarrassingly parallel since running each sample through the model is independent of each other. Another issue is identifiability. This is when multiple combinations of the unknowns of the system can yield the same predictive result [discussed in 3].

Because of these issues UQ for the inverse problem is largely ignored in the machine learning community. Instead of providing detailed uncertainties to validate the performance of our machine learning models, we fall back on using accuracy and precision. This can be problematic is cases where the test data is not labeled or there is minimal amounts of data to test against. Chapter 3 will discuss the importance of UQ for machine learning as well as its advantages in real world scenarios.

UQ for the inverse problem is still in its early stages of research. One of the most common ways to extract uncertainty from inverse problems is with a *Bayesian approach*. The basic idea revolves around *Baye's rule*,

$$\mathbb{P}(u|y) \propto \mathbb{P}(y|u)\mathbb{P}(u)$$

where u is the input/parameters to a model and y is the observable data. In words,

the *posterior* $\mathbb{P}(u|y)$ is what we want to find out about unknown inputs u given the data y, and that is proportional to the *likelihood* $\mathbb{P}(y|u)$ multiplied by the *prior* $\mathbb{P}(u)$.

The big issue with this method is that it is heavily dependent on knowing the correct prior distribution for the given model, as raised by Inglesias and Stuart [26, 27]. For certain subject areas this prior distribution does not exist or is very hard to get. In Chapter 3 we will discuss *Bootstrapping* which is a way that do not involve directly using a prior distribution and instead focus a data-driven methodology. Most importantly we will also discuss a way to interpret the posterior distributions for decision making purposes in later chapters.

2.2 Anomaly Detection

Anomaly detection is the act of finding patterns within data that differs from normal behavior. These unique parts of the data are refereed to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities, or contaminants depending on the field of study [11].

Anomalies are important to data science because they could mean a variety of different things; such as, noise in the data that should be removed or malicious activity on a system, for example, cyber-attacks or credit card fraud. All of which can be categorized into are three types of anomalies according to Chandola [11]; *Point anomalies, Contextual Anomalies, and Collective Anomalies.*

Point anomalies are most basic, or simplest form of anomaly. They are individual instances of the data that are considered dissimilar to the rest of the data. An example of this would be in credit card fraud. If the common transactions are less than some value v and a new transaction t_1 comes along greater than v, we would deem that a point anomaly.

Contextual anomalies are very similar to point anomalies. The key difference is that they have some context component associated with the data (typically temporal [56, 65] or spatial [34, 58]). Continuing with the credit fraud example, lets say that my common transactions are less than v, except in July when I go on vacation I spend more than v. A new transaction t_2 arrives with a value greater than v occurring in September. This transaction t_2 would be considered a temporal contextual anomaly since I only spend more than v in July.

Collective anomalies are a group of related data instances that may not be anomalous on an individual level, but when combined as a group they are considered anomalous. Lets assume that some businessman travels between Los Angeles and New York fairly often, so it would not be uncommon to see transactions from those two cities. Now lets consider two transactions; t_3 occurred in Los Angeles at 1:00pm and t_4 occurred in New York at 1:05pm. These transactions alone are not unusual, but together they are a collective anomaly since it is not possible for the businessman to travel across the country in five minutes. Collective anomalies are commonly studied in sequence data [14, 64], graph data [48], and spatial data [58].

Change Point Detection is determining when the pattern in a sequence changes; which can be viewed as an instance of the collective anomaly problem. The groups in this case are sequences of a stochastic process or time series. Examples of change detection problems include edge detection [40], wavelet analysis [33, 47], and seismology [54, 63]. A detailed example of extracting uncertainty from a seismic change detection problem is discussed in Chapter 4.

2.3 How Others Interpret UQ

In a lot of cases, when statisticians talk about uncertainty they typically mean that there models account for error or that they plotted error bars around their solution.

In a paper done by Sebastian et al they create a machine learning model for automated text categorization [57]. They account for error in their models and when they present their results it is done with simple error bars on a graph; a plus/minus window around the mean. Brohand et al wrote a paper titled "Uncertainty estimates in regional and global observed temperature changes: A new data set from 1850" [8], in this paper they show some results with uncertainty. They show that by incrementally adding in different sources of measurement errors which increases their uncertainty, and is visualized by 95% confidence intervals around the mean. Bertozzi et al have a paper titled "Uncertainty quantification in graph-based classification of high dimensional data" [5] where they visualize their posterior distributions as histograms and summarize them with a mean value of the distribution. In some of the cases the distributions are bimodal and a simple mean will not capture what is fully happening.

In my experience with uncertainty bases papers and conferences, most of the people are statisticians who account for error in their models and display their results with error bars. What I would like to see if a deeper dive into interpreting their distributions because that is what is missing for decision making analysts. Their is a gap between what the statisticians do and what analysts need. In Chapters 4, 5, and 6 we will discuss specific decision making problems and how we can improve the current approaches by interpreting the uncertainty distributions with High Density Regions (brief background in Sec. 2.4) rather than with simple error bars.

2.4 High Density Regions

High Density Regions (HDRs) are a helpful tool when it comes to interpreting your data, which is how what this dissertation is going to be focusing on. HDRs point highlight a given range of your data distribution. In the case of a likelihood distri-

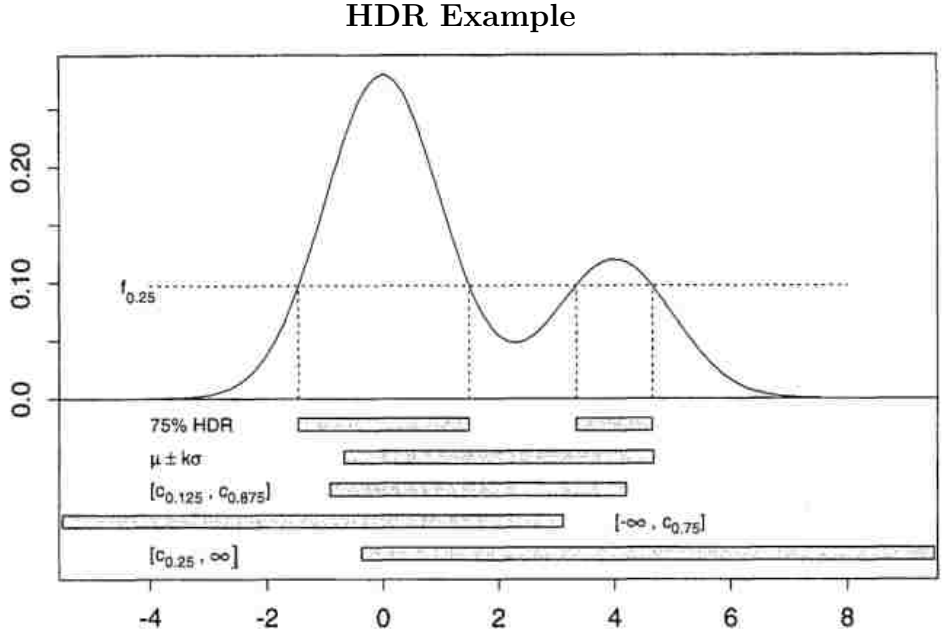


Figure 2.1: Here is an example of how HDRs differ from other known methods; such as mean plus/minus standard deviation and confidence intervals. HDRs can have split ranges while the other methods can only have a single range. Multiple ranges are important when the distribution contains multiple, distinct modes. Image from Hyndman [25].

bution, it points out the areas of the distribution that contain the most likelihood. The main goal behind HDRs is to find a set of ranges that contain a specific amount of area under the curve using the smallest amount of volume [25]; in our case a curve is a likelihood distribution.

One of the main benefits of using HDRs is that it does not have any assumptions about the shape of the distribution. In Fig. 2.1 we see how HDRs can handle a bimodal distribution better than mean plus/minus a standard deviation, as well as confidence intervals. The other methods work great when there is only a single mode, but will be less meaningful when there are multiple modes. In later chapters we will show how we use HDRs to interpret our data. For more information on how to calculate and graph HDRs check out Hyndman [25].

Chapter 3

Proposed Method

In most real world applications the results are typically in the form of a single answer plus or minus a uniform error; this is an oversimplified version of *uncertainty quantification* (UQ). True UQ involves a probability distribution for its answer. In this dissertation we provide a framework in which we can extract the uncertainty from modeling based methods and provide a way to interpret the results. This process of extracting uncertainties and interpreting will offer great insight and details about what our models are really telling us.

Fig. 3.1 shows how machine learning problems fit within the inverse UQ problem. At each step in the pipeline there are error distributions that can be evaluated and later combined into a single distribution. In this thesis we will focus on perturbing the indirect observations in order to capture the *measurement errors* and *model form uncertainties*.

Extracting data error uncertainties from modeling problems are difficult in real world scenarios. In some cases we have very few examples of real world data to build our models; with so little examples we cannot rely on past knowledge to characterize the uncertainty. Because of this reason we will use a *Monte Carlo* or *bootstrapping*

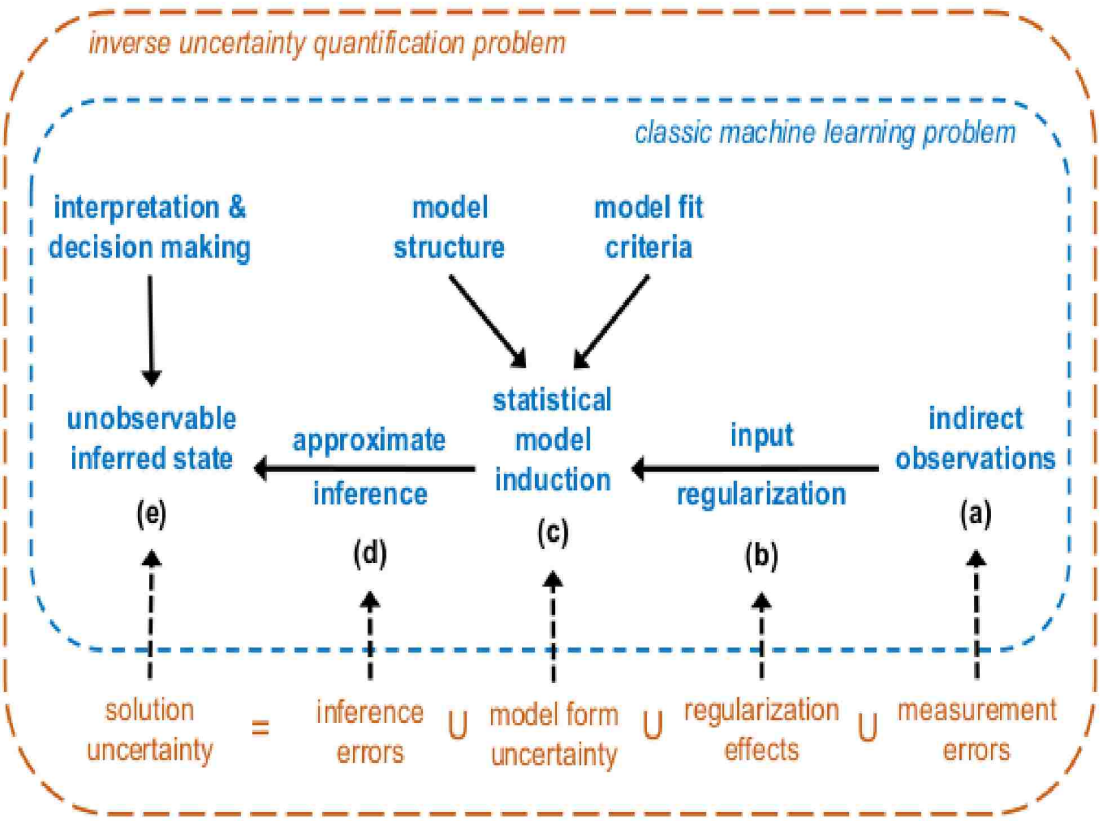


Figure 3.1: Flow chart of the inverse uncertainty quantification problem [18].

method to generate samples by perturbing the input data (discussed in further detail in Section 3.1) in order to alleviate the reliance on prior knowledge.

The essence of the framework is to create many generated samples based off of the original data, run them all through the model to get varying results, and then save them all into a distribution to show the likelihood of all the results. Fig. 3.2 shows a quick overview of the flow of the problem.

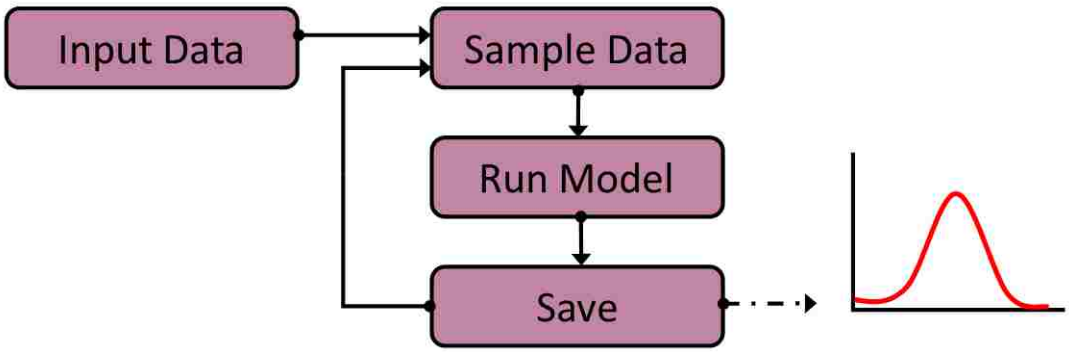


Figure 3.2: The general framework for extracting the data error uncertainty distribution from modeling problems.

3.1 How is it Done

In order to produce a probability distribution of our results we need our framework to run many samples over the domain of our problem space, typically on the order of 1,000 or 10,000. This is typically achieved using one of two methods, *Monte Carlo (MC) or Bootstrapping*, each having its own unique take on how to generate samples.

When the domain of our system follows a known distribution we can use a Monte Carlo method [31, 43] (specifically MC not and not MCMC) where we randomly draw samples from the known probability distribution until we obtained our desired number of samples. These kinds of sampling methods are used in a wide variety of fields; such as, physical sciences [24, 29] [39], climate change [61], computational biology [44, 49], and artificial intelligence [9, 13] to name a few.

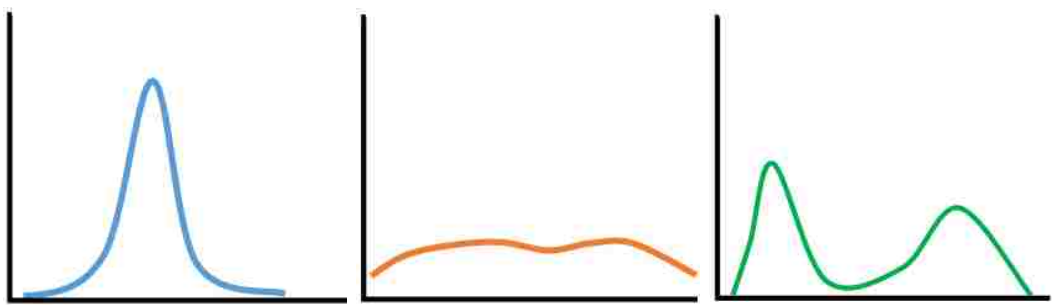
If we do not know the probability distribution of our domain, then we use a Bootstrapping method [20, 21] which is a statistical method that uses random sampling with replacement. In the most general form we are given a single sample of size N. We then take that original sample and sample from that N times with replacement; this is often referred to as a 'bootstrap sample.' We would repeat this process until

we have generated the desired number of samples. Other more complex versions include Bayesian Bootstrap [55], Smooth Bootstrap [15], Parametric Bootstrap [37], and non-Parametric Bootstrap [59].

Both Monte Carlo and Bootstrapping follow the same pipeline to create a probability distribution for its result; draw/generate a new sample, run it through your model, save the result, and repeat until satisfied. The advantage to Bootstrapping is that you do not need to fully understand the domain of the system.

3.2 What do the Distributions Tell Us?

Once we have the distributions from our uncertainty extraction we can gleam some insight as to where the true answer lies. The most likely locations of our answer are the modes, or peaks, of the probability distributions. Fig. 3.3 shows three examples of what we might see.



(a) tight (b) wide (c) multiple

Figure 3.3: Plots of three different kinds of probability distributions.

Fig. 3.3a is a very tight distribution which indicates there is one very likely point for our answer. Fig. 3.3b is a wide distribution which means that the method/model we used was unable to find an answer confidently. Fig. 3.3c has two distinct modes which indicates that are two likely answers occurring in this data. So not only can

uncertainty distributions tell us what the answer might be, but also if there are multiple likely answers.

We can use *High Density Regions* (HDRs) to help us with this process of identifying the characteristics of of our distributions. Using metrics like mean and stand deviation can help us with determining whether our distribution is tight or wide, but it assumes a normal distribution. We want to allow multiple modes, various skewness, and different kurtosis values; and HDRs can handle all of these situation. HDRs does not have any assumptions about the shape of the distribution, all they care about is where does most of the area lie under the curve. With the information given by the HDRs we can determine the most likely results with minimal assumptions about the shape of the distribution. Knowing the most likely results helps with the decision making process. We will discuss specific ways of using the HDR results in later chapters.

Chapter 4

UQ for Seismic Onset Detection

Precisely determining a change point is no easy task, especially when is comes to detecting seismic signals of interest. A seismic station records all kinds of ground based noise ranging from the general background hum of the earth to cars driving on a nearby road, all of which hide our signal of interest (which may be an earthquake or a man-made event such as a mining explosion). By using data-driven Uncertainty Quantification we can get a more accurate reading of where the true change point may occur in difficult situations when the signal may be buried with in the noise.

Currently human analysts are the gold standard for detecting seismic signals even though there are known errors and biases with humans. Depending on the analyst's experience level and the standard practices within their organization, some analysts pick earlier than others. Zeiler [66] conducted a study where they gave multiple analysts, varying in experience and organizations, the same set of waveforms and asked them to determine the onset time for each waveform. They found both variance in onset determinations among analysts, and variance within a single analysts' predictions when given the same waveform at different times.

The goal of this chapter is to demonstrate a way to automate the generation of

the uncertainty in the seismic signal onset so that we can provide better detail to the analysts during the verification phase, and so that we can improve downstream analyses; such as determining location and event type. In the remainder of the chapter, we will provide some further background on picking arrival times for seismic signals. Then we will discuss our proposed method for extracting uncertainty. After that, we will demonstrate our method on an example from the SPEAR dataset that was collected by Zeiler [66]. Finally we will conclude with a discussion of the work and our future work.

4.1 Background

Seismic stations record movement through the earth at the location of the station. A typical station consists of three channels corresponding to different axes of movement; a vertical axis, an east-west axis, and a north-south axis. All of these channels have different time series data represented as a waveform. The goal of an analyst, or automated method, is to determine when a signal of interest occurs on a given waveform.

The change point from noise to signal in seismology is often referred to as the *first arrival time* or *detection onset time*. Determining first arrivals is done through a combination of automated methods and human analysts. There is a first pass with an automated method to locate possible first arrivals over the entire data set. Then an analyst goes through all the candidate detections to verify and refine the onset times, as well as find missed detections. These data sets can be quite large having thousands of detections over multiple days and numerous stations. The precision of these first arrival times is incredibly important to downstream analyses. For long distance events such as earthquakes, changing the detection time by 0.1 seconds can move the estimated location by tens of kilometers, which in turn can impact the

event's type classification and other characteristics [53].

The most common automated method is *Short-Term Average over Long-Term Average* (STA/LTA). STA/LTA calculates takes the average energy of a short window length (such as 1 sec) and the average energy of a much longer trailing window (such as 60 secs), and computes the ratio. If the ratio of the short window to the longer window is greater than some specified threshold, that point is declared a possible onset time. These windows slide across the entire dataset determining possible onset times (see Rodriquez [54] for more details).

Automated detections are further refined by using autoregressive models and a fitting metric, such as *Akaike Information Criterion* (AIC; described in [1]). Two separate models, $\mathcal{M}_1$ and $\mathcal{M}_2$, are used for the noise and signal portions of the window respectively, as shown in Figure 4.1; and are typically pre-built based on past physics knowledge. AIC is then used to optimize the onset time as the point at which the two models meet. Several variations on this procedure are described by Kamigaichi [32]. Importantly, the uncertainty of the onset time is typically described only by a confidence interval calculated as a function of the signal-to-noise ratio.

In the approach described in the rest of this chapter we will show that we can extract uncertainty distributions for the estimated arrival time using a parametric bootstrap approach as well as build our model on the fly, instead of assuming a one size fits all scenario for the model. With these improvements we can show in greater detail of where the first arrival time may occur.

4.2 Method

The input for the method assumes we are given a window of a zero-meaned, filtered seismic waveform that contains noise followed by a signal where an initial guess for

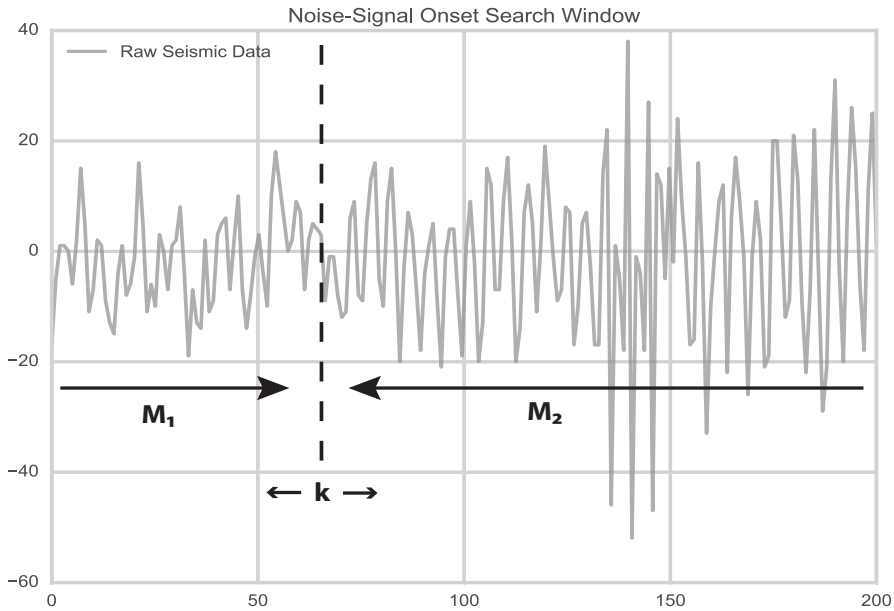


Figure 4.1: Sliding window for refining the estimated arrival time. $\mathcal{M}_1$ is the model used for the noise, $\mathcal{M}_2$ is the model used for the signal, and k is the change point from $\mathcal{M}_1$ to $\mathcal{M}_2$. We slide k through the search space to find the best fit for $\mathcal{M}_1$ and $\mathcal{M}_2$.

the onset detection, given by a STA/LTA picker (see Sec. 4.1 for more details), is centered in the middle of the window. This allows the algorithm to assume that the left half of the window contains mostly noise and the right half contains mostly signal; which is important when fitting the *Autoregressive* (AR) model [45] in later steps. In other experiments we attempted fitting both the noise and the signal halves, but fitting the signal on the fly was unreliable [12].

The first step is to pre-process the data for the algorithm. For this method this means squaring the waveform, which is a benefit for a couple of reasons. The first reason is that it moves the data from the raw-waveform space into the amplitude-over-time space. One of the biggest indicators of an incoming signal is the change in amplitude, as seen in Fig. 4.3. Typically the noise level is small compared to the signal. The second reason for squaring the data is to amplify the signal-to-noise ratio

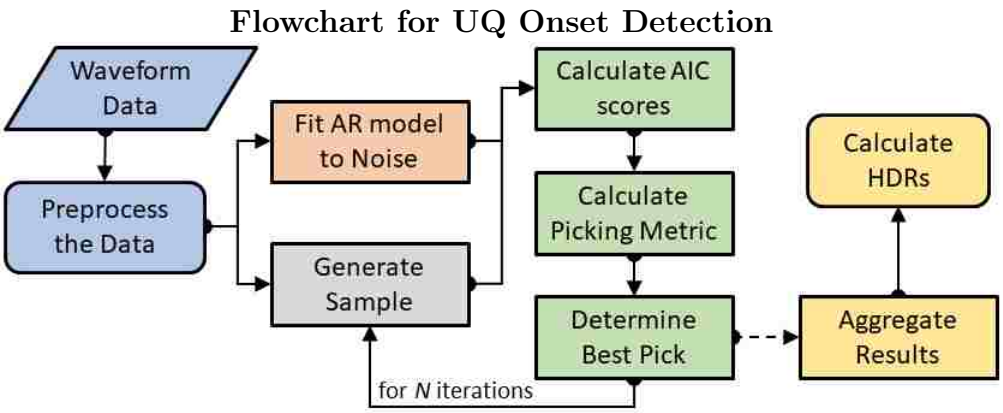


Figure 4.2: Flow chart for creating probability distributions for seismic onset detections.

(SNR) so that smaller signals stand out more against the background noise levels.

The rest of this section will explain the necessary steps to create a probability distribution for the pick uncertainty.

4.2.1 Fitting the Noise Model

The overall goal is to determine when the signal begins, or in other words, when does the noise stop. This method focuses on fitting an AR model to the noise and determining at which point in time does the noise model stop fitting the data. Unlike the method described in Sec. 4.1, we focus only on fitting a noise model and ignore creating a model for the signal. This is partly due to the fact that the signal fitting process is unreliable as well as redundant to the information that can be extracted from the noise.

The first step is to fit the AR model to the noise data. Since one of our assumptions is that the left half of the window is contains primarily noise, we can fit our AR model for the noise to the left half of the window, as seen in Fig. 4.4. Since all waveforms are slightly different we loops over multiple lag lengths for the AR model

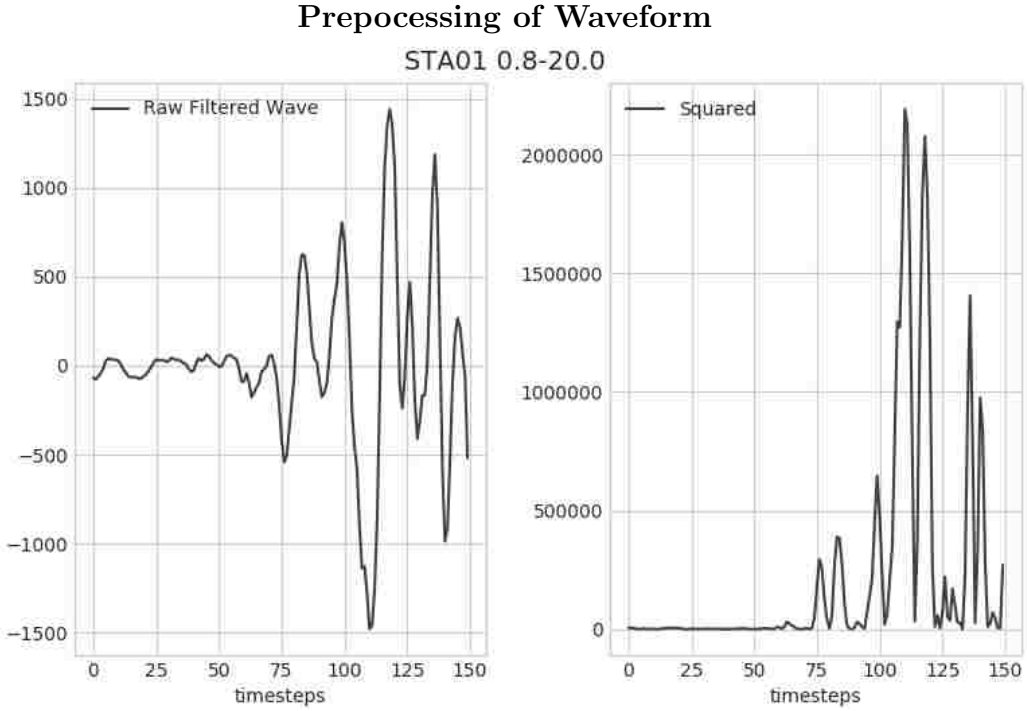


Figure 4.3: The graph on the left is the the zero-meaned waveform with a 0.8-20.0 Hz bandpass filter. The graph on the right is the squared version of the same filtered waveform.

for the noise and determine which is the best fit using *Akaike Information Criterion* (AIC) [1].

Calculating AIC

AIC is a metric to determine how well a model fits a set of data. The two major factors for influencing an AIC score are the number of free parameters, k, being fit in the model and the log-likelihood, llh, of the model fitting the data, as seen below.

$$AIC = 2k - 2(llh) \tag{4.1}$$

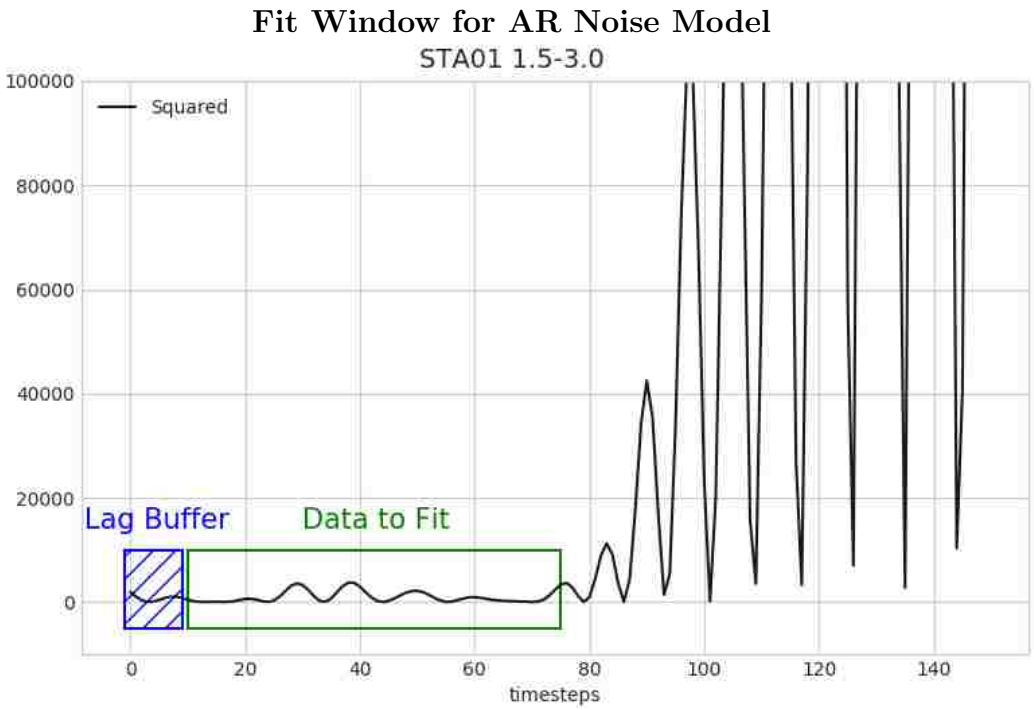


Figure 4.4: Data to fit an AR model to capture the statistical properties of the noise.

The log-likelihood for an AR model is calculated as follows,

$$llh = -\frac{n * ln(2\pi\sigma^2)}{2} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2 \tag{4.2}$$

where σ^2 is the error from the model, $\hat{x}_i$ is i-th data point, and x_i is the i-th data point from the AR model.

4.2.2 Calculating AIC Curves

Once we have and AR noise model fitted to the data, we can now create an *AIC Score Curve* to help us determine at which point the waveform starts to become dominated by an incoming seismic signal. In order to do this we take a sliding window across the entire waveform to determine how well our AR noise model fits the data in the window. This process is demonstrated in Fig. 4.5. The lowest score indicates where

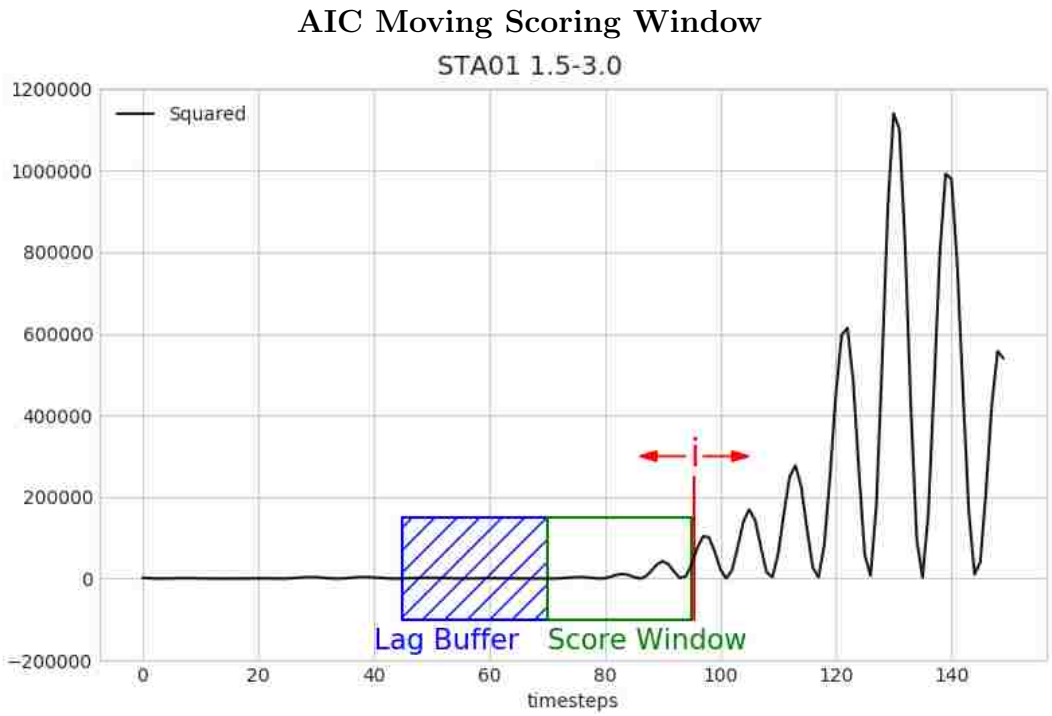


Figure 4.5: For each point i we calculate the AIC score of the AR noise model to the data in the *Score Window* which is the same length as the lag required for the AR noise model.

the model fits the data the best; in other words, at which point the data is for sure noise, but this is not what we are looking for. We want to know where it stops being noise, and this is why we need a picking metric to determine the best solution.

4.2.3 Picking Metric

We cannot use the AIC curves alone. They only show at which points the AR noise model fits the best and worst, not where the signal starts. In order to solve this problem we implemented our own *Picking Metric* that takes into account sudden changes in the slope of the AIC curve as well as our knowledge that the initial onset is more likely to happen earlier than later.

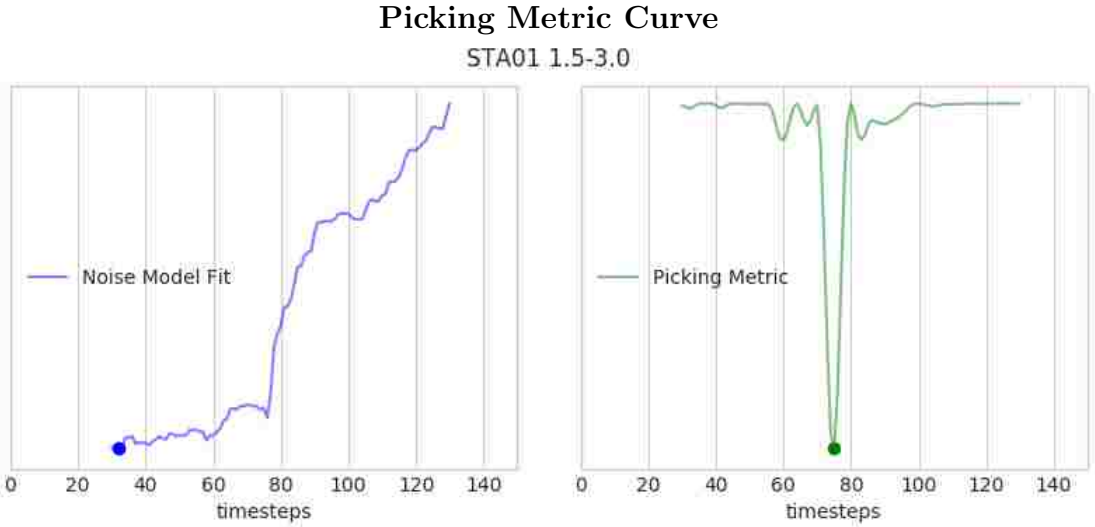


Figure 4.6: On the left is the result of the moving AIC score window, and on the right is the picking metric curve. The dots on each curve indicate the best score for the given curve (i.e. the lowest point on the curve). Note that the best point of the picking metric curve occurs at the same timestep as the sudden change in slope of the noise model fit.

Combing our AIC curve from Sec. 4.2.2, as well as equations 4.4 and 4.5, we can calculate our picking metric for each point i in our search window as follows,

$$PickMetric_i = AIC_i * W_i * (MAX\_SLOPE - Slope_i). \tag{4.3}$$

The point with the lowest $PickMetric$ score is the best pick.

Slope of AIC

Using a pure AIC score can be misleading, as shown in Fig. 4.6. It picks where the AR model fits the noise the best, which is deep within the noise part of the waveform and not at the transition point from noise to signal. On the other hand, the AIC score starts getting increasingly bad as it enters the signal part of the waveform. In the AIC score curve this is indicated by a sudden change in slope, which is calculated as shown below.

$$Slope_i = \left| \frac{AIC_i - AIC_{i-s}}{s} - \frac{AIC_{i+s} - AIC_i}{s} \right| \tag{4.4}$$

where s is the number of timesteps back in which we calculate the slope.

Logistic Weight Curve

Since we are focusing on the first arrival we weight earlier picks as better. Occasionally a sudden change in slope occurs later in the waveform, which could indicate a secondary phase of the signal. We are only interested in the start of the signal and thus we use a logistic weight curve to amplify the score of earlier timesteps and lower the scores of later ones.

The weight curve is a typical logistic curve as follows:

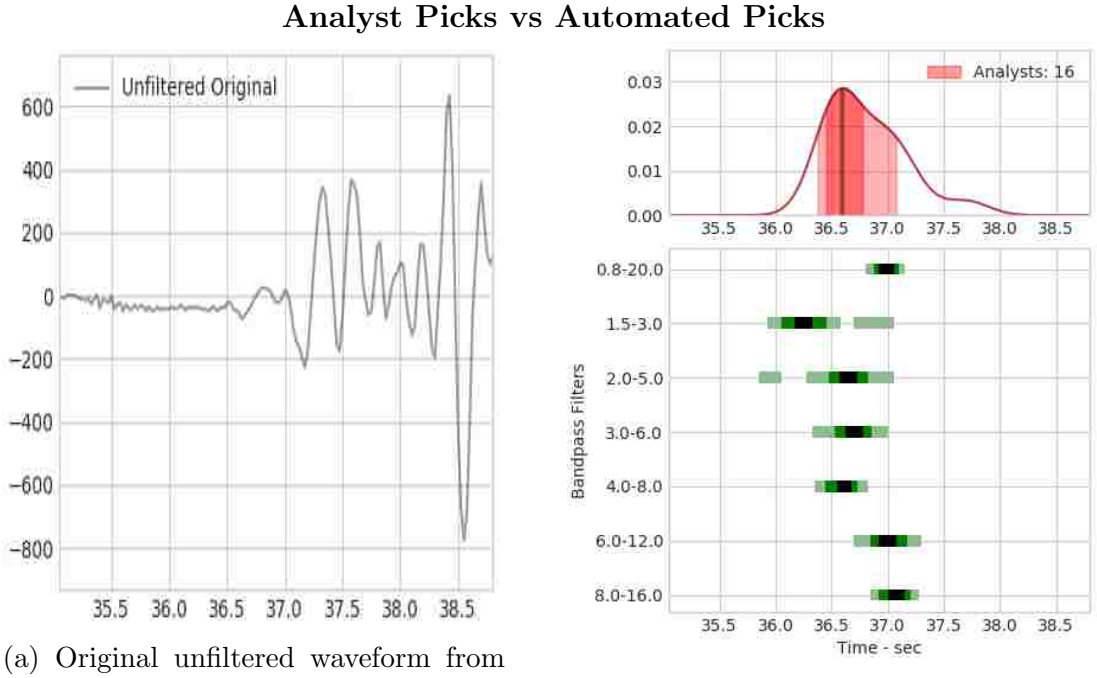
$$W_i = \frac{1}{1 + e^{2/\delta * (i - \delta)}}, \tag{4.5}$$

where $\delta = search\_window\_size/2$.

4.2.4 Generating the Distribution

Creating the distribution is done by sampling the data multiple times and running the picking metric on all of the samples. For each sample we have a best pick based off of the picking metric which can all be aggregated into a probability distribution over time of where the true pick occurs.

The process used for sampling the data is a Gaussian blur like technique. We take fifty percent of the data points and adjust their amplitude by a random draw from a Gaussian distribution defined by the points neighbors. In other words, copy the data and add noise to half of the data points.



(a) Original unfiltered waveform from station 12.

(b) Varying bandpass results.

Figure 4.7: (a) Shows the original unfiltered waveform from STA12. (b) The top distribution in red is generated from the picks made by 16 analysts and shows HDRs: 66%, 33%, and 05%. The graph on the bottom shows the condensed results from the automated method for varying bandpass filters in green with the same HDR values. A more detailed view of the results can be seen in Fig. 4.9.

4.3 Results

To evaluate the quality of our automated picks, we compared the output of our methods to picks made by human analysts using the data collected by Zeiler and Velasco [66]. Note that in the seismic domain, the complexity of the earths subsurface and its impact on wave propagation makes providing ground truth for real events almost impossible. Synthetic waveform generation is also a challenge as it requires highly detailed models, extensive computation, and still may not accurately reflect real-world conditions.

The dataset includes 26 waveforms sampled at 40 Hz, each having 5 to 18 an-

alysts with varying degrees of seismic background knowledge pick the arrivals of seismic events. We applied multiple ranges of bandpass filters that an expert analyst provided for us. For each filtered waveform, we created search windows spanning 3.5 seconds (or 140 samples) centered around the first arrival, as determined by an STA/LTA. The search window length was chosen to fully encompass P-phase of the seismic signal as well as enough time to fully characterize the background noise; in the case of this dataset we found that was a length of 3.5 seconds works well.

In Fig. 4.7 we show the results from our automated method for the vertical channel on station 12 (STA12). Fig. 4.7a shows the original unfiltered waveform. In Fig. 4.7b we have a comparison of the analysts pick distribution and a set of condensed automated pick distributions for varying waveforms. These automated pick distributions were generated with a thousand iterations (we chose 1000 iterations in order to run the method in under 1 minute on a laptop). An import note about this figure is that the majority of bandpass filters agree with the analysts, especially ranges 3.0-6.0 Hz and 4.0-8.0 Hz; we do not have any knowledge of what filter bands the analysts used.

In Fig. 4.9 we show a more detailed view of the automated distribution results, as well as examples of the picking metric and a comparison to a normal distribution for the results. Each row is the result of a specific bandpass filter from a set of filters bands given to us by an expert analyst. The left column is has three things: the filtered waveform in gray, the pre-processed waveform in black, and the results from our automated method represented as a likelihood distribution in green. A detailed explanation of how the curves were calculated is explained in Sec. 4.2.4. The main take-away from this are the shaded regions of the distribution. These shaded region represent *High Density Regions* (HDRs)[25] of 66%, 33%, and 05%. The regions highlight where the most likely pick is.

In the middle column there are two curves: the AIC noise model fit in blue and

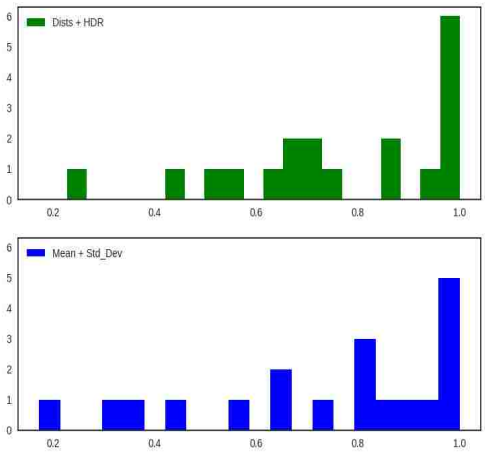
the picking metric in green. For both of these curves the lowest value indicates the best score which is marked with a circle. How these curves were calculated is described in Sec. 4.2.2 and Sec. 4.2.3.

In the right column we show what the result distribution would look like if we assumed our results were normally distributed; in practice, seismic signals are reported as an arrival time and a standard deviation. If we compare these distributions to their counterparts in the left column we notice two import things. Firstly, the HDR regions in the left column are much tighter; one prime example is the 6.0-12.0 Hz bandpass row. A tighter HDR region indicates a higher level of confidence in the results, as well as a smaller search window to find the true onset time. Secondly, the left column offers more detail about the results, specifically if there are multiple modes; an example of this is bandpass row 1.5-3.0 Hz.

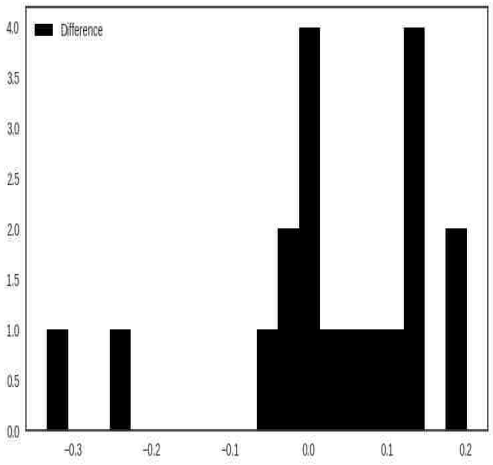
We also directly compared to how much better the HDR analysis does compared to just using a mean and standard deviation. We used the 66% HDR ranges from the analyst distributions as ground truth. We then calculated the range of values using the mean+stddev from our automated distribution as well as the range of values for 66% HDR of the same distribution. In Fig. 4.3 we have the results of the overlaps. Fig. 4.8a shows the percent of the ranges that are fully within our ground truth ranges; the top shows the HDR overlap and the bottom shows the mean+stddev overlap. The overlaps look fairly similar from this point of view, but if we take the difference in overlap on a per waveform basis we see that the HDR analysis performs better.

In Fig. 4.8b we show the difference in overlap on a per waveform basis. Positive values indicate that he HDR analysis had more overlap with the analysts than the mean+stddev. In the figure we see that the majority of the results are positive, which means our HDR analysis does a better job than a straight mean+stddev analysis.

Comparing to Analysts



(a) Percent of overlap with the analysts. Top is "Dist+HDR", bottom is "Mean+Standard Deviation."



(b) Difference in analyst overlap per waveform. Positive indicates "Dists+HDR" had more overlap with the analysts.

4.4 Conclusion

In this chapter we illustrated a viable way to automate uncertainty for change point detection of a seismic signal onset time. This added information can greatly improve analyst pick consistency and refinement speed by showing statistical evidence of where the true onset time may occur. We also demonstrated a way to interpret this added information by using HDRs, as well as the advantages over assuming the results are normally distributed. In the future we would like to be able to identify multiple phases of the signal, automate the bandpass filtering portion, as well as propagate the uncertainty to downstream analyses.

The techniques used in this chapter have been around in the statistics community for a long time, but they are not used in the hands of analysts who have to make crucial decisions. There is a gap between statistics and real world analysts which results in doing simple summary statistics and making assumptions about the data. As data scientists we need to show how these techniques can be used and explain

how to interpret the results, which is what this chapter shows.

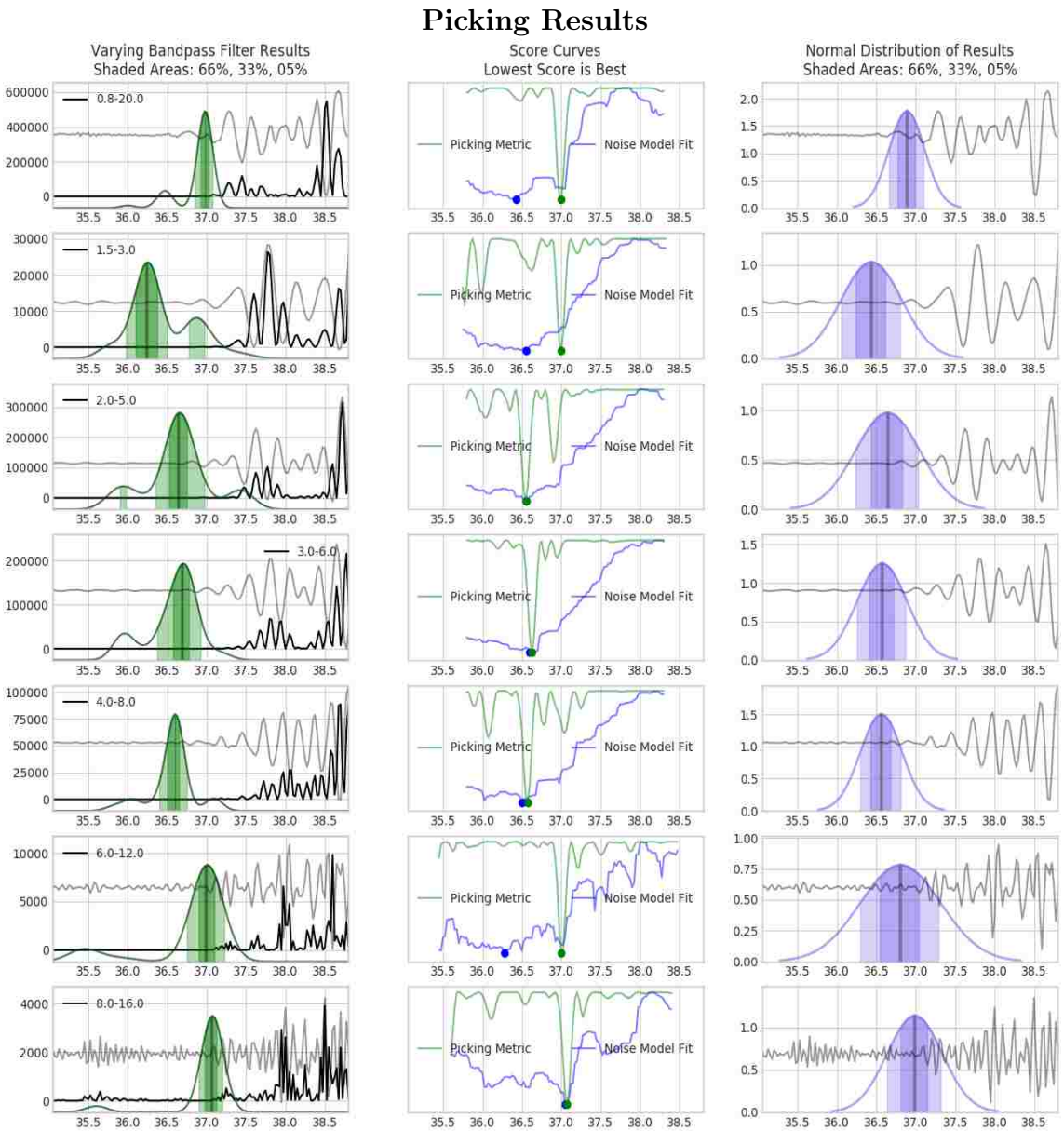


Figure 4.9: Each row represents the results for a specific bandpass; from top to bottom we have 0.8-20.0 Hz, 1.5-3.0 Hz, 2.0-5.0 Hz, 3.0-6.0 Hz, 4.0-8.0 Hz, 6.0-12.0 Hz, and 8.0-16.0 Hz. The Left column shows the distribution results in green with varying sized HDRs, the original filtered waveform in gray, and preprocessed input in black. The Middle column shows the AIC model fit for the noise in blue and the picking metric in green (lower scores are better). The Right column shows what the distribution results would look like if we assumed they were normally distributed instead of no assumptions of the shape (like the leftmost column).

Chapter 5

UQ for Arctic Tipping Points

The Arctic systems are a vital part of our Earth climate system, and in recent history is has started to change rapidly. According to the Arctic Report Card 2017 [28], "Arctic shows no signs of returning to the reliably frozen region it was a decade ago." The changes in the Arctic can cause various changes globally. The shutdown of *Atlantic thermohaline circulation* (THC) [51] can cause regional cooling and significant weather impacts in the northern hemisphere, which in turn can lead to food scarcity. The Greenland ice sheet melting [68] would increase sea level and alter ocean currents, which would lead to risks of flooding to coastal infrastructures. The loss of sea ice [41] would change mid-latitude weather causing damage from weather events (such as hurricanes). The permafrost thawing [16] would release a significant amount of greenhouse gases into the atmosphere leading to more thawing of the permafrost, as well as climate change-based instability. We need better prediction models than we have currently. Arctic changes are happening faster than current models have predicted, and there is evidence that the models are biased towards stability [19].

Data from ice core samples has shown instability patterns before known as *Dansgaard-Oeschger* (D-O) events [7, 22] (see Sec. 5.1 for more details). These are

events where the temperature in the Arctic rapidly increased over a short period of time. These events are characterized as tipping points because they result in a large change in the system's state due to a positive feedback loop in the internal dynamics accelerating state change [17].

In the rest of this chapter we will show how we can determine the likelihood that a time series is entering a tipping before the event happens using a combination of *Autoregressive* (AR) models [45] and *High Density Regions* (HDRs) [25]. We will demonstrate this method by detecting D-O events in paleo-temperature data from ice core samples taken from the Greenland ice sheet. This problem is a type of change point analysis where we can only see past data points, unlike the seismic problem seen in Chapter 4 where we had access to all the data points.

5.1 Background

The data used for this experiment was ice core samples taken from the Greenland ice sheet, specifically site GISP2 (see Fig. 5.1 for a satellite image of the site location). These samples can give us an approximation of the temperature of Greenland's past up to 140,000 years ago at 30 year resolution. This approximation is calculated by calculating the density of oxygen isotopes (δ^{18}O) present in the ice which is proportional to the temperature at that time the ice froze [23]. The most interesting part of the data is that it captures D-O events. A D-O event is characterized by a rapid warming of the Greenland ice sheet over 30-40 years, followed by a cooling period of a couple hundred years. One example occurred about 11,500 years ago, the temperature rise for the Greenland ice sheet was 8 °C over a 40 year period, where usually it is a 5 °C change over 30-40 years [2]. These events are significant because they can change the North Atlantic Ocean circulation [7], which can influence climate on a global scale [10].

Figure 5.1: Satellite image of the location of where the GISP2 ice core samples were taken. Spatial coordinates N: 73.0 S: 72.0 E: -37.0 W: -38.0. Image was taken from Google Earth.

5.2 Method

The goal of this method is determine the likelihood that a tipping event is about to occur. In order to do this we fit an AR model to period of time that we know does not contain a tipping point. Once we have our fitted AR model, for each time point t_i we project the next time point, t_{i+1}, many times (we used 3,000 projections for our experiments) using our fitted AR model. These projections can be aggregated together to form a likelihood distribution over the next possible values in the time

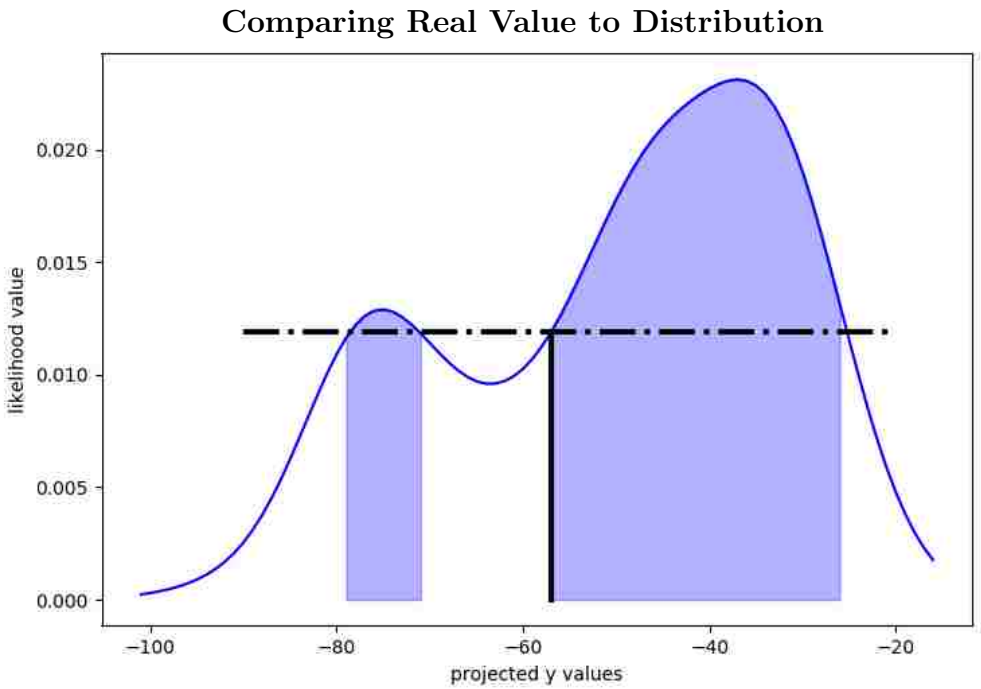


Figure 5.2: Shown above we have an example of how we calculate which HDR time point t_{i+1} falls into. The blue distribution is the likelihood over possible values for time point t_{i+1}. The solid, vertical, black line at $x = -57$ represents the actual value for time point t_{i+1} and the horizontal, dashed, black line at $y = 0.011$ represents the cutoff point to calculate the HDR. In this case the shaded area represents the HDR of 70.14%.

series.

Now that we have our distribution of possible values for time step t_{i+1}, we can compare the actual value for t_{i+1} to our calculated distribution using HDRs. HDRs inform us of smallest intervals that contain a specific percentage of area in the distribution. In Fig. 5.2, we show an example of how we compare the real value to our distribution and retrieve the HDR value. In the example we now that time point $Y(t_{i+1}) = -57$. This gives us HDR intervals of $[(-79, 70), (-57, -25)]$ which corresponds to 70.14% of the distribution. Zero percent indicates a very good match and one hundred percent indicates a very bad match.

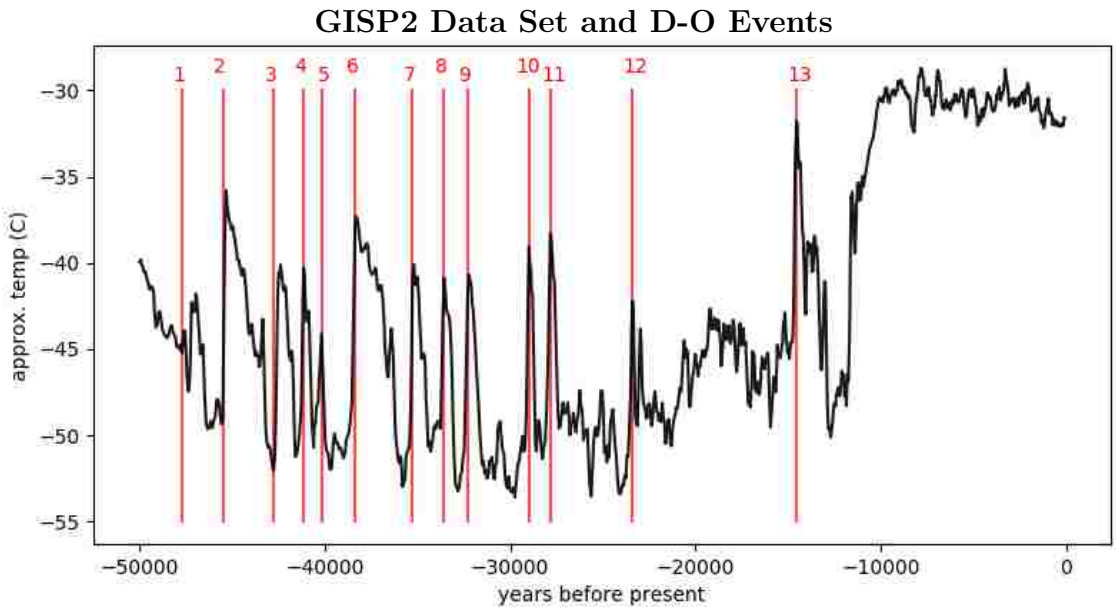


Figure 5.3: A plot of the GISP2 data set (black line) with an overlay of the 13 D-O events present within this data set (red, vertical lines).

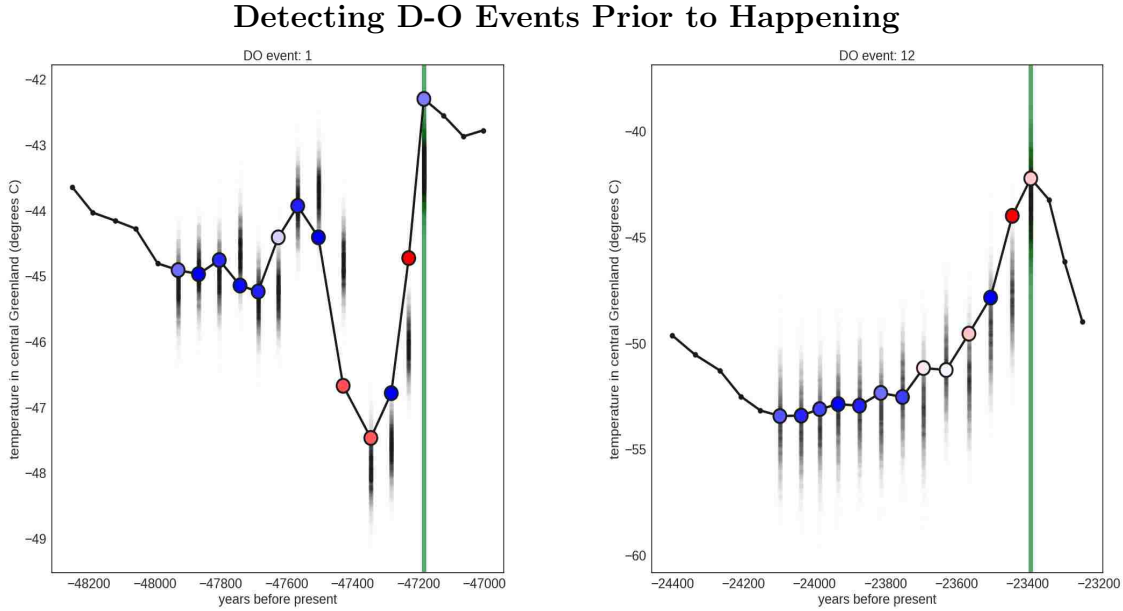
Once we have the HDR percentage we calculate our likelihood score of our time series entering a tipping point. This is done by multiplying our HDR percentage with the difference in slope between t_{i-1} to t_i and t_i to t_{i+1}, like so,

$$Tipping\_Score(t_{i+1}) = HDR(t_{i+1}) * \left| \frac{|Y(t_i) - Y(t_{i-1})|}{t_i - t_{i-1}} - \frac{|Y(t_{i+1}) - Y(t_i)|}{t_{i+1} - t_i} \right|. \quad (5.1)$$

The closer to zero, the less likely the time series is entering a tipping event at time point t_{i+1}.

5.3 Results

Using the process described in Sec. 5.2 we attempted to see if we could detect that our ice core time series data from GISP2 is entering a D-O event before the D-O event actually happened. Our GISP2 data set goes back 50,000 years at a 50 year resolution and contains 13 D-O events (see Fig. 5.3).



(a) Results for the 1st D-O event.

(b) Results for the 12th D-O event.

Figure 5.4: The green vertical line represents the actual D-O event for this time window. The small, black dots on the left most side of each window are the points that were used fitting the AR model. Each colored dot represents a likelihood score that a tipping event is about to occur; red being very likely and blue being very unlikely. The black vertical lines are the predicted next points that are used for calculating the likelihood score; the darker parts of the line indicate more points predicted at that point. In both examples there are distinct red dots before the D-O event occurs, as well as some light blue/red indicators three or more points before the event.

For each of the D-O events we fit our AR model with a maximum lag of eight to a stable period of time before the D-O event, around 15 data points before the D-O event. Then for each point after the stable period we projected the next data point 3,000 times (after 1,000 the results don't change much) to create a probability distribution and compared the real next point to the distribution to create a likelihood score of whether we are about to enter a tipping point or not, as described in 5.2.

In Fig. 5.4 there are two examples of the results. The vertical green line in these figures shows where the D-O event occurs. The red and blue dots represent how

likely the time series is about to enter a tipping event; red indicates very likely and blue indicates very unlikely. In both cases we see that there is a dark red dot right before the D-O event. In addition to right before the event, we also see that there are some lightly colored red dots three or more points before the D-O event. Of the thirteen D-O events that we have, 11 out of the 13 contained non-blue point 2+ data points before the D-O event. This implies we are able to detect early with some confidence.

Another important note is the difference of complexity between the two time series. Fig. 5.4b is a much more traditional D-O event where the temperature steadily increases at a rapid rate. On the other hand, Fig. 5.4a has a more sinusoidal shape where there is a potential false positive. Our process is able to identify the potential false positive as not likely and still be able to detect the true event.

5.4 Conclusion

In this chapter we demonstrated that it is possible to detect a tipping event, specifically a D-O event, before it happens using a combination of AR modeling for projections and HDRs to interpret likelihood of a tipping event. This is import because if we can detect the tipping event before it happens, there is a chance that we can reverse the process and stay in a stable zone. Future work for this project includes converting the method into a sliding window environment and processing the entire data set. This is challenging since it is not clear when to fit, or refit, the AR model. Another improvement for this project is to be able to handle multiple time series from different sources; for example, having ice core samples from multiple sites around Greenland. This addition would help in the reduction of false events since true events would have to correlate across multiple sources.

Chapter 6

UQ for Protein Folding Simulation Stability

Proteins are a sequence of amino acid building blocks that are vital to maintaining life in biological systems. According to Liu et al. [38], "One of the greatest challenges in molecular biology today is that of determining how the sequence of a protein – the exact ordering of amino acids it is composed of – specifies its structure and function." To determine how a protein might fold a simulation is run, but it is very expensive to output the simulation at every time step since there are so many data points. A typical work around is to output the state of the simulation once every n time steps. The issue with this approach is that in between the n time steps the protein may have folded into a stable structure and then unfolded due to a change of energy in the system, hence missing the part we care about.

In the rest of this chapter we will give some background on protein folding simulations. Then we will discuss a way to determine when the simulation is in a stable state using *High Density Regions* (HDRs) [25]. Finally we show some results from a handful of simulations.

6.1 Background

Protein folding simulations search for trajectories leading to conformations close to the native (folded) protein structure originating from an unfolded conformation. During the folding process, the protein changes its conformations into what are called meta-stable and transition stages [4].

In a metastable stage, consecutive protein conformations keep a similar structure and display only small variations. In a transition stage, consecutive protein conformations change from one meta-stable stage to another and exhibit large structural variations. In order to identify these stages, it is important to identify when one or multiple trajectories eventually converge to the same conformation. Work has been done to understand intra-trajectory and inter-trajectory convergence. These studies [6, 50, 62, 67] explore multiple folding trajectory spaces in parallel and determine what conformations are more likely to be stable.

Computational trajectory analysis usually performs a large scale comparison of trajectory frames, constructing a centralized dissimilarity matrix using all the trajectory data, reducing the dimensionality of the matrix, and then clustering the low dimensional matrix. The centralized nature of the algorithms in Best el al. [6] and Phillips et al. [50] makes their analysis inefficient when dealing with large proteins and long trajectories. Other work in [62] analyzes simple statistical data of long trajectories at a very large scale. Our previous work [30, 67] deals with this issue in a local to global fashion, rendering the parallel analysis very efficient for large datasets and is suitable for in-situ analysis.

We used 31 simulated protein folding trajectories from MoDEL, the Molecular Dynamics Extended Library [46]. MoDEL is a large library of molecular dynamics trajectories of representative protein structures. Trajectories of all monomeric soluble structures have been studied by means of state-of-the-art atomistic molec-

ular dynamics simulations in near-physiological conditions. Trajectories used for our analysis range from 2,000 to 20,000 time steps (i.e, number of data points M) and from 58 to 747 residues (i.e., number of dimensions N). Table 6.1 shows their characteristics.

Table 6.1: Characteristics of 31 MoDEL Trajectories

Characteristic	Mean	Stdev	Min	Max
Number of residues	193.06	145.29	58	747
Simulation time (ps)	9,779.03	3,425.85	2,000	20,000

6.1.1 Trajectory analysis

We perform the protein folding trajectory analysis as if it was a clustering problem. Simulations can be performed in parallel, with different nodes taking care of different segments of a trajectory, or, more accurately, different trajectories given particular starting conditions. As simulations progress, in-situ analysis is necessary to determine what conformational spaces have been analyzed and whether the current conformation is stable or transitional. To perform this analysis in parallel, we characterize each conformation (i.e., a specific conformation associated with a trajectory frame) by its collection of secondary structures given the Ramachandran plot [52]. That is, every residue was characterized by the torsion angle phi, ϕ, (angle between the C-N-CA-C atoms) versus the torsion angle psi, ψ, (angle between the N-CA-C-N atoms), and omega ω (usually restricted to be 180 deg for the typical trans case or 0 deg for the rare cis case). Based on the constraints of the torsion angles (ϕ, ψ, and ω) as described by the Ramachandran, we can associate each amino acid residue in the protein with one of six types of secondary structures: α-helix, β-strand, Polyproline PII-helix, γ'-turn, γ-turn, and cis-peptide bonds. As a protein folds and unfolds over time, its residues may participate in very different types of secondary structures, but if conformations are revisited over time, they should cluster together. We

hypothesize that by clustering secondary structures, multiple fine grained clusters associated with specific secondary structure transformations will arise over time. Sequences of fine grained clusters will form a cluster fingerprint. This fingerprint can be used to identify stable phases and to differentiate conformational search spaces.

6.2 Method

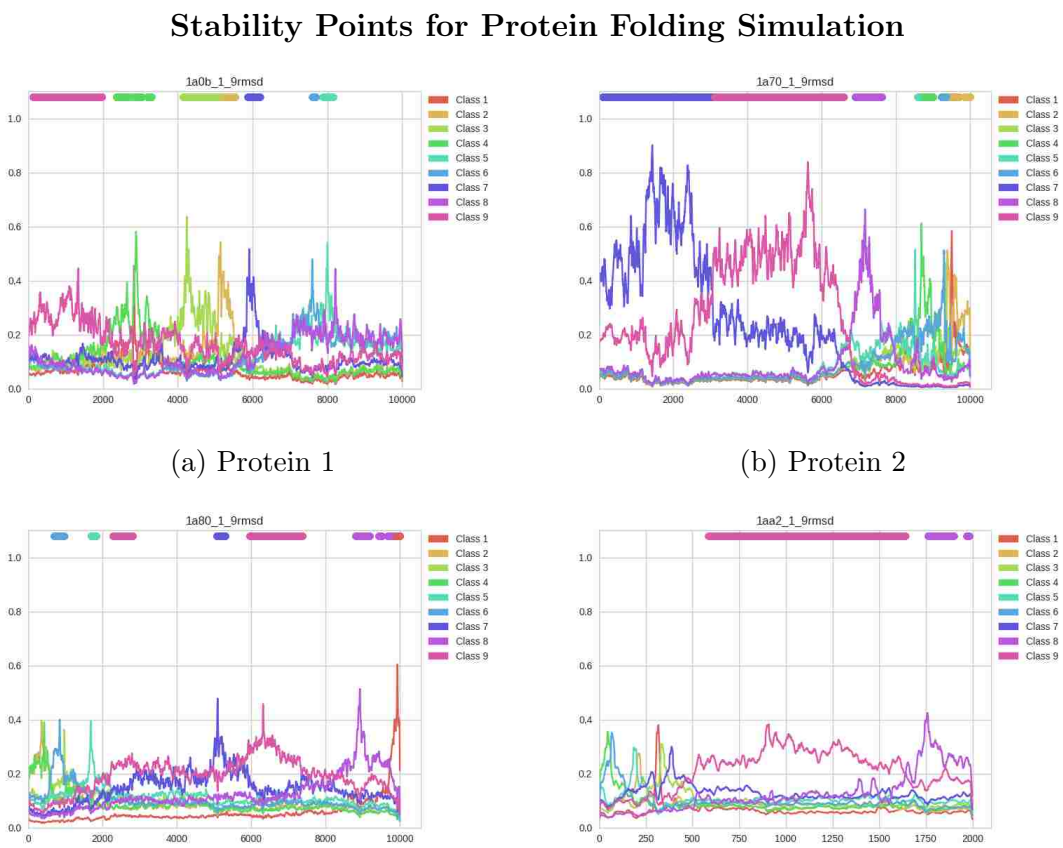
After a trajectory is completed, we selected N distinct conformations (aka. stable protein foldings of interest) sampled by using a power law distribution with respect to the distance to the mean conformation. This setup is designed to find a set of diverse representative conformations along the trajectory. For each of them, we compute the root mean squared deviation with respect to each frame in the trajectory. Given a set of root mean squared deviation time series, we preprocessed the data by converting the distance measures into probabilities that a particular time step (i.e., frame) of the trajectory is a given conformation.

$$Pr(l \; is \; stable \; at \; i | l \in L, d_{k,i} \in D_i) = \frac{1/d_{l,i}}{\sum_{k=1}^{N} 1/d_{k,i}}, \tag{6.1}$$

where L is the set of N distinct conformations and D_i is the set of distance measures for each conformation at frame i. We then create a probability distribution of stability for each representative conformation, which for simplicity we denote as a label, at time step i using the previous 100 time steps. Using the probability distributions, we calculate the center of the 70% *High Density Region* (HDR) for each label. This generates a score of stability ranging from 0 to 1 for each label at time step i, where 1 indicates high label stability and 0 indicates low stability. To determine if frame i is not stable we compare the two highest label stability scores as follows,

$$Stability(s_{p,i}, s_{q,i}, w) = \begin{cases} s_{p,i} - s_{q,i} < w, & not \; stable \\ otherwise, & p \; is \; stable \end{cases}, \tag{6.2}$$

where w is a predefined threshold, $p, q \in L$, $s_{l,i}$ is the stability score of label l at frame i, and label p has the higher stability score at frame i.



(a) Protein 1

(b) Protein 2

(c) Protein 3

(d) Protein 4

Figure 6.1: Shown above are four different protein simulations. The nine time series are the probabilities that a given class is stable, defined in Eq. 6.1, over time for the nine possible classes. The solid lines at the top of each subfigure correspond to stable class over that period of time, defined in Eq. 6.2. The y-axis shows the probability of stability, and the x-axis is the time step.

6.3 Results

Using the method described in Sec. 6.2 we ran our process on thirty different protein folding simulations. In these simulations we wanted to know for each time step whether the protein was in one of nine stable configurations, as well as whether it was not stable. In Fig. 6.1 we show a subset of the results.

For each result we show nine time series of the stability scores at every time step of the simulation. At the top of each subfigure we show which class is stable for that given period of time; blank spots indicate that the time series is not stable at the point in time. We compared these results to a Cross-Correlation matrix [60] that was then clustered, as seen in Fig. 6.2.

In Fig. 6.3 we show two comparisons of the stability labels to the cross-correlation clusters. The top rows are the stability labels and the bottom rows are the cross-correlation clusters over time. The big takeaway here is that the stability labels and the clustering labels transition at similar time frames, as well as the stability labels are blank at the same time as when the clustering labels are switching a lot. In Fig. 6.3a there was 81.4% agreement of when a label should or should not exist, and the areas of disagreement can be attributed to the cross-correlation clusters trying to label areas of transitioning conformations; there were a total of 9 stability labels and 19 cross-correlation clusters. Fig. 6.3b had an agreement of 69.3% with the same number of clusters and labels.

The major advantage with using our method is that it can be done during simulation time while cross-correlation plus clustering is a post-process method that takes a lot of compute time. Also, our methodology only labels conformations of interest and ignores the transition periods between stable conformations.

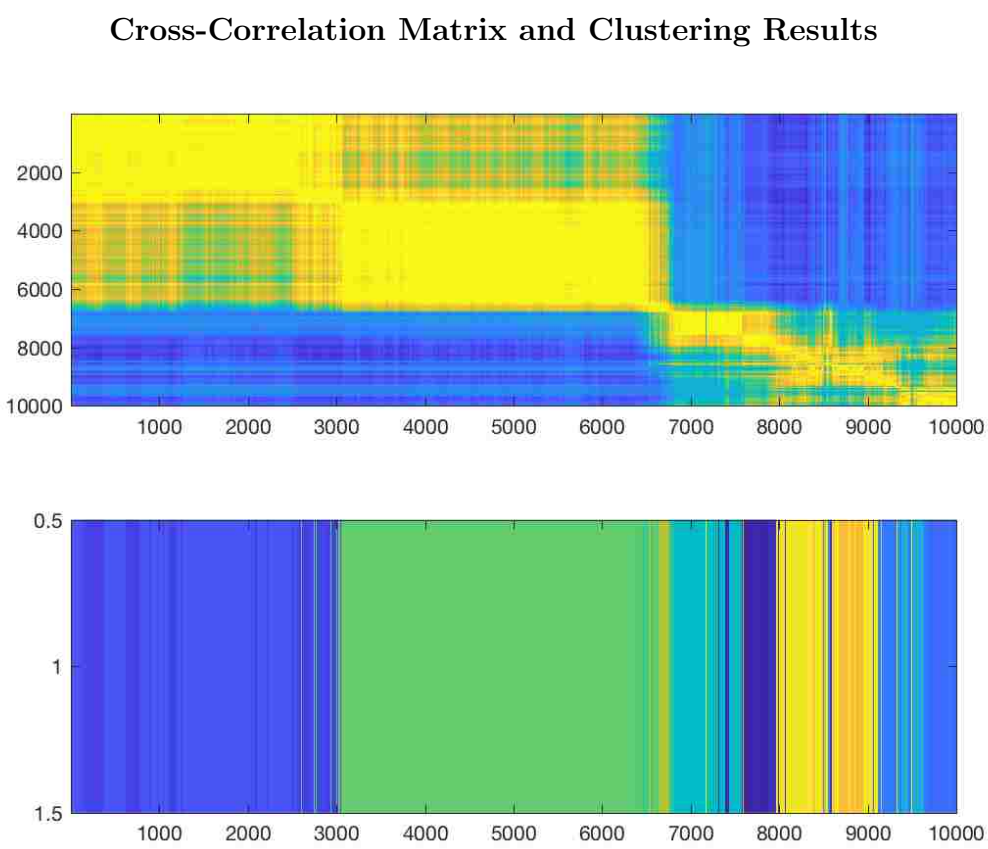
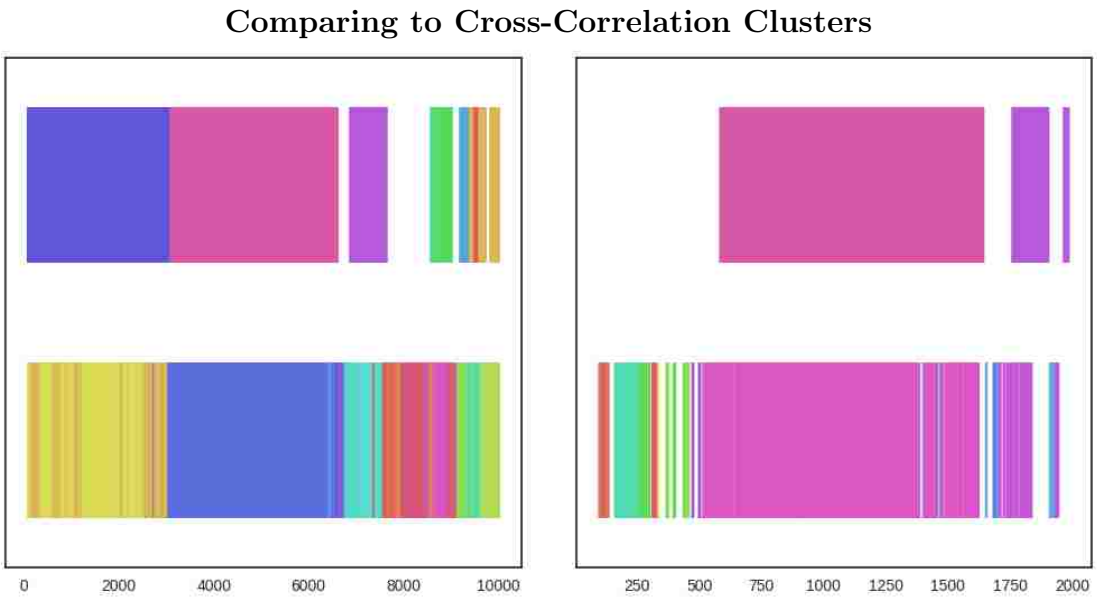


Figure 6.2: This is the same protein as seen in Fig. 6.1b. The top is the cross correlation matrix where each cell of the matrix represents how similar one frame of the simulation is to another; yellow being very close and blue being far apart. The bottom is the clustering label results over time.

6.4 Conclusion

In this chapter, we showed that it is possible to detect when a protein folding simulation is in a stable configuration. This method can be done in real time given that the root mean squared deviations time series can be produced in real time. We also compared our methodology to a cross-correlation plus clustering methodology which cannot be done in real time, and showed we can produce similar results as well as ignore labeling areas of transition. Future work includes using a different model for



(a) Protein 2 (b) Protein 4

Figure 6.3: Two examples of how the cross-correlation clusters compare to the our stability labels. The top rows are the stability labels and the bottom rows are the cross-correlation clusters over time. Blank area represent no label.

stability, possibly using a combination of an Autoregressive model and High Density Regions (similar to Chapter 5).

Chapter 7

Discussion and Conclusion

In this paper we discussed the how to interpret our distribution results using High Density Region analysis. In Chapter 4 we went through a seismic onset detection problem where the output was a likelihood distribution over possible onset times. We used HDRs to highlight the most likely time ranges for where an onset may occur. This is import for analysts so that they can quickly see which areas of the waveform are the most important to investigate in finer detail. In Chapter 5 we presented an arctic tipping point problem where we wanted to know when our time series started diverging from its current trajectory. This is important because a rapid change in the time series is an indicator that we are about to enter a tipping event. If we can detect it early enough there is a chance to stop the divergence. We used a forward projection analysis in order to see where the next point should be and then compared that distribution to the actual next point. Using an HDR analysis we could see which HDR percentile the next point was in and used that to determine how likely our time series was diverging into a tipping point. Lastly, in Chapter 6 we went through a protein folding simulation problem where we wanted to know when the protein was in a stable conformation. Finding these stable conformations is import to post-simulation analysis because it informs the analyst of how a specific

protein will react. Given a set of distributions of how likely a timestep is in a specific conformation we were able to use HDRs to interpret whether the protein was in a stable conformation. This was done by using HDRs to determine if the mass of the distribution was mostly above a specified threshold which is an indicator of stability. All three of the problems presented in this dissertation are considered change point detection problems. They all involved determining points in the time series where the pattern changed.

These techniques have been used in the statistics community for a number of years, but have been rarely utilized in this fashion by decision making analysts. We mostly see point estimates with error bars, which glosses over the fine grain details of the underlying uncertainty distribution; such as multiple modes or how tight the distribution is. HDRs can provide this fine grain insight with minimal effort. Being able to highlight the important features of the distribution can help improve decision making capabilities.

Future work for this research includes applying it to other pattern recognition and change point problems; such as trajectory analysis. We also want to explore how we can combine our uncertainties from multiple sensors/modalities in an effort to have an overall uncertainty for a given problem. Lastly we would like to improve our visualization techniques for uncertainties when we have higher dimensional data. Heat maps work well for 2-D and 3-D data sets, but tend to be overwhelming beyond that. If we can better visualize and highlight the important areas of our results and data, we can better utilize the tools from the statistics community in order to help analyst make more confident decisions.

References

[1] H. Akaike, *A new look at statistical model identification*, IEEE Transactions on Automatic Control **19** (1974), no. 6, 716–723.

[2] Richard B Alley, *Ice-core evidence of abrupt climate changes*, Proceedings of the National Academy of Sciences **97** (2000), no. 4, 1331–1334.

[3] Paul D Arendt, Daniel W Apley, Wei Chen, David Lamb, and David Gorsich, *Improving identifiability in model calibration using multiple responses*, Journal of Mechanical Design **134** (2012), no. 10, 100909.

[4] David Baker, *Metastable states and folding free energy barriers*, Nature Structural and Molecular Biology **5** (1998).

[5] Andrea L Bertozzi, Xiyang Luo, Andrew M Stuart, and Konstantinos C Zygalakis, *Uncertainty quantification in graph-based classification of high dimensional data*, arXiv preprint arXiv:1703.08816 (2017).

[6] C. Best and H. C. Hege, *Visualizing and identifying conformational ensembles in molecular dynamics trajectories*, Computing in Science Engineering **4** (2002), no. 5.

[7] Gerard C Bond, William Showers, Mary Elliot, Michael Evans, Rusty Lotti, Irka Hajdas, Georges Bonani, and Sigfus Johnson, *The north atlantic's 1-2 kyr climate rhythm: Relation to heinrich events, dansgaard/oeschger cycles and the*

little ice age, Mechanisms of global climate change at millennial time scales (1999), 35–58.

[8] Phillip Brohan, John J Kennedy, Ian Harris, Simon FB Tett, and Phil D Jones, *Uncertainty estimates in regional and global observed temperature changes: A new data set from 1850*, Journal of Geophysical Research: Atmospheres **111** (2006), no. D12.

[9] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton, *A survey of monte carlo tree search methods*, IEEE Transactions on Computational Intelligence and AI in games **4** (2012), no. 1, 1–43.

[10] Stephen J Burns, Dominik Fleitmann, Albert Matter, Jan Kramers, and Abdulkarim A Al-Subbary, *Indian ocean climate and an absolute chronology over dansgaard/oeschger events 9 to 13*, Science **301** (2003), no. 5638, 1365–1367.

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar, *Anomaly detection: A survey*, ACM computing surveys (CSUR) **41** (2009), no. 3, 15.

[12] David J. Stracuzzi Charlie Vollmer, Matt Peterson and Maximillian G. Chen, *Using data-driven uncertainty quantification to support decision making*, Statistical Data Science Conference (2017).

[13] Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck, *Monte-carlo tree search: A new framework for game ai.*, AIIDE, 2008.

[14] Sanjay Chawla and Pei Sun, *Slom: a new measure for local spatial outliers*, Knowledge and Information Systems **9** (2006), no. 4, 412–429.

[15] Michael R Chernick, Wenceslao González-Manteiga, Rosa M Crujeiras, and

Erniel B Barrios, *Bootstrap methods*, International Encyclopedia of Statistical Science, Springer, 2011, pp. 169–174.

[16] Torben R Christensen, Torbjörn Johansson, H Jonas Åkerman, Mihail Mastepanov, Nils Malmer, Thomas Friborg, Patrick Crill, and Bo H Svensson, *Thawing sub-arctic permafrost: Effects on vegetation and methane emissions*, Geophysical research letters **31** (2004), no. 4.

[17] Vasilis Dakos, Stephen R Carpenter, Egbert H van Nes, and Marten Scheffer, *Resilience indicators: prospects and limitations for early warnings of regime shifts*, Philosophical Transactions of the Royal Society B: Biological Sciences **370** (2015), no. 1659, 20130263.

[18] Maximillian G. Chen David J. Stracuzzi, Michael C. Darling and Matthew G. Peterson, *Data-driven uncertainty quantification for multi-sensor analytics*, SPIE (2018).

[19] Richard Davy and Igor Esau, *Global climate models bias in surface temperature trends and variability*, Environmental Research Letters **9** (2014), no. 11, 114024.

[20] Bradley Efron et al., *Second thoughts on the bootstrap*, Statistical Science **18** (2003), no. 2, 135–140.

[21] Bradley Efron and Robert J Tibshirani, *An introduction to the bootstrap*, CRC press, 1994.

[22] Dominique Genty, D Blamart, R Ouahdi, M Gilmour, A Baker, J Jouzel, and Sandra Van-Exter, *Precise dating of dansgaard–oeschger climate oscillations in western europe from stalagmite data*, Nature **421** (2003), no. 6925, 833.

[23] PM Grootes, Minze Stuiver, JWC White, S Johnsen, and J Jouzel, *Comparison of oxygen isotope records from the gisp2 and grip greenland ice cores*, Nature **366** (1993), no. 6455, 552.

[24] Robin Hill, Brendan Healy, Lois Holloway, Zdenka Kuncic, David Thwaites, and Clive Baldock, *Advances in kilovoltage x-ray beam dosimetry*, Physics in medicine and biology **59** (2014), no. 6, R183.

[25] Rob J Hyndman, *Computing and graphing highest density regions*, The American Statistician **50** (1996), no. 2, 120–126.

[26] Marco Iglesias and Andrew M Stuart, *Inverse problems and uncertainty quantification*, SIAM News, volume July/August (2014).

[27] Marco A Iglesias, Kui Lin, and Andrew M Stuart, *Well-posed bayesian geometric inverse problems arising in subsurface flow*, inverse problems **30** (2014), no. 11, 114001.

[28] J. T. Mathis J. Richter-Menge, J. E. Overland and Eds. E. Osborne, *2017: Arctic report card 2017*, http://www.arctic.noaa.gov/Report-Card.

[29] Xun Jia, Peter Ziegenhein, and Steve B Jiang, *Gpu-based high-performance computing for radiation therapy*, Physics in medicine and biology **59** (2014), no. 4, R151.

[30] T. Johnston, B. Zhang, A. Liwo, S. Crivelli, and M. Taufer, *In situ data analytics and indexing of protein trajectories*, J Comput Chem. **38** (2017), no. 16.

[31] Malvin H Kalos and Paula A Whitlock, *Monte carlo methods*, John Wiley & Sons, 2008.

[32] O. Kamigaichi, *A fully automated method for determining the arrival times of seismic waves and its application to an on-line processing system*, Proceedings 34th GSE session, GSE/RF/62 (Geneva, Italy), G.S.E., 1992.

[33] Rebecca Killick, IA Eckley, Philip Jonathan, et al., *A wavelet-based approach for detecting changes in second order structure within nonstationary time series*, Electronic Journal of Statistics **7** (2013), 1167–1183.

[34] Yufeng Kou, Chang-Tien Lu, and Dechang Chen, *Spatial weighted outlier detection*, Proceedings of the 2006 SIAM international conference on data mining, SIAM, 2006, pp. 614–618.

[35] O. P. Le Maître and O. M. Knio, *Introduction: Uncertainty quantification and propagation*, pp. 1–13, Springer Netherlands, Dordrecht, 2010.

[36] Sang Hoon Lee and Wei Chen, *A comparative study of uncertainty propagation methods for black-box-type problems*, Structural and Multidisciplinary Optimization **37** (2009), no. 3, 239–253.

[37] Stephen MS Lee and G Alastair Young, *Parametric bootstrapping with nuisance parameters*, Statistics & probability letters **71** (2005), no. 2, 143–153.

[38] Yanxin Liu, Peter Freddolino, Martin Gruebele, and Klaus Schulten, *Md simulation of protein folding*, http://www.ks.uiuc.edu/Research/folding/.

[39] HT MacGillivray, RJ Dodd, BV McNally, and HG Corwin, *Orientations of galaxies in the local supercluster*, Monthly Notices of the Royal Astronomical Society **198** (1982), no. 2, 605–615.

[40] Raman Maini and Himanshu Aggarwal, *Study and comparison of various image edge detection techniques*, International journal of image processing (IJIP) **3** (2009), no. 1, 1–11.

[41] JA Maslanik, C Fowler, J Stroeve, S Drobot, J Zwally, D Yi, and W Emery, *A younger, thinner arctic ice cover: Increased potential for rapid, extensive sea-ice loss*, Geophysical Research Letters **34** (2007), no. 24.

[42] Justin Matejka and George Fitzmaurice, *Same stats, different graphs: Generating datasets with varied appearance and identical statistics through simulated annealing*, Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM, 2017, pp. 1290–1294.

[43] Nicholas Metropolis and Stanislaw Ulam, *The monte carlo method*, Journal of the American statistical association **44** (1949), no. 247, 335–341.

[44] Mariusz Milik and Jeffrey Skolnick, *Insertion of peptide chains into lipid membranes: an off-lattice monte carlo dynamics model*, Proteins: Structure, Function, and Bioinformatics **15** (1993), no. 1, 10–25.

[45] Terence C Mills, *Time series techniques for economists*, Cambridge University Press, 1991.

[46] INB. Molecular modeling and Bioinformatics Group, *Molecular dynamics extended library*, http://mmb.pcb.ub.es/MoDEL/index.jsf.

[47] Christopher FH Nam, John AD Aston, Idris A Eckley, and Rebecca Killick, *The uncertainty of storm season changes: Quantifying the uncertainty of autocovariance changepoints*, Technometrics **57** (2015), no. 2, 194–206.

[48] Caleb C Noble and Diane J Cook, *Graph-based anomaly detection*, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 631–636.

[49] Manuel Ojeda and Enrique Iglesia, *Formic acid dehydrogenation on au-based catalysts at near-ambient temperatures*, Angewandte Chemie **121** (2009), no. 26, 4894–4897.

[50] J. Phillips, *Validating clustering of molecular dynamics simulations using polymer models*, BMC Bioinformatics **12** (2011), no. 1.

[51] Stefan Rahmstorf, *Bifurcations of the atlantic thermohaline circulation in response to changes in the hydrological cycle*, Nature **378** (1995), no. 6553, 145.

[52] G.N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan, *Multipolar representation of protein structure*, Journal of Molecular Biology **7** (1963), no. 95.

[53] F Ringdal and ES Husebye, *Application of arrays in the detection, location, and identification of seismic events*, Bulletin of the Seismological Society of America **72** (1982), no. 6B, S201–S224.

[54] Ismael Vera Rodriguez, *Automatic time-picking of microseismic data combining sta/lta and the stationary discrete wavelet transform*, CSPG CSEG CWLS Convention, convention abstracts, 2011.

[55] Donald B Rubin et al., *The bayesian bootstrap*, The annals of statistics **9** (1981), no. 1, 130–134.

[56] Stan Salvador, Philip Chan, and John Brodie, *Learning states and rules for time series anomaly detection.*, FLAIRS Conference, 2004, pp. 306–311.

[57] Fabrizio Sebastiani, *Machine learning in automated text categorization*, ACM computing surveys (CSUR) **34** (2002), no. 1, 1–47.

[58] Shashi Shekhar, Chang-Tien Lu, and Pusheng Zhang, *Detecting graph-based spatial outliers: algorithms and applications (a summary of results)*, Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2001, pp. 371–376.

[59] Léopold Simar and Paul W Wilson, *A general methodology for bootstrapping in non-parametric frontier models*, Journal of applied statistics **27** (2000), no. 6, 779–802.

[60] James H Steiger, *Tests for comparing elements of a correlation matrix.*, Psychological bulletin **87** (1980), no. 2, 245.

[61] Thomas Stocker, *Climate change 2013: the physical science basis: Working group i contribution to the fifth assessment report of the intergovernmental panel on climate change*, Cambridge University Press, 2014.

[62] T. Tu, *A scalable parallel framework for analyzing terascale molecular dynamics simulation trajectories*, CM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC08), 2008.

[63] A Velasco, C Young, and D Anderson, *Uncertainty in phase arrival time picks for regional seismic events: an experimental design*, Tech. report, Tech. rep., US Department of Energy, 2001.

[64] Christina Warrender, Stephanie Forrest, and Barak Pearlmutter, *Detecting intrusions using system calls: Alternative data models*, Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on, IEEE, 1999, pp. 133–145.

[65] Andreas S Weigend, Morgan Mangeas, and Ashok N Srivastava, *Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting*, International Journal of Neural Systems **6** (1995), no. 04, 373–399.

[66] Cleat Zeiler and Aaron A. Velasco, *Seismogram picking error from analyst review (SPEAR): Single-analyst and institution analysis*, Bulletin of the Seismological Society of America **99** (2009), no. 5, 2759–2770.

[67] B. Zhang, T. Estrada, P. Cicotti, and M. Taufer, *Enabling in-situ data analysis for large protein folding trajectory datasets*, IEEE International Parallel and Distributed Processing Symposium, 2014.

[68] H Jay Zwally, Waleed Abdalati, Tom Herring, Kristine Larson, Jack Saba, and Konrad Steffen, *Surface melt-induced acceleration of greenland ice-sheet flow*, Science **297** (2002), no. 5579, 218–222.