Signal-based Bayesian Seismic Monitoring

David Moore

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2016-192
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-192.html

December 1, 2016

Signal-based Bayesian Seismic Monitoring

by

David Andrew Moore

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Stuart Russell, Chair
Doug Dreger
Jitendra Malik
Martin Wainwright

Fall 2016

Signal-based Bayesian Seismic Monitoring

Abstract

Signal-based Bayesian Seismic Monitoring

by

David Andrew Moore

Doctor of Philosophy in Computer Science

University of California, Berkeley

Stuart Russell, Chair

This thesis presents a new approach to seismic monitoring, the task of detecting seismic events from potentially noisy and cluttered signals recorded across multiple stations. Unlike previous work, which represents seismic signals by a lossy set of discrete detections, we specify a generative probability model of raw seismic waveforms, incorporating a rich representation of the physics underlying the signal generation process, including source mechanisms, wave propagation, and station response. Inference in this model recovers the qualitative behavior of geophysical methods including waveform matching and double-differencing, all as part of a unified Bayesian monitoring system that simultaneously detects and locates events from a network of stations.

Our model of seismic signals combines physically meaningful latent variables such as phase travel times, amplitudes, and signal decay rates, with data-driven models based on historical signals. Detailed waveform structure is represented using Gaussian process models of wavelet coefficients, encoding a general assumption that seismic signals are spatially correlated, and allowing us to detect and locate events even from weak signals at a single station. We show that the wavelet coefficients can be marginalized out using message passing applied to a state-space representation of the signal model, allowing for practical inference using a reversible jump Metropolis-Hastings algorithm.

We evaluate our system, SIGVISA (Signal-based Vertically Integrated Seismic Analysis), on a task of monitoring the western United States for a two-week period following the magnitude 6.0 event in Wells, NV in February 2008. During this period, SIGVISA detects between two to three times as many events as detection-based systems, while reducing mean location errors by a factor of four. We provide evidence that SIGVISA detects some events that are missed even by the regional monitoring networks that we use as a ground-truth comparison. A primary driver of monitoring research is the verification of nuclear test ban treaties, which are particularly concerned with detecting events in regions with no nearby historical seismicity. In our experiments, SIGVISA matches or exceeds the detection rates of existing systems for such events, and even detects a number of such events missed by human analysts.

in memory of Maggie

Contents

List of Figures

List of Tables

Acknowledgments

Thanks to everyone who contributed to the work in this thesis. Steve Myers and Kevin Mayeda both provided invaluable advice on issues from modeling to experimental setup, suggested relevant reading material, and have been helpful and patient teachers of remedial seismology for computer scientists. Steve also saved us political headaches by helping access American IMS waveform data through LLNL. It's been my privilege to work with great undergrad and master's researchers — Min Joon Seo, Alex Ding, Sharad Vikram, Xiaofei Zhou, Zhiyuan Lin, and Jun Song — all of whom have contributed directly or indirectly to this work. I'm also thankful to my committee for their helpful feedback and corrections. Any remaining errors are mine alone.

Of course, none of this would have been possible without the vision and encouragement of my advisor, Stuart Russell, who has helped me see past the technical weeds to stay motivated by the big questions. I admire his fearlessness in pursuing ambitious, difficult, and important problems; I hope some of that has rubbed off. I'm also grateful to the other members of RUGS, our unusual group of students, for helpful discussions, advice, and many delicious Gregoire lunches.

I wouldn't have made it to Berkeley without the love and support of my parents, who have always encouraged me to pursue my interests and mostly didn't complain when this involved staying up all night doing strange things with computers. My undergrad advisor, Andrea Danyluk, got me started with research as a freshman and went out of her way to supervise my thesis even while serving as a full-time dean. More broadly, I'm grateful to the Williams CS department for creating such a vibrant and welcoming environment; I can't imagine a better place to be an undergrad studying computer science.

Finally, thanks to all the friends who have made these years in Berkeley such a special time. Shaddi, for being a great roommate and helping California feel a bit more like home. The 1044 extended family (you know who you are) for all the parties, Shabbat dinners, Tahoe trips, Game of Thrones viewings, and countless other warm gatherings past and yet to come. Jake, for being with me through good times and bad: you are still missed. And Arjun, who has contributed to this work mostly by repeatedly interrupting it; someday your investment will pay off.

This work was primarily supported by the Defense Threat Research Agency (DTRA), grant HDTRA-1111-0026. Preliminary stages were supported by the Comprehensive Test Ban Treaty Organization (CTBTO), award 2010-1225, which also provided access to data via the virtual Data Exploitation Centre (vDEC). Computational resources were provided by an Azure for Research grant from Microsoft.

Chapter 1

Introduction: the monitoring problem

This thesis is concerned with *seismic monitoring*: the task of detecting and locating seismic events given waveforms recorded by a network of seismic stations.

This is of obvious relevance to seismologists, who have a scientific interest in cataloguing natural earthquake activity. The advent of nuclear weapons and underground nuclear testing has created an additional political motivation: the detection of nuclear explosions. An underground nuclear explosion releases energy comparable to a moderate-sized earthquake and is recorded similarly by seismometers. A sufficiently sensitive monitoring system can detect nuclear tests and infer the test site location along with the yield of the explosion.

Verification of the Comprehensive Test Ban Treaty (CTBT) is a major focus of monitoring research, although individual states also operate their own monitoring networks for both geopolitical and scientific purposes. In this chapter, we situate the monitoring problem in the context of the CTBT, introduce and motivate the formulation of the problem as Bayesian inference, and detail the contributions of the *signal-based* approach to monitoring developed in this thesis.

1.1 The Comprehensive Test Ban Treaty

The text of the CTBT obliges each state signatory "not to carry out any nuclear weapon test explosion or any other nuclear explosion, and to prohibit and prevent any such nuclear explosion at any place under its jurisdiction or control." The treaty additionally establishes an international organization headquartered in Vienna, the Comprehensive Test Ban Treaty Organization (CTBTO), having among its responsibilities the verification of this ban (CTBTO, 2015).

The treaty specifies a verification regime centered around a worldwide network of seismic, hydroacoustic, infrasound, and radionuclide detectors, known as the International Monitoring System (IMS). In this thesis we focus on seismic monitoring, with other sensor types left for future work. The IMS seismic network (Figure 1.1) consists of both three-component and array seismometers (Section 2.4), which are certified to meet certain technical standards and

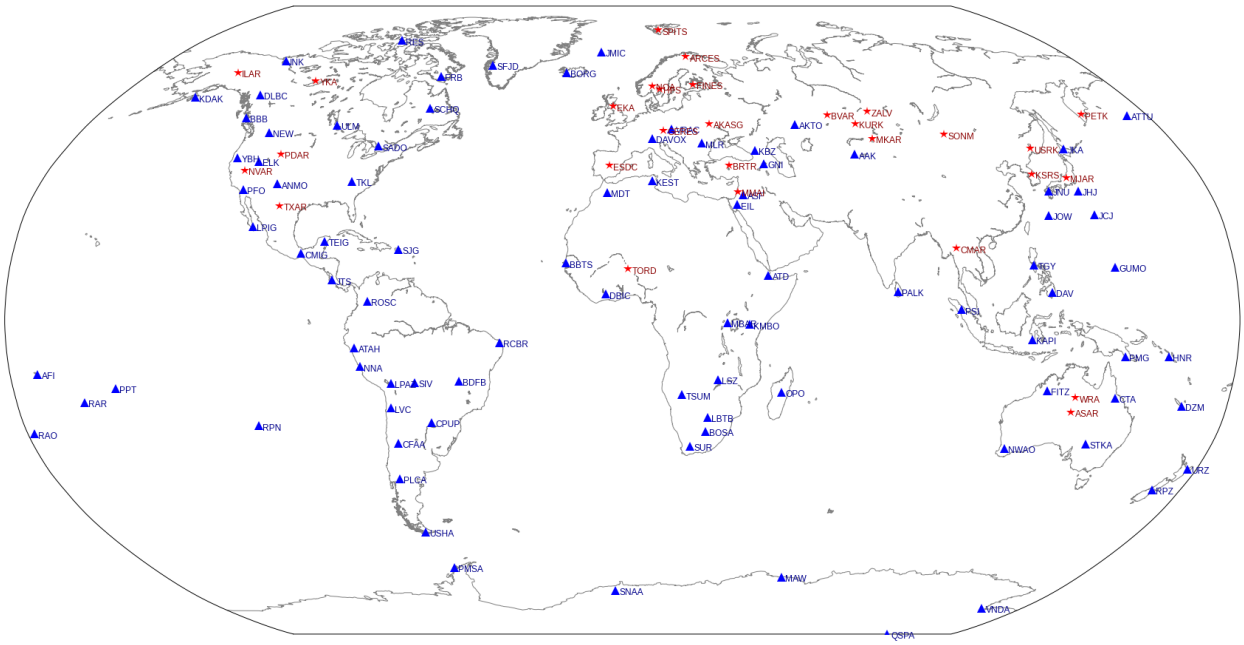


Figure 1.1: The IMS global seismic monitoring network. Red stars indicate array stations.

are installed with tamper-detection devices. Data from the monitoring stations are cryptographically signed and transmitted to the International Data Center (IDC) in Vienna for processing (CTBTO, 2008).

At the IDC, an initial bulletin of seismic events is produced by an automated monitoring system (Section 2.5.3), then reviewed by human analysts (Section 2.5.4) to generate a worldwide catalog of reviewed events that is made available to the state parties. The seismic bulletin does not itself attempt to distinguish natural seismicity from potential nuclear tests; state parties may view certain events as suspicious and choose to conduct further investigations, involving analysis by expert seismologists or other sources such as radionuclide evidence.

If any state party suspects a treaty violation, they may request an on-site inspection by a team of scientific experts, subject to approval by an Executive Council representing the collective signatories. The inspection area is limited to 1000 square kilometers, extending for a maximum linear distance of 50km in any direction (CTBTO, 2015). A successful inspection therefore requires a precise estimate of the suspected event's location.

Effective verification is crucial to the operation of the CTBT: states will not compromise their own nuclear capabilities unless they have confidence that all other parties are adhering to the same restrictions. Improvements in monitoring technology are therefore directly relevant to the treaty's political viability.

The treaty enters into force upon ratification by a listed group of 44 states, all of whom have possessed nuclear technology in the form of nuclear power or research reactors. Many states have already ratified the treaty, but holdouts include the United States, China, Egypt,

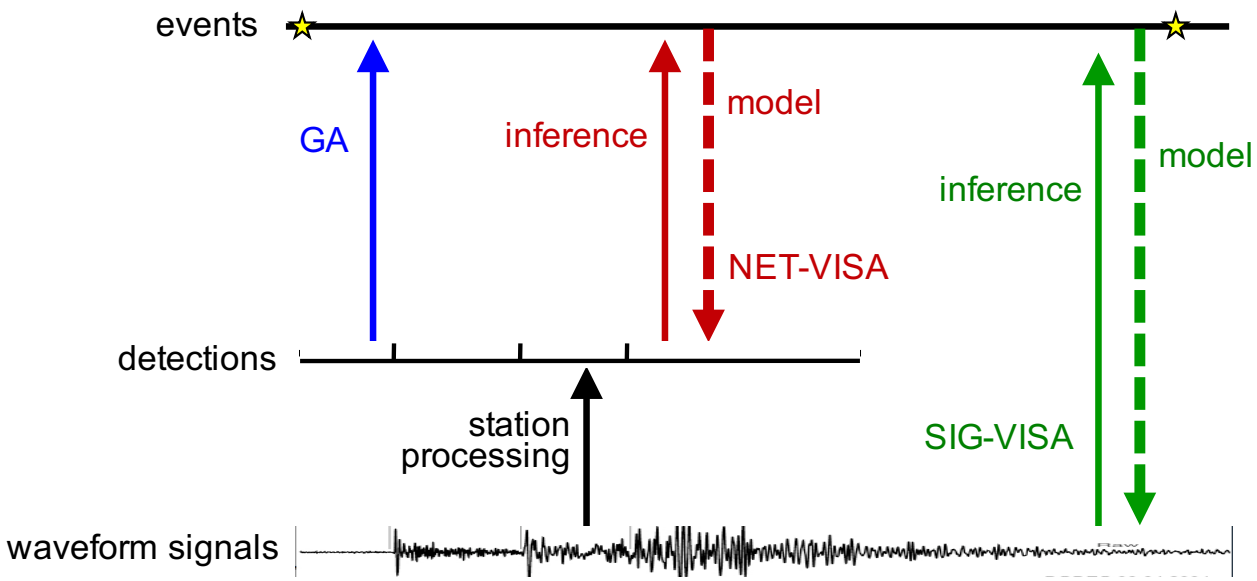


Figure 1.2: High level structure of traditional monitoring (GA), detection-based Bayesian monitoring (NETVISA), and signal-based Bayesian monitoring (SIGVISA, this thesis). Where GA and NETVISA are limited to detections produced by bottom-up station processing, inference in a signal-based model incorporates much richer information from the signals themselves.

Indonesia, Iran, Israel, India, North Korea and Pakistan (Gorbachev, 2010). In 1999, the US Senate voted 51-48 not to ratify the treaty. One of the primary objections raised was the difficulty of effective verification, and in particular the inadequacy of currently available monitoring technology.

1.2 Monitoring as Bayesian inference

This thesis approaches the monitoring problem via the framework of Bayesian inference. That is, we define a *prior distribution* $p(\text{events})$ on seismic events and a *forward model* or *likelihood* $p(\text{signals}|\text{events})$ describing how those events generate the signals we observe. We then apply Bayes' rule to yield a *posterior* distribution

$$p(\text{events}|\text{signals}) \propto p(\text{signals}|\text{events})p(\text{events}),$$

which concentrates probability on event histories (bulletins) that plausibly explain the observed signals. The posterior represents the unique and mathematically correct distribution over events, as defined by the laws of probability theory given our modeling assumptions and observed data.

The Bayesian approach is in contrast to traditional systems, such as the Global Association (GA) system in use at the IDC (Section 2.5.3), that map directly from observations to

an event bulletin without committing to an explicit model or representing the uncertainty in their inferences (Figure 1.2).[1] Compared to such systems, an explicit Bayesian formulation has several advantages:

- **Explicit assumptions.** The prior and forward model collectively encode and make explicit our assumptions about the data-generating process. Where the algorithms in a traditional system encode implicit assumptions that may be unclear or even inconsistent, a Bayesian system decouples the domain model from the algorithms used to perform inference. The model is intelligible to and interpretable by seismic experts, and improvements to the model lead directly to improved system performance.

- **Unifying top-down and bottom-up processing.** The processing in a traditional system is purely bottom-up, moving from observed signals to a set of discrete *detections* at each station, which are then associated into events (Figure 1.2). Each step of this process discards potentially valuable information from the observed signal: setting the detection threshold too high leads to missed events, while setting it too low creates many false detections. Inference in a Bayesian model effectively incorporates top-down as well as bottom-up processing, interpreting each signal in the context of evidence from other observations, so that potentially promising events are not discarded due to missed detections.

- **Negative evidence.** By relying on a principled mathematical problem formulation, a Bayesian system incorporates all available evidence, including *negative* evidence. If an event in a particular location should be observed by a particular station, and that station does not in fact observe any events, this ought to weigh heavily against that event hypothesis. Traditional systems typically do not enforce this form of consistency.

- **Multiple sensors.** Bayesian modeling provides a natural path to incorporation of multiple sensor modalities (infrasound, hydroacoustic, etc.) by simply including them as additional components in the forward model. Even within purely seismic monitoring, we use this same approach to integrate information from multiple stations, so that the overall network incorporates evidence weighted according to the individual sensitivity and noise level of each station.

- **Quantified uncertainty**. Where traditional systems output a single estimated bulletin, a Bayesian posterior represents the uncertainty inherent in the inference process. This includes error ellipses for the locations of individual events, but additionally represents more complex ambiguities that may exist regarding the events themselves. If a set of signals could be explained equally well by one event in location A, or two events in locations B and C, the Bayesian posterior will contain both hypotheses, while a traditional system must select one arbitrarily.

[1] Traditional systems may include models of specific phenomena such as wave velocities, as components, but the phase picks, associations, and bulletins ultimately produced are not derived by inverting any coherent overall forward model.

Previous work (Arora et al., 2013) has developed a Bayesian monitoring system, Network Processing Vertically Integrated Seismic Analysis (NETVISA), based on a generative model of the discrete detections produced by traditional station processing (Section 2.5.1). Although this has been quite successful, providing a strong proof of concept for the Bayesian approach, the performance of detection-based systems is inherently limited by the noisy, myopic, and lossy nature of the bottom-up detection pipeline. Our current work extends Bayesian monitoring to remove the dependence on station processing, performing inference directly on continuous waveform observations (Figure 1.2). As we show in Chapter 7, this allows our system to detect significantly more events than existing automated systems, including many events missed by human analysts.

1.3 Contributions

This thesis presents a new approach to seismic monitoring, using Bayesian inference in a generative model of continuous seismic signals, and argues for the advantages of this approach. In particular, we describe and evaluate a system that we call Signal-based Vertically Integrated Seismic Analysis (SIGVISA), which consists of a joint probability model of seismic events and signals as well as a set of algorithms for training and performing Bayesian inference in this model.

We begin by surveying related work and background in seismic monitoring (Chapter 2), as well as relevant technical background in Bayesian modeling, machine learning, and signal processing (Chapter 3). We then motivate and detail the SIGVISA generative model, consisting of a prior distribution on events and a forward model of seismic signals (Chapter 4). We argue that this model incorporates into a single framework the phenomena underlying existing detection and location techniques such as multilateration (Section 2.5.2), waveform correlation matching (Section 2.6), and double-differencing (Section 2.7).

In Chapter 5 we present an algorithm for inference in the SIGVISA model, involving MCMC applied to a collapsed (Rao-Blackwellized) model structure. To accelerate the inference process, we provide custom proposals for birthing seismic events, given by the posteriors of simpler surrogate models that capture different aspects of the full model. We introduce a procedure for processing large datasets via parallel inference, and for improving inference quality by merging results from an ensemble of MCMC chains. In Chapter 6 we describe a training procedure that fits the model parameters to historical data, and can be parallelized to efficiently handle large training sets.

Chapter 7 evaluates our system on an application to monitoring seismic events in the western United States. We demonstrate that SIGVISA detects many more events than detection-based systems while operating at the same precision, including events in lower magnitude ranges, and that it is able to locate these events with significantly greater accuracy. Using only IMS network data, SIGVISA is able to detect some events that are missed even by more sensitive regional networks. We additionally show that its performance on *de novo* events, of particular relevance to monitoring applications, equals or exceeds that

of detection-based systems. Finally, in Chapter 8 we conclude and discuss directions for potential future work involving extensions and improvements to the SIGVISA model.

Chapter 2

Seismic background and related work

This chapter surveys the existing landscape of seismic monitoring. We begin with a brief review of seismic sources and wave propagation, including phase types, travel-time models, and waveforms recorded at seismic stations. We then describe the detection-based monitoring pipeline used by the CTBTO, including the use of "picking" to convert continuous waveforms into discrete detections, network-level processing to form and locate events, and review by human analysts. We also review alternative approaches to detection-based network processing, including the Bayesian approach implemented in NETVISA.

We then consider processing techniques that incorporate signal information directly, including waveform correlation matching for event detection and location, and batch event relocation via double-differencing. Finally we discuss previous attempts to model directly the shape of seismic signal envelopes, which inspire the forms used in our current work.

2.1 Seismic sources and waves

A seismic source is any event that releases energy in the form of seismic waves; this could be a fault rupture, rockslide, volcanic eruption, bolide impact, nuclear explosion, series of mining shots, or any other disruption. Sources may have both spatial and temporal extent (e.g., slippage lasting tens of seconds along a fault line extending tens of kilometers), though typical models, including those in this thesis, assume a point source that can be localized in space and time.

The waves produced are of two major types: compression waves, which displace the earth in the direction of travel, like sound waves in air, and shear waves, whose displacement is perpendicular to the direction of travel, like ripples in a pond. Because compression waves travel more quickly and are typically the first to be detected, they are called *primary* or P-waves, while shear waves are called *secondary* or S-waves. Both P and S waves are considered *body waves* because they travel through the solid earth.

Interactions of P and S waves with geological discontinuities can produce a multitude of additional reflected, refracted, and diffracted body wave arrivals. These can significantly

complicate observed waveforms. Furthermore, P and Sv (vertically polarized S) waves may convert to the other type, so that a S wave arriving at a station may have traveled most of the way as a P wave before converting at a nearby discontinuity.

Body waves reaching the surface may also generate new *surface waves* propagating horizontally. Interference among S waves reaching the surface produces *Love* waves, which are characterized by particle motion in the horizontal plane perpendicular to the direction of travel, while interactions between P and S waves generate *Rayleigh* waves (also known as "ground roll"), characterized by elliptical partical motion normal to the surface and parallel to the direction of travel. Because surface waves travel in a two-dimensional plane, their amplitudes decay only with the square root of distance (as opposed to linearly for unguided body waves), so they may retain their strength even at long distances; surface waves are responsible for most of the damage to surface structures from large earthquakes.

Seismologists have developed a number of ways to quantify the magnitude of a seismic source. Historically the *body-wave magnitude*, m_b, and *surface-wave magnitude*, M_s, have been defined respectively in terms of the maximum (log) amplitude of observed body and surface waves, accounting for event–station distance and origin depth (Aki and Richards, 1980). These quantities are straightforward to compute and are widely used in monitoring systems, but have no intrinsic physical meaning; they are really properties of the observer rather than the event itself. The same event may have different body-wave and surface-wave magnitudes, which may also differ from station to station and even be defined differently by different monitoring networks.

A more principled approach is to consider the *seismic moment*

$$M_0 = \mu A D,$$

which measures the total energy released and is determined by the area A of fault rupture in square meters, the average displacement D in meters, and the shear modulus μ of the ruptured rock, in pascals (N/m^2), so that M_0 has units of energy (joules). The *moment magnitude* is then defined as a logarithmic function of the moment,

$$M_w = (2/3) \log_{10} M_0 - 10.7,$$

where the constants are chosen to align roughly with the traditional M_S and m_b scales (Kanamori and Hanks, 1979). As a direct measure of energy release, the moment magnitude is more physically meaningful than body- or surface-wave magnitudes, though it must be estimated rather than calculated directly from observed signals. For nuclear explosions, the energy release is commonly reported in terms of the *yield* in kilotons of TNT equivalent. Of the energy released by an earthquake or explosion, only a small fraction, known as the *seismic efficiency*, is actually radiated as seismic waves (Kanamori, 2001); the remainder is released as heat or retained as potential energy in deformed rocks, so that well-calibrated yield estimates must account for the efficiency of the source mechanism as well as the amplitudes of observed signals.

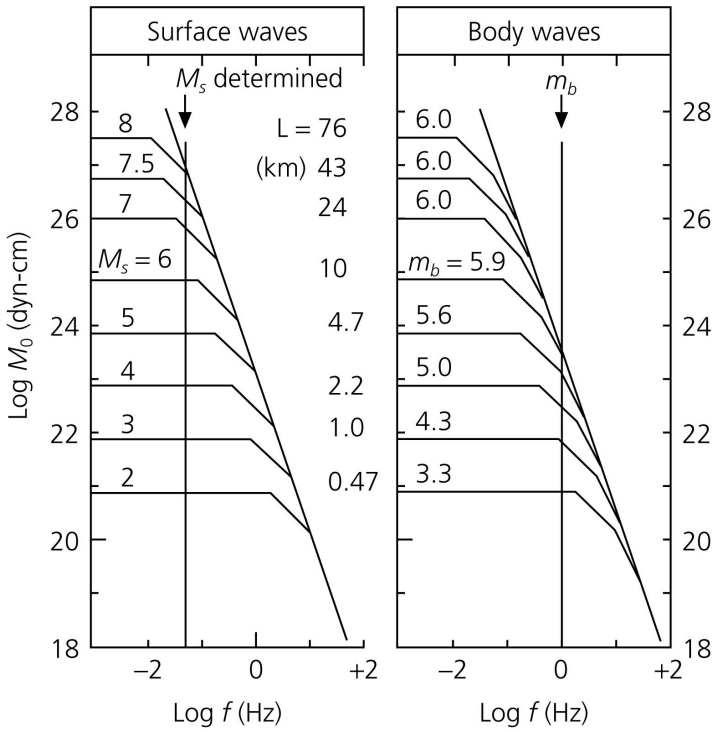


Figure 2.1: Energy (M_0) radiated at different frequencies by earthquakes of varying surface- and body-wave magnitude. High-magnitude events release more energy but their frequency content drops off at lower frequencies. (Source: Stein and Wysession, 2009)

Both the directionality of wave propagation and the portion of energy radiated as P versus S waves may be affected by the source mechanism. For example, nuclear explosions generate primarily P waves, radiated isotropically (equally in each direction), while a fault rupture will generally produce both P and S waves which may be focused according to the orientation of the fault. The relative lack of S waves from exploisions leads to weaker surface waves, so the difference $m_b - M_s$ between body- and surface-wave "magnitudes" has historically been an effective discriminant between earthquake and explosion sources (Marshall and Basham, 1972).

Source characteristics may also affect the frequency content of seismic waves. Typical events release roughly constant energy in all frequencies up to some *corner frequency* where a drop-off begins (Figure 2.1); this corner frequency is lower for higher-magnitude events. The classical Brune source model (Brune, 1970) attempts to model the frequency spectra of natural earthquakes, as a function of event magnitude, while the Mueller–Murphy model (Mueller and Murphy, 1971) provides an analogous model for nuclear explosions. Richer models of seismic source physics are an active and ongoing area of research.

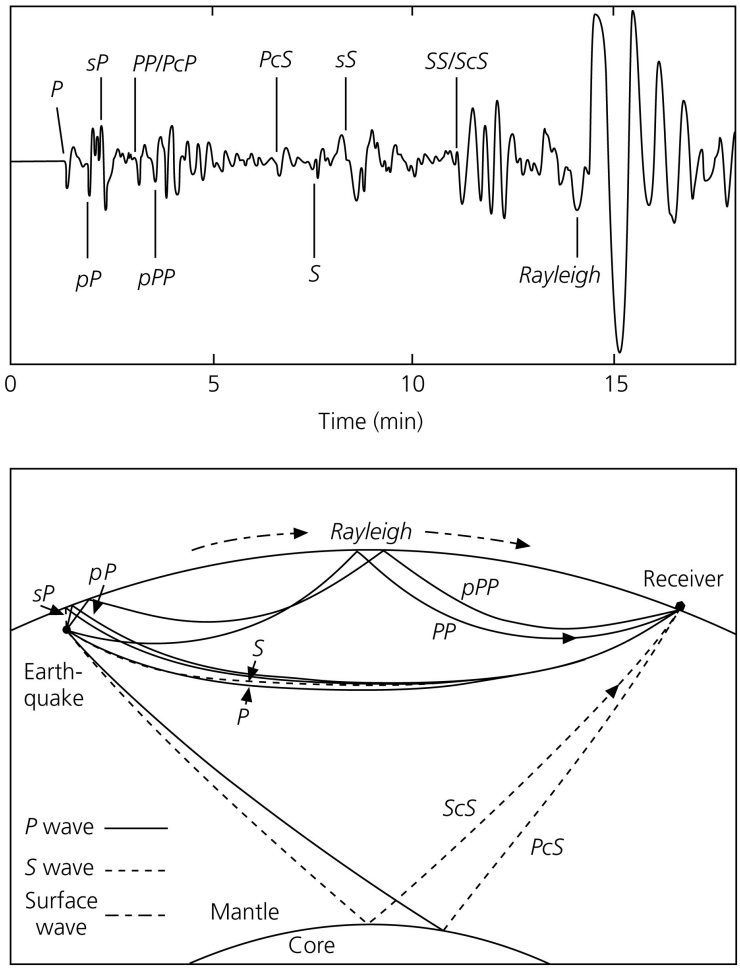


Figure 2.2: Stylized seismic waveform (top) illustrating arrivals at different times from multiple event phases (bottom), including direct waves as well as waves reflected from the surface and from the core–mantle boundary. (Source: Stein and Wysession, 2009)

2.2 Phases and travel times

Seismic waves follow a variety of paths from their source to a given detecting station; these paths are taxonomized as seismic *phases*. Phases are categorized by their wave type — P, S, Love, Rayleigh, etc. — as well as additional indicators for the specific path followed by each wave. For example, P waves are subdivided into *Pg* waves that travel directly through the crust at short distances, "plain" P waves that travel long distances through the mantle, *Pn* waves guided by refraction along the crust–mantle boundary, and *pP* waves reflected from the surface, among many other possibilities (Figures 2.2 and 2.3). The particular phases observed from a given event depend in general on its depth and distance from the receiving station.

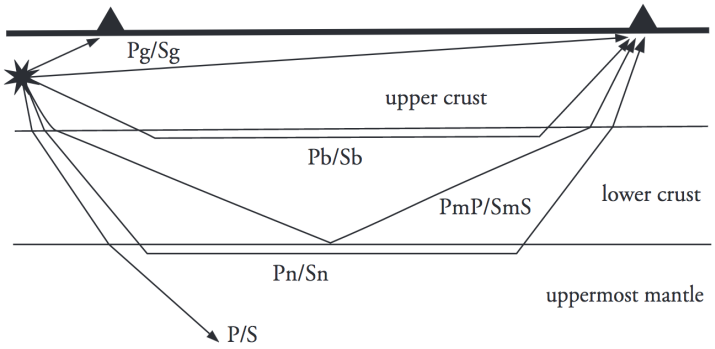


Figure 2.3: Examples of phase paths observed at local and regional distances ($< 20°$) for a source in the upper crust. (Source: Storchak et al., 2003)

Given an event's origin location, it is possible to predict the arrival time of each phase by considering the length and characteristics of the event–station path. P waves propagate through the crust at around 5-8 kilometers per second, compared to S waves at 3-4 km/s. Surface waves tend to arrive later than body waves because they take a less direct path and typically propagate at slower speeds. A model of phase travel times that considers only the event depth and event–station distance is known as a one-dimensional (1D) model; the IASPEI-91 model (Kennett and Engdahl, 1991) is an example.

The precise velocities of seismic waves may be highly nonuniform, varying significantly according to the local geology, so it is possible to improve on 1D travel-time models by tailoring predictions to the specific location of each event. This can be done by using historical data to estimate an explicit velocity field at each point (voxel) in the earth, and then calculating travel times by raytracing along the specific paths followed by each phase from a given event location to a given station. Such a model is known as a 3D travel-time model; an example is the LLNL-G3D (Simmons et al., 2012) model. Although 3D models can be significantly more accurate than 1D models, they are also more difficult to estimate and, once estimated, require more extensive computation to generate predictions.

In reality, seismic energy follows a continuum of paths, so that an exhaustive taxonomy of seismic phases can devolve into an arbitrary clustering rather than identifying truly natural categories. Nonetheless, discrete phase classifications and their associated travel time models are a useful tool for understanding seismic wave propagation, and are a central component of the monitoring systems we review in this chapter as well as our current work.

2.3 Observed waveforms

A seismic station records a continuous waveform, or seismogram, measuring ground displacement at each moment in time. This recording includes motion from incoming seismic phases (Figure 2.2) as well as ambient background noise. Phase arrivals generally register as

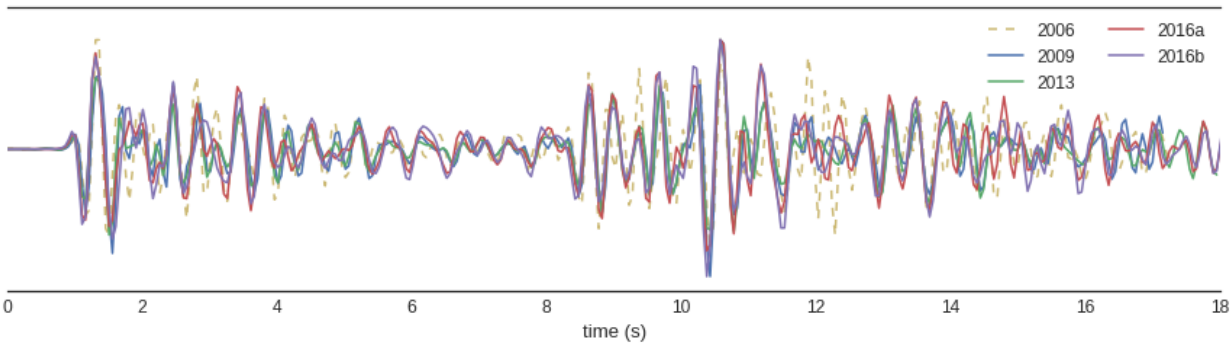


Figure 2.4: Correlated signals from P-wave arrivals of five known North Korean nuclear tests at Mudanjiang, China (station MDJ). The 2006 test (dotted) is weaker and noisier than the others but still shows significant correlation.

distinct spikes corresponding to the first wavefront arriving along the shortest path, followed by a gradual decay as additional energy arrives via longer paths and earlier arrivals continue to reverberate in the local geology (see the discussion of coda in Section 2.8). Background noise is generated by natural phenomena such as ocean waves and tidal fluctuations, flowing water in rivers, wind, low-level reverberations from previous seismic events, along with human activity including industry and road and rail traffic. Noise levels vary from station to station, and in some cases may follow daily or seasonal cycles.

The detailed fluctuations in signals from arriving phases are a function of the source mechanism as well as a path-dependent *transfer function*, in which seismic energy is modulated and distorted by the geological characteristics of the paths followed by each phase. Since event–station paths are themselves functions of the source location, events with similar locations and depths tend to generate highly correlated waveforms as long as the source mechanisms are not too different (Figure 2.4). The lengthscale at which such correlations are observed depends on the local geology and may range from hundreds of meters up to tens of kilometers. Since wave propagation is essentially deterministic and geological structures are essentially static on human timescales, significant correlations can be observed even from events occurring decades apart. A pair of nearby events generating correlated waveforms is known as a *doublet*.

Recorded seismic waveforms also depend on the response of the recording instrument itself. In this thesis we implicitly assume a linear station response, in that we model recorded waveforms as a simple sum of displacements from all incoming phases and a background noise process (eqs. (4.1) and (4.2)). We do not attempt to model an explicit response curve for each station, although the station response is captured implicitly to some extent by path-dependent signal models learned from historical data at each station (Section 4.6).

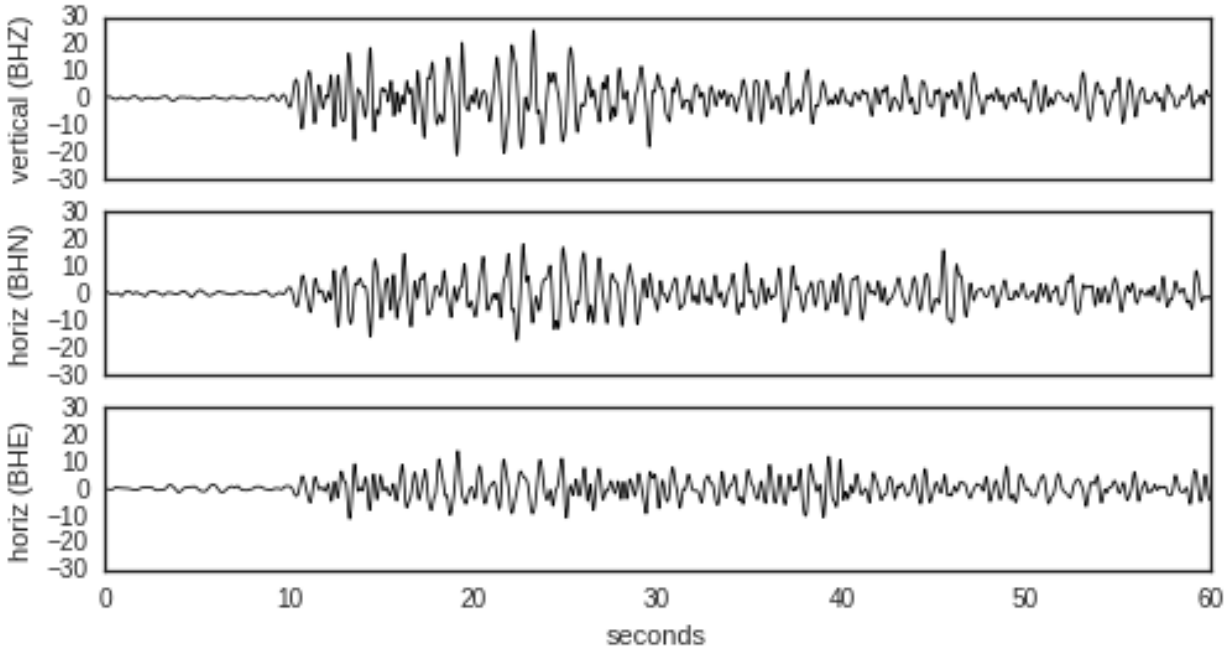


Figure 2.5: Signals recorded at a three-component station (FITZ), showing the P arrival of LEB evid 5270227.

2.4 Station types

Instead of just recording a single waveform, many seismic stations use multiple signals to infer the *direction* of incoming seismic waves. In particular, the IMS monitoring network uses both *three-component* and *array* stations, which take different approaches to estimating directional information.

Three-component stations measure ground displacement along three orthogonal axes: vertical, north–south, and east–west (Figure 2.5). A signal registering strong motion in one dimension may appear quiet along the others; in particular, compression and shear waves from a given source will generate motion in perpendicular directions to each other. Using all three components it is possible to estimate the angle of incidence of an incoming arrival using simple trigonometry; this is known as polarization analysis (Jurkevics, 1988). The angular component within the horizontal (surface) plane is known as the *azimuth*, while the vertical angle of incidence is typically parameterized in terms of the *slowness*, the reciprocal of signal velocity projected onto the horizontal plane. A wave arriving from directly below a station and traveling vertically has infinite slowness, while a wave traveling entirely in the horizontal plane has slowness equal to the reciprocal of its absolute velocity. Estimating slowness from a three-component station requires assumptions regarding the speed of signal propagation in the local medium. Azimuth and slowness estimates from three-component stations can also be compromised due to local scattering in the neighborhood of the receiver,

i.e., reflected waves that appear to come from directions other than their original source.

Much sharper directional information is available from array stations, which consist of multiple sensors distributed over a spatial region, with width ranging anywhere from tens of meters to tens of kilometers. Some array elements may themselves be three-component stations, while others record vertical motion only. As a signal traverses the array, it is recorded by each sensor in turn; tracking this progress allows for analysis to recover its azimuth and slowness. This is done by finding the velocity vector that maximizes the coherence of the signals measured at each array element; this is known as array *beamforming*, by analogy to array antennas that use the inverse process to *produce* a coherent beam aimed at a known target. By aligning and averaging waveforms from individual elements using the estimated velocity vector, array stations reduce the noise level while reinforcing signals from the arriving event (Le Bras et al., 2002).

2.5 Detection-based monitoring

We now describe the steps of a traditional detection-based monitoring pipeline, from "picking" individual phase arrivals at each station to network processing that associates and locates seismic events. We focus on global monitoring by the IMS/IDC as our motivating example, although similar architectures are also used in the regional monitoring networks run by individual states and other organizations.

2.5.1 Picking

The first step of a traditional monitoring pipeline involves extracting, from the continuous signal recorded at each station, a set of discrete *detections* corresponding to phase arrivals from seismic events. This process is known as picking, and is typically based on short-term-average/long-term-average (STA/LTA) processing, which computes the ratio of signal amplitudes in a rolling short-term window (on the order of three seconds) to a long-term window (on the order of 30s). The expected STA/LTA of a stationary process is unity, but the STA and thus the ratio increases sharply upon the arrival of a burst of signal energy (Figure 2.6). A simple detection algorithm triggers when the STA/LTA rises above some threshold; more sophisticated approaches such as *z-detection* estimate the background variance in order to set a threshold adaptively (Withers et al., 1998). To prevent spurious detections, further detections are typically suppressed following a trigger until the STA/LTA falls below some background threshold. The detections produced by an STA/LTA trigger are sensitive to the choice of these thresholds, which determine the tradeoff between detection sensitivity and avoiding false positives.

Once an arrival has been picked, additional processing is performed to estimate a precise arrival time, amplitude, azimuth, slowness, and other features; a classifier may be used to predict the phase type of the detection (Le Bras et al., 2002).

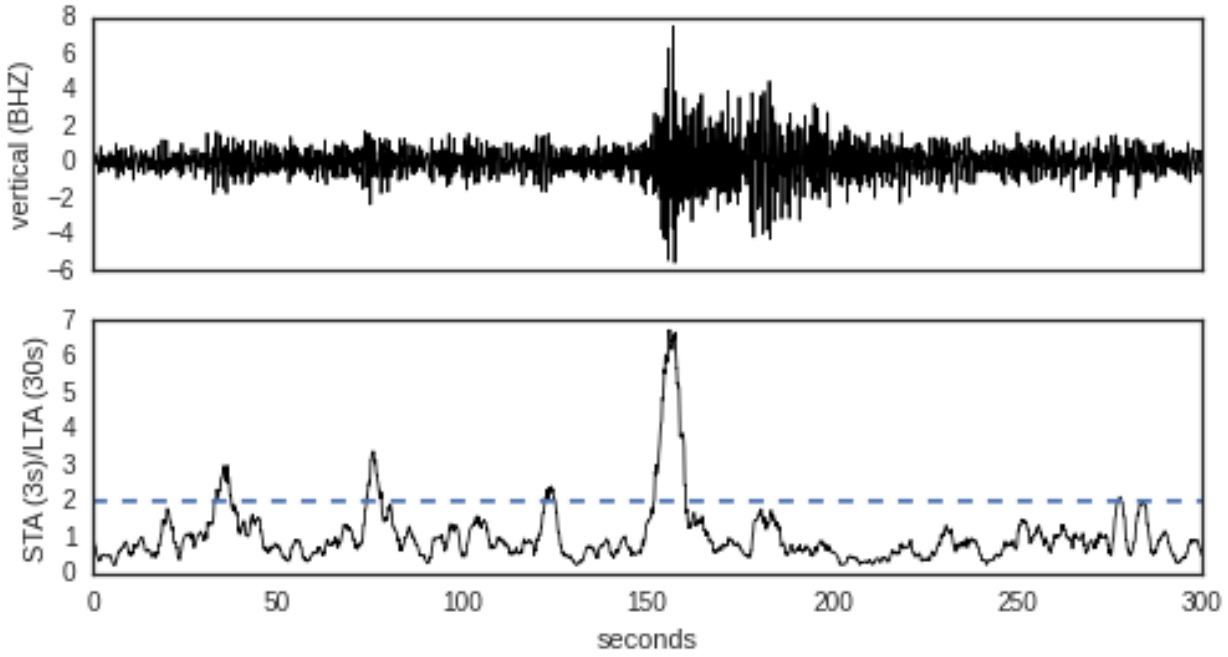


Figure 2.6: Example seismic waveform (top) processed using STA/LTA (bottom), with dotted line illustrating one possible detection threshold.

2.5.2 Event location

Given picked arrival times at multiple stations, it is possible to locate an event by *inverting* a travel-time model; that is, searching for an origin location and time such that the model-predicted arrival times match the observations. In the special case of three stations with known event–station distances, this process is known as *triangulation*; more generally we refer to location from multiple stations using a travel-time model as *multilateration*. Assuming constant velocity, observing a single phase (e.g., the direct P arrival) at a station constrains the origin to lie on a hypercone, i.e., a 3D surface embedded in 4D spacetime, containing for each possible origin time the set of all points whose event–station distance is consistent with the observed arrival time. Each additional station provides an additional constraint, so that in general four stations are required to localize an event. Solutions can also be constrained by incorporating azimuth and slowness information, by assuming that the event lies within the earth's crust, and by observing multiple phases (e.g., P and S) at a station. In some cases events may therefore be located even from a single three-component station (Magotra et al., 1987; Roberts et al., 1989).

Classical methods originating with Geiger (1912) treat travel-time inversion as a nonlinear least squares problem

$$\min_{\mathbf{x}} \sum_{i=1}^{\#stas} \sum_{j=1}^{\#phases} \left(t_{ij} - E[t_{ij}|\mathbf{x}]\right)^2, \tag{2.1}$$

where t_{ij} is the observed arrival time for phase j at station i and $E[t_{ij}|\mathbf{x}]$ is the model predicted arrival time for an event at space-time position $\mathbf{x}$. This minimization is typically solved by a sequence of iterated linear least squares updates. The least squares formulation implicitly assumes that the travel-time residuals are independent and Gaussian distributed. This independence assumption is often reasonable, but is violated when multiple event–station paths overlap, as with array elements (which for this reason are typically not treated as separate stations) or multiple events in a regional cluster.

Systems that perform *multiple-event location* can take advantage of correlations in travel time residuals among nearby events to locate a group of events more accurately than would be possible by considering each event individually. Particularly relevant to our current work is BAYHLoc (Myers et al., 2007), which formalizes multiple-event location as inference in a hierarchical Bayesian model, with travel-time correction terms at each station that are inferred jointly from data along with the event locations. Another approach, double-differencing (Section 2.7), uses waveform correlations to estimate precise *relative* arrival times that provide additional constraints for locating multiple events.

2.5.3 Network processing

Unfortunately, the detections produced by station picking do not come with event labels attached: detections recorded at two separate stations might be from the same event, two separate events, or simply random noise. Before applying the location techniques discussed above, a detection-based monitoring system must solve the *association* problem, determining which detections should be grouped together as arising from a hypothesized event, assigning specific phases to each detection, and which detections should be discarded as noise. This is a difficult problem with combinatorially many possibilities. In fact, association and location are interdependent problems, since given true event locations we can usually associate a plausible set of detections, and conversely, given correct associations it is straightforward to solve the location optimization (2.1).

The IMS treats association and location jointly under the heading of *network processing*, in which a global event bulletin is produced from the discrete detections recorded at each station. The *Global Association* (GA) system in use at the IDC does this via a complex heuristic algorithm consisting of multiple steps (Bras et al., 1994):

1. The earth is divided into a set of grid cells; the system searches for arrivals at the stations nearest to each cell, and uses these *driver* arrivals to predict the time of a hypothetical *seed* event in that cell.

2. For each seed event, the system searches for corroborating arrivals at additional stations. Each such arrival increases the score of the event; when the score passes a threshold (and other event definition criteria are satisfied) a preliminary event hypothesis is confirmed.

3. Event hypotheses are clustered, and redundant events (those whose arrivals are a strict subset of some other event's) are eliminated.

4. The system then performs several passes enforcing various consistency requirements. For example, each phase of an event can be detected only once at a given station, and each detection can be assigned to at most one event. As detections are reassigned, event locations are iteratively recomputed based on their new sets of associations. If an event's score falls below the definition threshold, it is removed and its associated arrivals made available to other events.

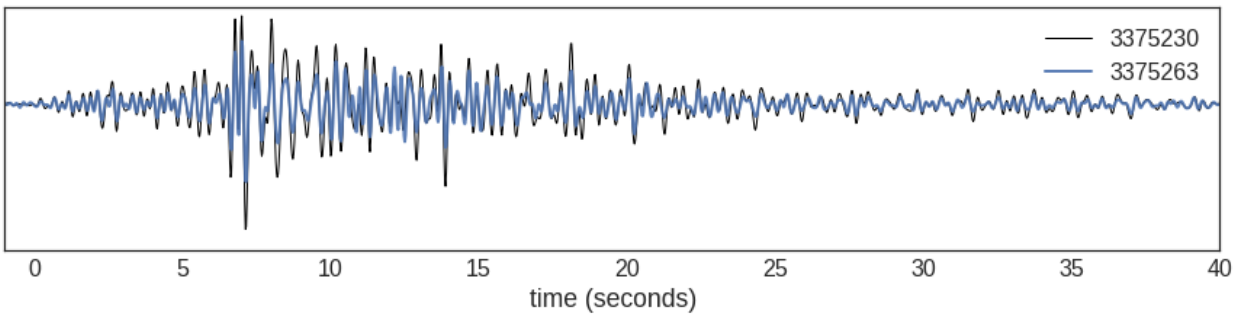
This process at the IDC produces a series of event bulletins, the *Standard Event Lists* SEL1, SEL2, and SEL3, of which SEL3 is the final and most complete.

An alternate approach, implemented by NETVISA (Arora et al., 2013), is to treat network processing as a problem in Bayesian inference (Figure 1.2). NETVISA specifies a generative probability model (Section 3.1.2) of seismic events and detections, so that any hypothesized bulletin can be assigned a score corresponding to its posterior probability under the model. The task of producing an event bulletin is then reduced to a simple hill-climbing search, in which events are added, removed, and relocated in order to maximize the posterior probability of the resulting bulletin. As argued in Section 1.2, this approach is conceptually and philosophically attractive, yielding a principled model-based objective that takes all available data into account, and separating the construction of an explicit domain model from the design of the search algorithm used to produce a bulletin. It has also shown concrete practical advantages in monitoring performance, including significant improvements in location accuracy and a 60% reduction in missed events compared to SEL3 (Arora et al., 2013).

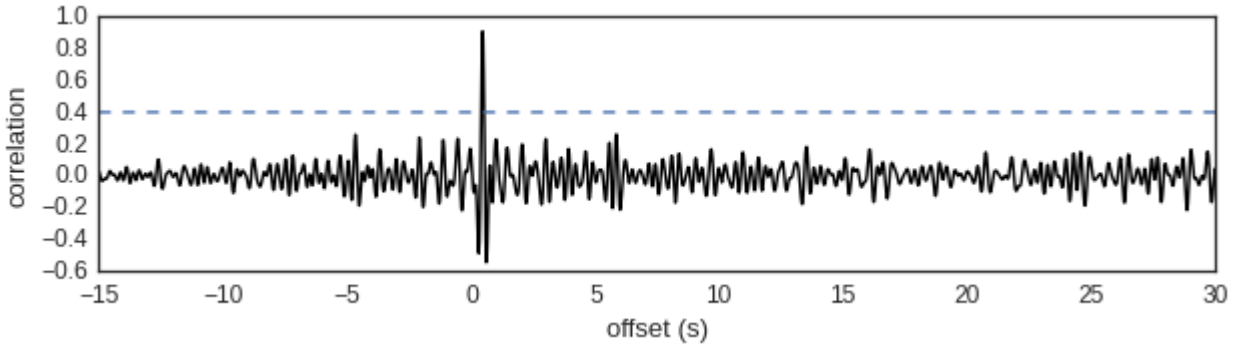
Other recent work, by Ballard et al. (2015), attempts to bridge the gap between detection-based monitoring systems such as GA and NETVISA, and the signal-based approach pursued in our current work. Their "auto analyst" system initially builds events following a classical approach similar to GA, then reinspects the signals observed at predicted phase arrival times to potentially add and associate additional detections that were missed by initial station processing. It also builds events using waveform correlation (Section 2.6), and can associate these events with detections that might otherwise generate false or mislocated events.

2.5.4 Analyst review

The final step in the IDC processing pipeline involves review of the automated SEL3 bulletin by a team of human analysts. The analysts examine each automatically built event to evaluate its associated arrivals, inspecting the signals with visualization tools including frequency filtering, beamforming, polarization analysis, and phase arrival predictions. Arrivals may be added, removed, retimed, or renamed as different phases, and events removed or their locations updated accordingly. Analysts may also use their own prior knowledge to constrain locations; for example, ruling out a deep event in a region where such events do not historically occur.



(a) Aligned signals (0.8-4.5Hz) showing correlation over a 40-second window.



(b) Cross-correlation trace from sliding a 30-second window of one signal against the other, with peak of 0.91 corresponding to alignment in Figure 2.7a.

Figure 2.7: Waveform correlations between a doublet pair from the Wells, NV aftershock sequence, ISC evids 3375230 and 3375263, recorded at IMS station ELK.

Analysts may also add new events that were not built by automated processing. For example, given a large event expected to produce aftershocks, they will align the signals of nearby stations to search for aftershock events, which are then added and any relevant detections associated so that they do not clutter other events. However, in general it is much easier for analysts to remove false events than to build new events from scratch; analyst review cannot be relied upon to catch events missed by automated processing.

Events reviewed by analysts form the Late Event Bulletin (LEB). Those that meet certain definition criteria are also included in the more strict Reviewed Event Bulletin (REB). For terrestrial events (those occurring in solid rock, as opposed to water or air) these criteria include a baseline of three P-type arrivals at primary seismic stations, with an additional weighted score threshold requiring reliable azimuth and slowness at two stations, or if that is not available, arrival times from additional stations (Le Bras et al., 2002).

2.6 Waveform correlation

An alternative to detection-based monitoring is the use of *waveform correlation* to detect and locate seismic events (Anstey, 1964). These methods exploit the fact, discussed above, that waveforms observed from events in nearby locations tend to correlate with each other, in some cases quite strikingly.

This effect can be exploited to detect events that would fall below a conventional picking threshold, and to locate those events precisely, even from a single station. To detect events, a historical waveform template $\mathbf{x}$ is slid over the incoming signal $\mathbf{s}$, and the normalized cross-correlation

$$xc(\mathbf{x}, \mathbf{s_t}) = \frac{\mathbf{x}^T \mathbf{s_t}}{\|\mathbf{x}\| \|\mathbf{s_t}\|},$$

is computed for a window at each index t. Correlation above some threshold indicates the potential presence of an event (Figure 2.7b). Furthermore, from this one observation, the location of the detected event may be assumed to lie near to the historical event from which the template was formed; in some cases this determines the location quite precisely (using a dataset from China, Schaff and Richards (2004) found that doublets meeting a strict correlation criterion were no more than 1km apart).

Correlation methods have shown promising results in large-scale event detection and location. Gibbons and Ringdal (2006), among others, have demonstrated detections of events that would otherwise be buried in noise, with Schaff and Waldhauser (2010) providing evidence that correlation detectors can lower detection thresholds by a full magnitude unit compared to pick-based systems. Schaff and Richards (2011) and Waldhauser and Schaff (2008) argue that around 13% of seismicity in both China and California can be precisely located from unambiguous correlations, while Slinkard et al. (2013) find that between 24% and 92% of events across several aftershocks can be recognized as doublets. Gibbons and Ringdal (2012) and Schaff et al. (2012) claim the ability to detect and discriminate very low-yield explosions at the DPRK (North Korean) test site using a single array. More recent work using multiple stations has suggested that large-scale correlation monitoring could almost double the size of the LEB (Schaff et al., 2012).

A clear drawback to correlation-based methods is that they cannot detect *de novo* events, i.e., events occurring in regions with little previous seismicity. This means that a correlation detector in isolation is not a reliable nuclear monitoring system, since an adversary can choose to locate their tests in areas with no historical coverage (though the presence of an event in such an unusual location may itself be seen as suspicious, assuming it can be detected by other means). However, correlations may still play a valuable role, both in detecting events at existing test sites (Figure 2.4), and in helping to "explain away" arrivals from weak events detected at only one or two stations, which would otherwise clutter and potentially confuse a network processing system as it tries to associate them with other events. Metaphorically, if we view nuclear monitoring as the search for a needle in a large haystack of detections, correlation methods may allow us to throw away much of the "hay", corresponding to known seismicity, so that any remaining needles are clearly visible.

Another drawback is that, since correlation methods do not formally model the source mechanism or transfer function, it is not obvious how to account for source effects caused by, e.g., events of different magnitude yielding different frequency spectra, or an explosion source in a region with only historical earthquake data. Even putting these aside, more pedestrian questions arise:

- How should we set the correlation threshold in a principled way to avoid false triggers? Are different thresholds appropriate for events in different regions?
- How long should the correlation window be, and which phases should it cover? Concretely, given a template that correlates strongly for 10 seconds, versus another that correlates more weakly for 60 seconds, which should we prefer?
- Should we correlate against all historical events in a region, some of which may be noisy or atypical, or attempt to combine them into a master "prototype" template that potentially discards useful information?
- What can we conclude quantitatively about the location of an event that correlates weakly with a historical template? If correlation evidence places an event in one location, while travel-time evidence suggests a different location, how can we resolve this ambiguity?
- How likely are false positives, i.e., signals that correlate over some period of time but are produced by events in very different locations?
- How should we interpret correlation evidence from multiple stations? For example, is an event with 0.7 correlation at one station more plausible than another with 0.4 correlation at three stations? What about an event that correlates well at station A, but not at another station B, where similarly-located events have historically shown strong correlations?

Research into correlation-based monitoring (including work cited above) has attempted to approach some of these issues by developing empirically motivated guidelines (Schaff et al., 2004) as well as more sophisticated techniques such as subspace detection (Harris, 1997). However, there is as of yet no clear guiding principle for designing correlation detectors or integrating them into a larger monitoring framework.

In this thesis we argue that these difficulties stem fundamentally from the fact that correlation is not a model-based procedure; it does not make explicit its assumptions about the data generating process, and provides no mechanism for quantifying the uncertainty in the inferences it produces. One contribution of our work is an attempt to put correlation methods on a more solid methodological footing, by constructing a Bayesian model in which each of the questions posed above has a well-defined quantitative answer. This avoids the need for arbitrary correlation thresholds, allows for fusion of correlation evidence from multiple stations, and naturally implements the "explaining away" effect by performing joint inference using both correlation and travel-time evidence.

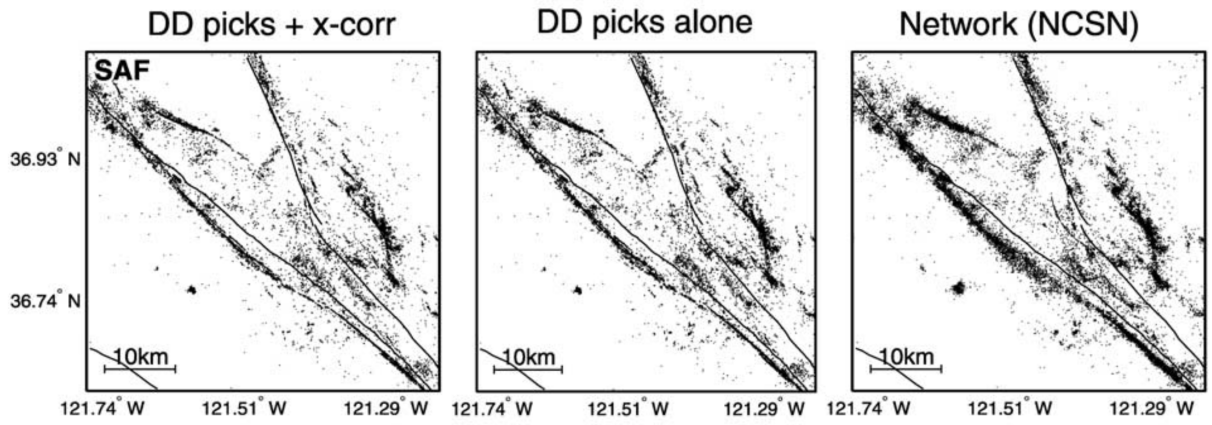


Figure 2.8: Double-difference event relocation along the San Andreas fault. Relative to the original single-event locations (right), joint relocation using double-differencing is able to resolve fault structures at a much finer level (center), with additional detail revealed using precise relative arrival times from waveform cross-correlation (left). Lines indicate surface-mapped fault lines. (Source: Waldhauser and Schaff, 2008)

In addition to the full SIGVISA model, which implements these properties as part of a network-level joint inference calculation, Appendix B also develops a class of simple statistics that may be applied at the signal-processing level, that attempt to preserve the simplicity and tractability of cross-correlation while maintaining a principled probabilistic interpretation.

2.7 Double difference relocation

In addition to detection and location, cross-correlation can be used as a source of very precisely picked *relative* arrival times. While traditional automated picking resolves arrival times to within one or two seconds, aligning two waveforms at their cross-correlation peak (assuming this is unambiguous) determines the difference in their arrival times to within about a tenth of a second. *Double-differencing* (Waldhauser and Ellsworth, 2000) exploits these precise relative arrival times to jointly locate multiple events. That is, it adds an additional set of terms to the basic travel-time inversion (2.1), minimizing the squared residuals between the *difference* in arrival times $d_{ijkl} = t_{ijk} - t_{ijl}$ for each pair of events k and l (at station i and phase j), which is measured precisely using cross-correlation, and the difference in arrival times that would be predicted by a travel-time model,

$$\min_{\mathbf{x}} \sum_{i,j,k,l} \left(d_{ijkl} - (E[t_{ijk}|\mathbf{x}_k] - E[t_{ijl}|\mathbf{x}_l])\right)^2 . \tag{2.2}$$

As with the basic travel time inversion, in general these terms may be weighted to account for the confidence of each measurement. Since the addition of a new event to this system can

change the locations inferred for earlier events, the double-difference inversion is generally performed as a batch *relocation* of catalog events that have already been detected by conventional means. Although relative arrival time constraints may not improve event locations in absolute terms, the relocated events generally have much more accurate *relative* locations, allowing them to present a much sharper picture of geologic structure such as fault lines (Figure 2.8).

As with the travel time inversion (2.1), a naïve double-difference inversion (2.2) implicitly assumes that difference residuals are Gaussian and independent. Waldhauser and Ellsworth (2000) attempt to relax these assumptions by heuristically reweighting terms during an iterative least-squares solution procedure, downweighting terms with large residuals or high inter-event distances, though it is not clear whether this scheme in fact corresponds to any coherent model of the distribution of difference residuals.

2.8 Models of seismic signals

Although the monitoring methods described in this chapter are primarily model-free, there is a wide body of work on explicitly modeling seismic signals and the processes that generate them (indeed, in some sense this is the main project of the entire field of seismology!). We first note the literature on *synthetic seismograms* (Helmberger, 1983), in which an explicit source model is propagated forward through an explicit earth model using direct numerical simulations such as finite-difference (Kelly et al., 1976) or finite-element methods (Komatitsch and Vilotte, 1998; Shearer, 2009). For well-understood sources in geologically uncomplicated areas, these methods can produce realistic signals, but they are hampered in general by the inability to represent uncertainty over the source and earth models; in most cases these models can be inferred only roughly from available data. The computational complexity of these numerical simulations also makes them impractical to incorporate in a real-time monitoring system, at least at present. In the long term, we might hope for a convergence between direct numerical simulations and signal-based Bayesian models such as the work in this thesis, as Bayesian models are developed to incorporate more fine-grained physical structure, or physical simulations extended to provide robust predictions in the face of uncertainty over exact conditions.

Another approach to modeling seismic signals is treat the waveforms themselves as stochastic, and instead attempt to represent only higher-level features such as the shape of the signal envelope. One line of work in this regard involves models of the envelope *coda*, the long decay following the initial peak of a direct phase arrival (Aki, 1969; Sato et al., 2012). The coda represents the scattered wave field containing the "echo" of the initial arriving wave front; while the initial signal peak represents energy arriving directly from the epicenter, and is highly contingent on path-specific effects, the scattered energy in the coda arrives from all directions and is relatively stable across origin locations (Mayeda, 1993; Mayeda et al., 2003). Modeling the coda statistically, as the result of scattering by random local heterogeneities, allows for the derivation of analytic forms for the envelope shape that

can provide strong empirical fits to observed signals, especially within narrow frequency bands. The repeatability of coda amplitudes makes them particularly useful for estimating event magnitudes and explosion yields. Mayeda et al. (2003) model the S wave coda within a range of narrow frequency bands for events in the Dead Sea Rift area, using the envelope form

$$A(t) \propto H(t - t_0) \cdot (t - t_0)^{-\gamma} \cdot \exp\left(-b(t - t_0)\right) \tag{2.3}$$

in which H is the Heaviside step function, t_0 the arrival time, and γ and b are parameters controlling polynomial (short-term) and exponential (long-term) decay rates. They find that by calibrating the decay parameters to historical signals, as a function of event–station distance, it is possible to obtain magnitude estimates that are significantly more consistent between stations than by using the direct arrival alone.

Extending that work, Pasyanos et al. (2012) apply the same coda form to estimate magnitudes jointly from multiple regional phases. They model signal envelopes as an additive combination of a noise process with contributions from individual phase arrivals,

$$A = A_{\text{noise}} + A_{\text{Pn}} + A_{\text{Pg}} + A_{\text{Sn}} + A_{\text{Lg}},$$

which matches the overall structure of the SIGVISA envelope model (4.1), although, due to their target applications of magnitude and source spectrum estimation rather than a full monitoring system, they consider only a step-function onset and restrict themselves to modeling envelopes rather than detailed waveform fluctuations.

Another envelope model is proposed by the Virtual Seismologist of Cua (2005), which implements an earthquake early warning system based on a Bayesian model of ground motion envelopes. Here the P and S-wave envelopes are parameterized by an arrival time t_0, rise time t_{rise}, amplitude A, duration Δt, and two decay parameters γ, τ, with envelope shape

$$E(t) = \begin{cases} A(t - t_0)/t_{\text{rise}} & \text{if } t_0 \le t < t_0 + t_{\text{rise}} \\ A & \text{if } t_0 + t_{\text{rise}} \le t < t_0 + t_{\text{rise}} + \Delta t \\ A(t - t_0 - t_{\text{rise}} - \Delta t - \tau)^{-\gamma} & \text{otherwise } (t > t_0 + t_{\text{rise}} + \Delta t) \end{cases}, \tag{2.4}$$

modeled as an initial linear onset, followed by a constant plateau, followed by a polynomial decay with rate γ and offset τ. The decay formulation differs from the exponential rate seen in the coda modeling literature, as in eq. (2.3), and appears to be inspired by Omori's law (Utsu, 1961) governing the frequency of aftershocks following a large event. The logarithm of each envelope parameter is modeled as a linear function of magnitude and event–station distance, similarly to the parametric components of the envelope models in Section 4.5 of this work, though our model also includes nonparametric (location-specific) components.

Chapter 3

Technical background

This chapter surveys the mathematical tools used to construct SIGVISA. We begin with a high-level discussion of probabilistic modeling and inference, leading to the concrete framework of Markov chain Monte Carlo (MCMC) as our inference algorithm of choice. We then review several mechanisms used as components of the SIGVISA model: Gaussian processes, state space models, autoregressive processes, and wavelet transforms.

3.1 Probabilistic modeling

A model attempts to represent the real, messy world in terms of idealized quantities whose relationships can be precisely defined. It is, by design, a simplification of the system it represents; a model that does not simplify is like the "map of the Empire whose size was that of the Empire, and which coincided point for point with it" (Borges, 1998) — impressive but useless. A good model identifies a set of abstract quantities whose interactions capture the important phenomena of interest while remaining simple enough to provide insight. "All models are wrong, but some are useful" (Box, 1976).

Many models in science, particularly classical physics, are deterministic: given exact knowledge of initial conditions, they specify a precise trajectory over future states. At higher levels of abstraction, however, determinism becomes impossible. To predict the precise waveform produced by a particular seismic event at a given station would require an infinitely fine-grained representation of earth structure, perfect descriptions of the source mechanism and detecting equipment, and exact knowledge of all possible noise sources, not to mention an impossible amount of computation. Practical modeling of complex phenomena therefore necessarily involves uncertainty: if a model must be simpler than the real world, and so cannot represent the world's full state, then the model cannot hope to make exact predictions; the best it can hope for is to accurately represent the space of possibilities. Thus probabilistic models are appropriate, and indeed necessary, even when a system's underlying dynamics are truly deterministic.

3.1.1 Probability theory and notation

Probability theory is the mathematical formulation of uncertainty, and forms the basis of all probabilistic modeling. A rigorous development of probability theory including the machinery of probability measures, sample spaces, conditioning, and random variables is beyond the scope of this thesis — see, e.g., Billingsley (2008). We do make use of some common notational conventions that merit explanation. We will often write $p(x)$ as shorthand for the density of a random variable X evaluated at a specific value x, which would more properly be written $p(X = x)$ or $p_X(x)$. We will also speak somewhat loosely regarding the subtleties of measures, mass functions, and densities, using the same notation $p(x)$ for discrete mass functions and continuous densities, and occasionally referring to densities as "probabilities" where the distinction is not technically crucial.

We will generally use bold uppercase symbols $\mathbf{X}$ for matrices and lowercase $\mathbf{x}$ for (column) vectors. For a fixed length signal defined at discrete time steps 0 through T, we use both function notation $s(t)$ and vector notation $\mathbf{s}$ interchangeably.

We will use the notation $\mathcal{N}(\mathbf{x}; \mu, \mathbf{\Sigma})$ to refer interchangeably to a Gaussian distribution defined on a random variable $\mathbf{x}$, with mean μ and (co)variance $\mathbf{\Sigma}$, or to the density (A.1) of such a distribution. We will occasionally simplify this to $\mathcal{N}(\mu, \mathbf{\Sigma})$ when it is clear from context which variable is being described.

3.1.2 Generative models

A model's representation of the world can be partitioned into unobserved quantities x and observed quantities y. For example, x might represent the set of seismic events occurring in a particular time period, and y the waveforms recorded across a network of stations. Typically we are interested in inferring the latent quantities x given our observations; in probabilistic terms we want the conditional distribution $p(x|y)$. It is occasionally possible to specify an appropriate form for this distribution directly; this is known as a *discriminative* model. Many off-the-shelf machine learning techniques, including linear and logistic regression, support vector machines, and supervised neural networks, are discriminative models. Much more commonly, however, it is not immediately obvious what form this conditional should take; e.g., we do not expect that the function that maps from seismic signals to event latitudes and longitudes is expressible as some simple linear transformation of the seismic signal.

Generative modeling is a strategy in which we first specify specify a joint distribution $p(x, y)$ over all possible worlds, and then apply the rules of probability theory to *derive* the desired conditional distributions. Building generative models allows us to develop domain-specific learning procedures that are informed by the causal structure of the domain at hand, without assuming generic functional forms or discriminative rules.

Typically the distribution over possible worlds is specified by describing a *generative process* in which the state variables are sampled in sequence, each conditioned on the values of the variables that came before. This is simply the chain rule from probability theory,

which states that a joint distribution can be written as a product of successive conditionals:

$$p(x, y, z, \ldots) = p(x)p(y|x)p(z|x, y)\ldots.$$

The chain rule allows us to build complex joint distributions as a product of simpler factors. Although not required, this form is especially natural when the generative process is *causal*, so that the conditional distributions specify the process by which each variable is generated given its predecessors. The variables at the beginning of the causal chain are sampled from an unconditional distribution $p(x)$ over initial conditions.

In Bayesian statistics, we often consider a *prior distribution* $p(x)$ over unknown quantities and a *forward model* or *likelihood function* $p(y|x)$ that describes how our observations are generated, given the unknowns. This encodes a generative model with joint distribution $p(x, y) = p(x)p(y|x)$. Now suppose we are interested in the conditional distribution $p(x|y)$ on latent quantities given observed data; since this inverts the forward model, problems of this type are sometimes called *inverse problems*. Simply applying the definition of conditional probability

$$p(x|y) = \frac{p(x, y)}{p(y)} = \frac{p(y|x)p(x)}{p(y)}, \tag{3.1}$$

we recover *Bayes' rule* (3.1) which gives the solution to inverse problems under uncertainty. The resulting conditional $p(x|y)$ is known as the *posterior distribution* and represents the available knowledge about the unknown quantities x, accounting for noisy measurements as well as the uncertainty inherent in the model (Gelman et al., 2014).

Generative processes involving a fixed set of random variables can be represented as directed acyclic graphs, sometimes called Bayesian networks (Koller and Friedman, 2009). To generate a possible world, each variable is sampled conditioned on its parents, with root nodes sampled from a prior distribution. By the chain rule, any joint distribution can be represented using a graph in which each variable depends on all previous variables, but in many models the natural causal structure is such that many variables depend directly only on a small number of parents; in this case the graph will imply certain conditional independence relationships among the variables. This added structure can be useful in deriving efficient inference algorithms, and in improving statistical efficiency by reducing the number of parameters that must be estimated to specify the model.

Models with an unknown or varying number of random variables are known as *open-universe probability models* (OUPMs) (Milch and Russell, 2010). The SIGVISA model described in this thesis is an OUPM, since the number of seismic events is not known in advance. OUPMs are naturally specified as *probabilistic programs*: generative processes defined by computer code, potentially including loops and conditional branches, rather than a fixed graph structure. A probabilistic program samples a possible world as it executes; the distribution over execution traces therefore corresponds to a distribution over possible worlds. The NETVISA model was originally written as a probabilistic program in Bayesian Logic (Milch et al., 2005); the SIGVISA generative story told in Section 4.2 is, in effect, an informally-specified probabilistic program.

3.2 Inference and MCMC

The task of computing the posterior in a generative model is called *inference*. By "computing" we mean representing the posterior in some form that makes it tractable to take expectations of arbitrary functions under the posterior distribution, or at least to extract a mean or other point estimate of the latent quantities. This form can be either an explicit probability density function or a set of samples from which expectations can be approximated using Monte Carlo averaging.

From an algebraic standpoint, the main difficulty of inference is the denominator of Bayes' rule, the normalizing constant or *marginal likelihood* $p(y)$, defined by the integral

$$p(y) = \int p(x, y)dx,$$

which is hard to compute in general because it sums over all possible latent states x. In special cases it can be computed analytically; more generally, *variational inference* attempts to approximate the marginal likelihood as the solution to an optimization problem (Bishop, 2006). However, for complex models where the posterior distribution has no simple algebraic form, it is often preferred to sidestep direct computation of the marginal likelihood and represent the posterior implicitly by a set of samples.

Commonly applied for this purpose are *Markov chain Monte Carlo* (MCMC) algorithms, which draw samples from a distribution π by simulating sample paths of a Markov chain having π as its stationary distribution (Brooks et al., 2011). A general approach for constructing such Markov chains is the *Metropolis–Hastings* (MH) rule, in which at each step a new state of the world x' is sampled from a *proposal distribution* $q(x'|x)$, and then accepted with probability

$$\alpha(x', x) = \min\left\{1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right\}, \tag{3.2}$$

where π is the target distribution, typically the posterior $\pi(x) \propto p(x|y)$. The advantage of this formulation is that $\pi(x)$ is only required up to a normalizing constant, relieving us of the need to compute the marginal likelihood. It can be shown that the Markov chain defined by the Metropolis–Hastings rule satisfies a condition known as *detailed balance*, which implies that it preserves the stationary distribution π (Andrieu et al., 2003). Showing convergence to the stationary distribution requires the additional condition of *irreducibility*, which states that every state is reachable from every other after a finite number of steps with nonzero probability. This is generally satisfied as long as the proposal distribution has support on the entire state space, so that any state can in principle be proposed from any other. Even given these conditions, however, there is generally no guarantee that the chain will converge (or "mix") to its stationary distribution particularly quickly, and indeed it is often difficult to tell whether an MCMC chain used in practice is mixing.

Rather than a single proposal distribution q which updates the entire model's state at once, it is common to construct multiple proposals q_i each of which updates a single

variable or a small set of variables. Applying an MH acceptance step to each individual proposal guarantees that they respect the desired stationary distribution, and thus that the *cyclic* chain constructed by applying these proposals in a fixed sequence also preserves the stationary distribution.

A common "default" proposal for continuous variables is the *random-walk* proposal, which proposes a new value from a distribution centered at the current value, typically a Gaussian:

$$q_{RW}(x'|x) = \mathcal{N}(x'; x, \sigma_n^2). \tag{3.3}$$

The acceptance rate of such a proposal depends on the variance σ_n^2, which typically must be tuned (by hand or with an adaptive algorithm) to ensure adequate mixing. Note that since this proposal is symmetric, i.e., $q_{RW}(x'|x) = q_{RW}(x|x')$, the proposal densities cancel from the MH acceptance probability (3.2), yielding the simpler expression

$$\alpha(x', x) = \min\left\{1, \frac{\pi(x')}{\pi(x)}\right\}. \tag{3.4}$$

3.2.1 Reversible jump MCMC

Standard treatments of MCMC assume models containing a fixed number of random variables. While the same machinery is applicable to open-universe models, this requires some extra care.

To understand the issues that arise, consider a fully discretized model π_d of seismic events, in which the space of possible events (intuitively, the surface of the earth, plus additional dimensions for time and magnitude) is gridded into many small bins, so each event is represented simply as a discrete index denoting the bin in which it lies. Given a prior on the number of events (also a discrete quantity), the hypothesis space of this model consists of a countable number of states $\mathbf{x}$, each of which corresponds to some event history, i.e., a collection of bin indices, where each state has some finite probability $\pi_d(\mathbf{x})$. To birth or kill an event is simply to move from one discrete state $\mathbf{x}$ to another state $\mathbf{x}'$ containing a different number of events, and this proposal can be evaluated using standard Metropolis–Hastings machinery. The fully discrete model effectively has no notion of dimensionality.

In practice, it is more convenient to define a model that includes continuous-valued variables. We can view this as the continuous limit of the discrete model previously described, in which the bins shrink to infinitesimals. Although this is conceptually straightforward, technical issues arise as we move from probability mass functions in the discrete case to density functions in the continuous case. The essential issue is that densities defined on spaces of different dimension are not directly comparable; to form probabilities from these densities we are required to integrate over local balls of different dimensionality.

Reversible jump MCMC (Hastie and Green, 2012) is a formalism for constructing MCMC chains that jump between continuous spaces of varying dimension. In its most general form, it separates the proposal into a set of r elementary random values $\mathbf{u}$ sampled from a known density g and a deterministic, invertible, differentiable transformation $h : \mathbb{R}^{n \times r} \to \mathbb{R}^{n' \times r'}$,

where $(\mathbf{x}', \mathbf{u}') = h(\mathbf{x}, \mathbf{u})$; that is, applying the transformation to the current state and random u gives a proposed state $\mathbf{x}'$ as well as a set of values $\mathbf{u}'$ that would be needed to invert the transformation, $(\mathbf{x}, \mathbf{u}) = h'(\mathbf{x}', \mathbf{u}')$. Fully specifying the inverse move also requires us to choose a distribution g' from which to generate $\mathbf{u}'$. Note that for h to be invertible it is necessary that the total dimensionality remains constant, $n + r = n' + r'$, but n and n' may differ.

In this setting, the standard MH acceptance rule is directly applicable with the addition of a Jacobian factor ∂h,

$$\alpha = \min\left(1, \frac{\pi(\mathbf{x}')g'(\mathbf{u}')}{\pi(\mathbf{x})g(\mathbf{u})}\left|\frac{\partial h(\mathbf{x}, \mathbf{u})}{\partial(\mathbf{x}, \mathbf{u})}\right|\right),$$

which is introduced by the change of variables from $(\mathbf{x}, \mathbf{u})$ to $(\mathbf{x}', \mathbf{u}')$. Although this Jacobian factor is necessary in general, in many cases we can construct g to sample values with the same units as $\mathbf{x}$, so that h does not need to change parameterizations and its Jacobian becomes the identity. This is the case for all of the moves used in SIGVISA.

3.3 Parametric and nonparametric models

Often our models involve uncertainty not just over specific unknown quantities, but the *functions* that relate those quantities to each other. More concretely, in a directed probability model it is common to write the conditional distribution of a variable y given parents $\mathbf{x}$ as a noisy function of the parents, i.e., in the form

$$y = f(\mathbf{x}) + \epsilon$$

for some function f, where ϵ is a noise variable (e.g., a zero-mean Gaussian). If the dependence f is not known a priori, it must be inferred from data, i.e., itself treated as uncertain within the model.

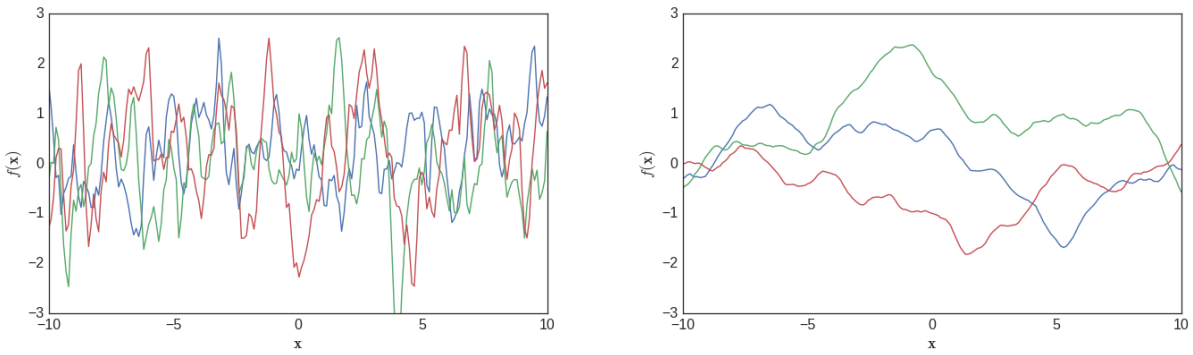
One approach is to represent the unknown function f *parametrically*, that is, in a fixed form defined by some set of parameters $\mathbf{w}$. For example, using a linear parameterization,

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}, \tag{3.5}$$

the task of estimating $\mathbf{w}$ to recover the function f is just standard linear regression. More generally, we can consider models that are linear in some set of *features* $\phi(\mathbf{x})$,

$$f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}), \tag{3.6}$$

or any number of nonlinear parameterizations such as the envelope models of Section 2.8. Choosing a parameterization is equivalent to imposing a (hard) prior on the function f, in that it restricts the hypothesis class to those functions expressible in our chosen parameterization. This enables statistically efficient learning and generalization assuming the true



(a) Short lengthscale ($\ell = 0.3$) (b) Long lengthscale ($\ell = 3.0$)

Figure 3.1: Samples from a GP with Matérn kernel ($p = 1$, $\sigma_f^2 = 1$).

function follows the parameterized form. When the parametric assumption is false, however, we will never recover the true function even given an infinite amount of data.

By contrast, *nonparametric* models make use of a representation that grows with the available data, so that (under reasonable conditions) they converge to the true function in the limit of infinite data. A simple example are nearest-neighbor models, which represent a function by memorizing all previously observed values, and predict new values at test points by simply regurgitating the closest historical observation, or an average of nearby observations. In the next section we introduce a more sophisticated class of nonparametric models, Gaussian processes, which allow us to frame function learning as Bayesian inference, and therefore incorporate uncertainty over functions into larger probability models in a principled way.

3.4 Gaussian processes

Gaussian processes (GPs) are a class of distributions on real-valued functions; they provide formal machinery for representing uncertainty over and performing inference on very general classes of functions, potentially defined only by weak structural assumptions such as smoothness. This makes them useful in Bayesian modeling when considering functions that do not fall into neat parametric classes. For example, this work attempts to model the function from a seismic event's coordinates (longitude, latitude, and depth) to the waveforms it produces at a particular station; this is certainly not a linear function of the coordinates! Nonetheless this function does have some structure: we expect that events near to each other will produce similar waveforms, which is a sort of smoothness assumption, and we would like our model to represent this belief.

Historically, GPs have been developed independently in several disciplines, beginning with the mathematical formalization of stochastic processes (Doob, 1953) and the construction of Brownian motion (Wiener, 1949). In geostatistics, Gaussian process regression is

known as *kriging* (Krige, 1951) and is viewed as a form of weighted interpolation. More recently, GPs have been understood in machine learning as a tool for nonparametric Bayesian regression and a probabilistic analogue to kernel methods such as support vector machines (Rasmussen and Williams, 2006).

Formally, a GP distribution on a scalar-valued random function $f(\mathbf{x})$ is characterized by a mean function $\mu(\mathbf{x})$ and a covariance function, or *kernel*, $k(\mathbf{x}, \mathbf{x}')$, such that for any finite set of inputs

$$\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})^T,$$

the random vector $f(\mathbf{X}) = (f(\mathbf{x_1}), f(\mathbf{x_2}), \ldots, f(\mathbf{x_n}))^T$ is multivariate Gaussian distributed with mean vector $\mu(\mathbf{X})$ and covariance matrix $k(\mathbf{X}, \mathbf{X})$:

$$f(\mathbf{X}) \sim \mathcal{N}(\mu(\mathbf{X}), k(\mathbf{X}, \mathbf{X})),$$

where $\mu(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ are functions defined on individual datapoints and pairs of points, respectively, and overloaded to vector inputs in the natural way. Note that the distribution over any subset of input points can be formed by simply dropping all but the relevant entries of the mean and covariance matrix; this parallels the result (A.4) for marginals of finite Gaussian distributions. In this sense a GP can be thought of as an infinite-dimensional extension of the multivariate Gaussian distribution.

In practice, the mean function μ is typically assumed to be zero, without loss of generality since this corresponds simply to a constant shift. By contrast, the covariance function k expresses our prior beliefs about the properties of f, and can be chosen to encode assumptions of smoothness, periodicity, symmetry, linearity, low dimensionality, or sophisticated combinations of these and other properties (Duvenaud, 2014). In this thesis we focus on the Matérn covariance (Rasmussen and Williams, 2006), which has historically been popular for geophysical applications, and is defined by

$$k_{Matern}(r) = \sigma_f^2 \exp\left(-\frac{r\sqrt{2p+1}}{\ell}\right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{r\sqrt{8p+4}}{\ell}\right)^{p-i}. \tag{3.7}$$

This is a *stationary* covariance, meaning that it depends only on the distance $r(\mathbf{x}, \mathbf{x}')$ between its two inputs; choosing our metric to be great-circle distance allows us to model functions defined on the (spherical) surface of the Earth. For seismic events we will generally incorporate depth as well, so that our inputs $\mathbf{x}$ are (lon, lat, depth) tuples, and distances are computed by

$$r(\mathbf{x}, \mathbf{x}') = \sqrt{\text{greatcircle}((\text{lon, lat}), (\text{lon}', \text{lat}'))^2 + (\text{depth} - \text{depth}')^2}. \tag{3.8}$$

The Matérn covariance function has three *hyperparameters*: σ_f^2 controls the marginal variance of sample functions, ℓ defines a characteristic lengthscale of variation, and p, which we restrict

to be integer-valued, roughly speaking specifies a degree of differentiability.[1] Figure 3.1 shows the effect of the lengthscale ℓ on functions sampled from a Matérn GP. In this work we fix $p = 1$; other hyperparameters are tuned from data by maximizing the (penalized) maximum marginal likelihood, as described below.

Typically when learning a function we do not observe its values directly, and instead have access only to noisy observations $\mathbf{y} = f(\mathbf{X}) + \varepsilon$ for some noise process ε. If the noise is independent and identically distributed (i.i.d.) Gaussian, i.e., $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I})$, then the observations are themselves Gaussian, $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K_y})$, where $\mathbf{K_y} = k(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}$. The i.i.d. noise assumption is common in applications, though in Section 4.9 of this work we will also consider a more complex observation model, in which noisy function evaluations are projected through a linear (wavelet) transformation and then observed with autoregressive noise.

The most common application of GPs is to Bayesian regression (Rasmussen and Williams, 2006), in which function values $\mathbf{f}_* = f(\mathbf{X}_*)$ at test points $\mathbf{X}_*$ are predicted by conditioning on (noisy) training observations (Figure 3.2). In this case the conditional distribution conveniently assumes a closed Gaussian form,

$$p(\mathbf{f}_*|\mathbf{y}; \mathbf{X}, \mathbf{X}_*) = \mathcal{N}(\bar{\mathbf{f}}_*, \mathbf{\Sigma}_*^f),$$

with posterior mean and covariance

$$\bar{\mathbf{f}}_* = k(\mathbf{X}_*, \mathbf{X})\mathbf{K_y}^{-1}\mathbf{y} \tag{3.9}$$

$$\mathbf{\Sigma}_*^f = k(\mathbf{X}_*, \mathbf{X}_*) - k(\mathbf{X}_*, \mathbf{X})\mathbf{K_y}^{-1}k(\mathbf{X}, \mathbf{X}_*). \tag{3.10}$$

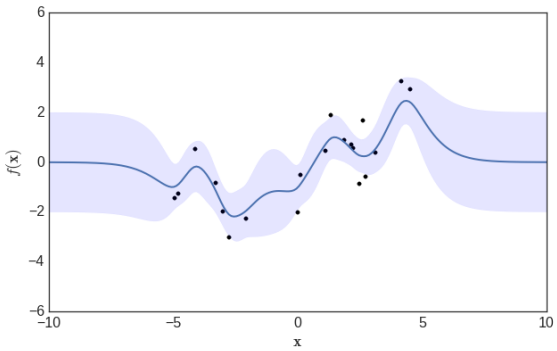
This follows immediately from the standard result (A.5) for conditionals of multivariate Gaussian distributions.

The main practical difficulty in computing Gaussian process posteriors is the matrix inverse $\mathbf{K_y}^{-1}$, which requires time cubic in the number of training points (in practice we do not compute the inverse directly but instead store the Cholesky factorization of $\mathbf{K_y}$, which is more numerically stable but still requires cubic time). For this reason, tasks involving more than a few thousand training points are usually approached via approximate methods, the simplest of which is to simply train multiple "local" GPs on subsets of the training points (Section 6.5).

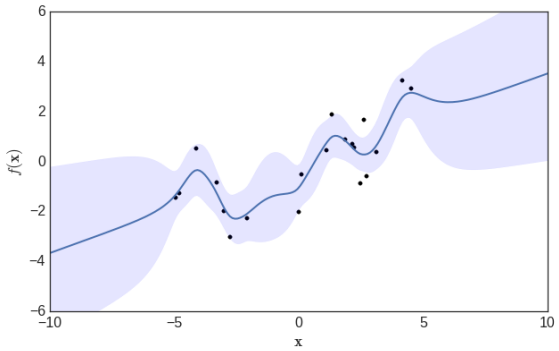
3.4.1 Semiparametric Gaussian processes

Although we typically specify a Gaussian process directly by its covariance function, and make predictions as in (3.9, 3.10) using the (inverse) training set covariance matrix, it is also

[1] Formally, a Matérn-kernel GP is p-times *mean-square differentiable*, where mean-square differentiability is the condition that there exists some stochastic process $\dot{f}$, on the same sample space as f, such that $\lim_{h\to 0} E\left[\left(\frac{f(\mathbf{x}+\mathbf{e}_i h)-f(\mathbf{x})}{h} - \dot{f}(\mathbf{x})\right)^2\right] = 0$ for each basis vector $\mathbf{e}_i$. This condition is implied by, but does not imply, almost-sure differentiability of the sample functions themselves (Adler, 1981).



(a) Standard GP (eqs. (3.9) and (3.10)).



(b) Semiparametric model with linear features $\phi(x) = (1, x), \mathbf{B} = \mathcal{I}$ (eqs. (3.14) and (3.15)).

Figure 3.2: Posteriors from a zero-mean GP conditioned on 20 training points (Matérn kernel, $p = 1$, $\sigma_f^2 = 1$, $\ell = 1$, $\sigma_n^2 = 0.5$, shaded $\pm$ 2 stddevs). Note that the semiparametric GP infers a linear trend extending beyond the range of its training points.

possible to induce a Gaussian process via a parametric representation such as (3.6),

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}),$$

in which we place a (multivariate) Gaussian prior $\mathcal{N}(\mathbf{b}, \mathbf{B})$ on the parameter vector $\mathbf{w}$, and the resulting function values $f(\mathbf{x})$ can be shown to follow a Gaussian process with prior mean

$$\mu_{\mathbf{w}}(\mathbf{x}) = \mathbf{b}^T \mathbf{x}$$

and (degenerate) covariance function

$$k_{\mathbf{w}}(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \mathbf{B} \phi(\mathbf{x}'). \tag{3.11}$$

Following this view, standard Bayesian linear regression is just a special case of Gaussian process regression corresponding to the choice of a linear kernel (3.11). However, in this special case a different kind of computation is available to us: we can maintain a finite-dimensional Gaussian posterior on the parameters, with cost cubic in the number of parameters rather than the number of training points (Gelman et al., 2014). For models with fewer parameters than observations, this is a useful tradeoff.

In this thesis, we will sometimes combine the two approaches, taking both the "function-space" (representation via training data) and "weight-space" (representation via explicit parameters) views within a single model. Suppose we want to model a function g given by the sum of a continuous function f sampled from a GP with kernel k_f, plus an unknown linear function $\mathbf{w}^T \phi(\mathbf{x})$ following a Gaussian prior $\mathbf{w} \sim \mathcal{N}(\mathbf{b}, \mathbf{B})$, so that

$$g(\mathbf{x}) = f(\mathbf{x}) + \mathbf{w}^T \phi(\mathbf{x}).$$

Such a model can be viewed as a form of linear regression in which the residuals are modeled by a GP, rather than the traditional assumption of i.i.d. noise. (Without loss of generality, we can additionally include i.i.d. Gaussian noise in the GP term f, defining an augmented kernel matrix $\mathbf{K_y}$ as above). Since both of its components are Gaussian, the additive process g is itself Gaussian with mean $\mathbf{b}^T\mathbf{x}$ and covariance

$$k_g(\mathbf{x}, \mathbf{x}') = k_f(\mathbf{x}, \mathbf{x}') + \phi(\mathbf{x})^T\mathbf{B}\phi(\mathbf{x}').$$

In principle we can plug these directly into (3.9, 3.10) to predict function values at test points conditioned on training data. In practice the resulting covariance matrices, of the form

$$k_g(\mathbf{X}, \mathbf{X}) = \mathbf{K_y} + \mathbf{\Phi}^T\mathbf{B}\mathbf{\Phi}$$

where $\mathbf{\Phi} = \phi(\mathbf{X})$ is the feature matrix of the training points, may be poorly conditioned, so is it useful to represent the parametric component explicitly. We can do this by applying the *Sherman–Morrison–Woodbury matrix inversion lemma* (Horn and Johnson, 2012),

$$(\mathbf{K_y} + \mathbf{\Phi}^T\mathbf{B}\mathbf{\Phi})^{-1} = \mathbf{K_y}^{-1} - \mathbf{K_y}^{-1}\mathbf{\Phi}^T(\mathbf{B}^{-1} + \mathbf{\Phi}\mathbf{K_y}^{-1}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}\mathbf{K_y}^{-1}, \tag{3.12}$$

which separates out the low-rank parametric component $\mathbf{\Phi}^T\mathbf{B}\mathbf{\Phi}$ from the nonparametric component $\mathbf{K_y}$. Following this approach and applying some additional algebra, given training points $(\mathbf{X}, \mathbf{y})$ it is possible to derive a Gaussian posterior on the parameters $\mathbf{w}$,

$$\begin{aligned} \mathbf{w}|\mathbf{X}, \mathbf{y} &\sim \mathcal{N}(\mathbf{c}, \mathbf{C}) \\ \mathbf{c} &= \mathbf{C}(\mathbf{\Phi}\mathbf{K_y}^{-1}\mathbf{y} + \mathbf{B}^{-1}\mathbf{b}), \\ \mathbf{C} &= (\mathbf{B}^{-1} + \mathbf{\Phi}\mathbf{K_y}^{-1}\mathbf{\Phi}^T)^{-1}, \end{aligned} \tag{3.13}$$

and using this posterior we can apply an alternate form for the Gaussian predictive distribution $\mathbf{g}_* \sim \mathcal{N}(\bar{\mathbf{g}}_*, \mathbf{\Sigma}^g_*)$ on test values $\mathbf{g}_* = g(\mathbf{X}_*)$, given by

$$\bar{\mathbf{g}}_* = \bar{\mathbf{f}}_* + \mathbf{R}^T\mathbf{c} \tag{3.14}$$

$$\mathbf{\Sigma}^g_* = \mathbf{\Sigma}^f_* + \mathbf{R}^T\mathbf{C}\mathbf{R}, \tag{3.15}$$

where $\mathbf{R} = \phi(\mathbf{X}_*) - \mathbf{\Phi}\mathbf{K_y}^{-1}k_f(\mathbf{X}, \mathbf{X}_*)$. These equations express the mean and covariance of the additive function g in terms of the standard GP prediction for f, plus parametric correction terms involving $\mathbf{c}$ and $\mathbf{C}$; derivations are given in section 2.7 of Rasmussen and Williams (2006). Because our model of g includes both a nonparametric term $f(\mathbf{x})$ and a parametric term $\mathbf{w}^T\phi(\mathbf{x})$, we refer to this as a *semiparametric* Gaussian process model. As shown in Figure 3.2, such models can learn structure that generalizes to regions where no training data are available; this is valuable, for example, in modeling *de novo* seismic events.

3.4.2 Hyperparameter selection using the marginal likelihood

The parameters of a GP kernel are known as *hyperparameters*; they determine the prior on functions expressed by the GP (Figure 3.1). The hyperparameters of the Matérn kernel (3.7) include the marginal variance σ_f^2, the lengthscale ℓ, and (under noisy observations) the noise variance σ_n^2; we refer to these jointly using a vector θ. We can select hyperparameters for a training dataset $(\mathbf{X}, \mathbf{y})$ by maximizing the *marginal likelihood* $\mathcal{L}(\theta)$, given by the Gaussian (log) density

$$\begin{aligned}\mathcal{L}(\theta) &= \log p(\mathbf{y}|\mathbf{X}, \theta) \\ &= -\frac{1}{2}\mathbf{y}^T {\mathbf{K_y}}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K_y}| - \frac{n}{2}\log 2\pi,\end{aligned}$$

as a function of the kernel matrix $\mathbf{K_y}$ (which itself depends on $\mathbf{X}$ and θ). For semiparametric models this becomes

$$\mathcal{L}(\theta) = -\frac{1}{2}(\mathbf{y} - \mathbf{\Phi}^T\mathbf{b})^T\left(\mathbf{K_y} + \mathbf{\Phi}^T\mathbf{B}\mathbf{\Phi}\right)^{-1}(\mathbf{y} - \mathbf{\Phi}^T\mathbf{b}) - \frac{1}{2}\log\left|\mathbf{K_y} + \mathbf{\Phi}^T\mathbf{B}\mathbf{\Phi}\right| - \frac{n}{2}\log 2\pi,$$

which, similarly to the predictive equations above, can be made more numerically stable using the matrix inversion lemma (3.12). In both cases, $\mathcal{L}(\theta)$ is differentiable with respect to $\mathbf{K_y}$, which itself is differentiable with respect to θ, so the optimal hyperparameters can be sought by gradient-based optimization, although the problem is generally not convex so multiple initializations may be necessary.

In practice it is often also helpful to guide the optimization towards plausible solutions by imposing a *hyperprior* $p(\theta)$, and maximizing the penalized likelihood

$$\log p(\theta|\mathbf{x}, \mathbf{Y}) = \mathcal{L}(\theta) + \log p(\theta) + C, \tag{3.16}$$

where C is a constant normalization term that may be ignored. This is the approach we generally take in this thesis.

3.5 Linear Gaussian state space models

State space models (Shumway and Stoffer, 2010) are a general class of probability models for time series data. Later in this thesis we will consider several special cases; here we review the basic machinery. A state space model consists of a *latent state* $\mathbf{x}_t$ that evolves stochastically over time, but is available to us only indirectly by way of noisy *observations* $\mathbf{y}_t$. Models are typically formulated so that the *Markov* property holds, meaning that the process evolution depends only on the current state; formally, $p(\mathbf{x}_{t+1}|\mathbf{x}_{0:t}) = p(\mathbf{x}_{t+1}|\mathbf{x}_t)$.

We will focus here on *linear Gaussian* state space models, which are discrete time processes in which both the transition model $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ and observation model $p(\mathbf{y}_t|\mathbf{x}_t)$ are linear functions with Gaussian noise,

$$\begin{aligned}p(\mathbf{x}_{t+1}|\mathbf{x}_t) &\sim \mathcal{N}(\mathbf{F}_t\mathbf{x}_t, \mathbf{Q}_t) \\ p(\mathbf{y}_t|\mathbf{x}_t) &\sim \mathcal{N}(\mathbf{H}_t\mathbf{x}_t, \mathbf{R}_t).\end{aligned}$$

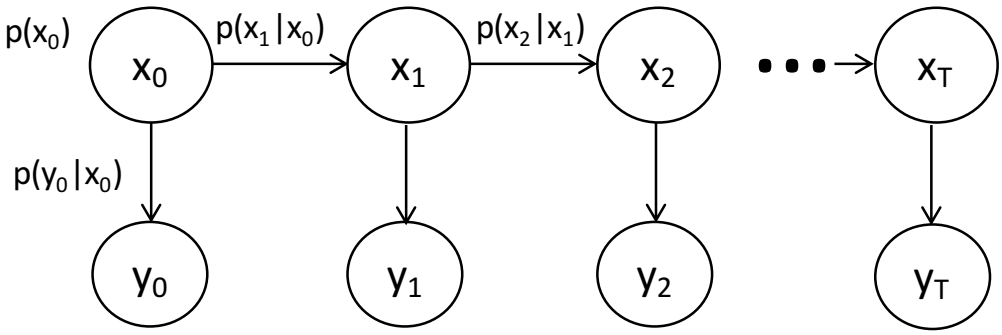


Figure 3.3: State space model structure illustrated as a Bayesian network.

In conjunction with a Gaussian prior distribution $p(\mathbf{x}_0) \sim \mathcal{N}(\mu_0, \mathbf{\Sigma}_0)$ on the initial state, these define a joint probability model over latent states $\mathbf{x} = (\mathbf{x}_t)_{t=0}^T$ and observations $\mathbf{y} = (\mathbf{y}_t)_{t=0}^T$,

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}_0)p(\mathbf{y}_0|\mathbf{x}_0) \prod_{t=1}^{T} p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{y}_t|\mathbf{x}_t).$$

This model structure can be visualized as a Bayesian network, shown in Figure 3.3.

Given some subset of the observations $\mathbf{y}$, we would often like to compute the posterior distribution $p(\mathbf{x}|\mathbf{y})$ on latent states. We may also wish to compute the *marginal likelihood* $p(\mathbf{y}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{x}$, which sums over all latent state sequences to give the overall probability the our model could generate the observations $\mathbf{y}$. For linear Gaussian models, these computations can be performed by an efficient recursive algorithm known as *Kalman filtering*, a special case of message passing on Bayesian networks (Koller and Friedman, 2009; Grewal and Andrews, 2014).

The Kalman filter calculation produces a sequence of state estimates,

$$p(\mathbf{x}_t|\mathbf{y}_0, \ldots, \mathbf{y}_t) \sim \mathcal{N}(\hat{\mathbf{x}}_t, \mathbf{P}_t),$$

consisting of the (Gaussian) posterior on the latent state at each step t given all observations up to that step; these are known as the *filtered* state estimates. They are computed by interleaving *prediction* and *update* steps. In each prediction step, the previous filtered state, represented by Gaussian mean and covariance $(\hat{\mathbf{x}}_{t-1}, \mathbf{P}_{t-1})$, is propagated through the transition model to yield a prediction for the current timestep t,

$$p(\mathbf{x}_t|\mathbf{y}_0, \ldots, \mathbf{y}_{t-1}) \sim \mathcal{N}(\hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}),$$

with predictive mean and covariance derived using (A.2) and (A.3) as

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1} \tag{3.17}$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_{t-1}\mathbf{P}_{t-1}\mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1}. \tag{3.18}$$

In the update step, we revise our prediction to incorporate the new observation $\mathbf{y}_t$. We first propagate the filtered state through the observation model to yield a *joint* distribution on the hidden and observed state,

$$p(\mathbf{x}_t, \mathbf{y}_t|\mathbf{y}_{0:t-1}) = \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}_{t|t-1} \\ \hat{\mathbf{y}}_t \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{t|t-1} & \left(\mathbf{H}_t\mathbf{P}_{t|t-1}\right)^T \\ \mathbf{H}_t\mathbf{P}_{t|t-1} & \mathbf{S}_t \end{bmatrix}\right), \tag{3.19}$$

where we define the observation mean $\hat{\mathbf{y}}_t = \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}$ and covariance $\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{R}_t$, and the derivation follows by applying the linear transformation $[\mathbf{I}, \mathbf{H}]$ to $\mathbf{x}_t$ (A.2), then adding independent Gaussian noise (A.3). Under this joint distribution, we apply eq. (A.5) to derive the conditional distribution $p(\mathbf{x}_t|\mathbf{y}_{0:t})$, i.e. the filtered posterior, with mean $\hat{\mathbf{x}}_t$ and covariance $\mathbf{P}_t$ given by

$$\begin{aligned}\hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{P}_{t|t-1}\mathbf{H}_t^T\mathbf{S}_t^{-1}(\mathbf{y} - \hat{\mathbf{y}}_t) \\ \mathbf{P}_t &= \mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1}\mathbf{H}_t^T\mathbf{S}^{-1}\mathbf{H}_t\mathbf{P}_{t|t-1}.\end{aligned}$$

These are typically computed using an intermediate variable $\mathbf{K}_t$, known as the *Kalman gain* (Grewal and Andrews, 2014),

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y} - \hat{\mathbf{y}}_t) \tag{3.20}$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1} \tag{3.21}$$

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^T\mathbf{S}_t^{-1}. \tag{3.22}$$

As part of the same calculation we can also obtain the marginal likelihood,

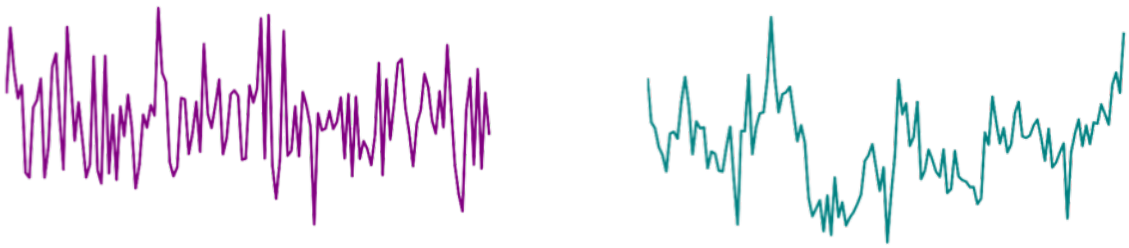
$$p(\mathbf{y}) = \prod_{t=0}^{T} p(\mathbf{y}_t|\mathbf{y}_{0:t-1}) = \prod_{t=0}^{T} \int p(\mathbf{y}_t, \mathbf{x}_t|\mathbf{y}_{0:t-1})d\mathbf{x}_t = \prod_{t=0}^{T} \mathcal{N}(\mathbf{y}_t; \hat{\mathbf{y}}_t, \mathbf{S}_t),$$

as a product of Gaussian densities at each observed step, where the last step uses eq. (A.4) to marginalize the joint distribution (3.19). In practice we perform this computation in log space to avoid underflow, computing the log marginal likelihood as a recursive accumulation,

$$\log p(\mathbf{y}_{0:t}) = \log p(\mathbf{y}_{0:t-1}) + \log \mathcal{N}(\mathbf{y}_t; \hat{\mathbf{y}}_t, \mathbf{S}_t), \tag{3.23}$$

concurrently with the filtering calculation.

Note that the filtering posteriors $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ are different from the full posteriors $p(\mathbf{x}_t|\mathbf{y}_{0:T})$ that we would obtain by conditioning on the entire length of the observed signal. The latter can be obtained by performing an additional backwards pass known as *smoothing* (Grewal and Andrews, 2014). One drawback is that this requires us to store the filtered mean and full covariance matrix of each latent state, so that they can be updated in the backwards pass, which may significantly increase memory requirements. In this thesis we focus on filtering since we will primarily be concerned with computing marginal likelihoods, which are given exactly by the filtering calculation (3.23).



(a) Independent and identically distributed (i.i.d) Gaussian noise.

(b) AR(2) noise with $\phi = (.7, .2)$.

Figure 3.4: Samples from i.i.d. versus autoregressive noise processes.

3.6 Autoregressive processes

An *autoregressive process* of order p is a stochastic process with no hidden state, in which the expected value at time t is a linear function of the values at times $t-p, \ldots, t-1$,

$$z_t = \sum_{k=1}^{p} \phi_k z_{t-k} + \varepsilon_t,$$

with coefficients ϕ, and Gaussian noise at each step,

$$\varepsilon_t \sim \mathcal{N}(0, \sigma_n^2).$$

Because each step can be written as a linear transformation of previous values, plus Gaussian noise, by eq. (A.2) the joint distribution on $\mathbf{z}$ is Gaussian with some covariance matrix $\mathbf{S}_\phi$,

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_\phi),$$

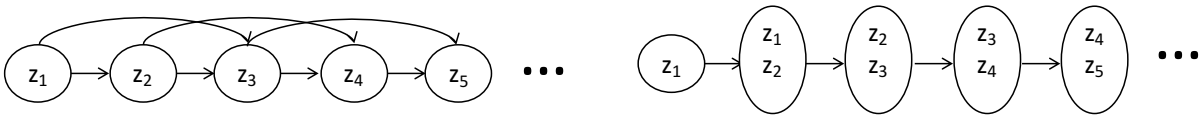
so AR processes are (discrete-time) Gaussian processes. As with other GPs, we can define a nonzero-mean process by shifting the signal by a constant μ, so we restrict to the zero-mean case without loss of generality. Because AR processes are *time-homogenous*, i.e., the same law applies at every timestep, their covariances are fully determined by the *autocorrelation function* ρ,

$$(\mathbf{S}_\phi)_{i,j} = \mathrm{cov}(z_i, z_j) = \rho(i-j)$$

which itself is determined by the process coefficients ϕ by way of a *characteristic polynomial*; see Shumway and Stoffer (2010) for details.

In contrast to an i.i.d. noise process, an AR process generates signals that vary smoothly over time (Figure 3.4). Because there is no hidden state, computing the likelihood of a signal under an AR model is a simple recursive accumulation,

$$\log p(\mathbf{z}) = \sum_{t=1}^{T} \log p(z_t | z_{t-p:t-1}) = \sum_{t=1}^{T} \log \mathcal{N}\left(z_t; \sum_{k=1}^{p} \phi_k z_{t-k}, \sigma_n^2\right). \tag{3.24}$$



(a) Bayesian network for an $AR(2)$ process.

(b) $AR(2)$ process in state-space form.

Figure 3.5: Bayesian network representations of AR processes.

3.6.1 State space model formulation

An AR process can be viewed as an order-p Markov process, in which each state depends on the p previous states. We can always express such a process as a standard, first-order Markov process by incorporating the required memory into the state space (Figure 3.5). This yields a linear Gaussian state space model, in which the latent state contains a memory of process values from recent timesteps,

$$\mathbf{x}_t = (z_t, z_{t-1}, \ldots, z_{t-p+1})^T,$$

and the observation model simply outputs the current value with no noise,

$$y_t = \mathbf{H}_{AR}\mathbf{x}_t = z_t,$$

with $1 \times p$ observation matrix $\mathbf{H}_{AR} = (1, 0, \ldots, 0)$ and trivial noise variance $\mathbf{R}_{AR} = 0$. The transition model predicts the next process value with added noise,

$$\begin{aligned}\mathbf{x}_{t+1} &= \mathbf{F}_\phi \mathbf{x}_t + \varepsilon_{t+1} \\ \varepsilon_{t+1} &\sim \mathcal{N}(0, \mathbf{Q}_{\sigma^2})\end{aligned}$$

with transition matrix and noise covariance

$$\mathbf{F}_\phi = \begin{pmatrix} \phi_1 & \phi_2 & \cdots & \phi_p \\ 1 & 0 & & 0 \\ 0 & 1 & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \qquad \mathbf{Q}_{\sigma^2} = \begin{pmatrix} \sigma_n^2 & 0 & \cdots & 0 \\ 0 & 0 & & 0 \\ 0 & 0 & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Representing an AR process in this way as a state space model, we can use the Kalman recursion (3.23) to compute the likelihood of an observed signal. Compared to the stateless recursion (3.24), using the state space representation allows us to handle missing observations, and (as we shall see in Section 4.9) to compose the AR process with other state space models. However, this flexibility comes at a computational cost: where the stateless recursion runs in $O(Tp)$ time, Kalman filtering requires $O(Tp^2)$ time because it updates a $p \times p$ covariance matrix at each timestep (note that the Kalman likelihood computation is still dramatically cheaper than the $O(T^3)$ required to evaluate a generic Gaussian density). It is possible to avoid the quadratic overhead in simple cases: if there are there are no missing observations or other disruptions, the filtering covariances will converge to a stationary point, after which we no longer need to compute them.

3.7 Wavelets

Real-valued signals are most commonly expressed in the *time domain* basis; that is, in terms of their value at each time step. However, working in a different basis can expose useful structure. For a periodic function $f(t)$, a common choice is the *frequency domain* or Fourier basis,

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{-2\pi i \xi x} d\xi,$$

in which the basis functions $e^{-2\pi i \xi x}$ are sinusoids and their complex-valued coefficients $\hat{f}(\xi)$ specify the contribution (amplitude and phase) of each frequency ξ. A frequency-domain representation can be used, among other things, for signal compression, forming a compact representation of a signal by discarding frequencies not relevant to the task at hand.

Many real-world signals are not fully periodic, and instead have frequency spectra that change over time. *Wavelets* are a class of orthogonal bases that bridge the gap between the time and frequency domains. There is a deep and beautiful theory of wavelet analysis (Mallat, 1999), which this thesis makes very little use of: we exploit wavelets only as a convenient representation for the repeatable structure of seismic signals (Section 4.6), yielding certain computational advantages, but do not claim that they are optimal for such purposes. Indeed, using our model to learn an optimal basis for seismic signals is an interesting avenue of future work. Nonetheless we present in this section a basic introduction to multiresolution wavelet transforms and the construction of wavelet bases, which attempts to provide intuition while avoiding unnecessary formalism.

3.7.1 The Haar transform

We focus specifically on Daubechies wavelets (Daubechies, 1992), and in particular begin with the simplest case, the Haar wavelet transform. The Haar transform of a discrete-time signal $f(t)$ is defined by the following two-step procedure:

1. Compute a set of *difference* or *detail* coefficients $\mathbf{d}$ by convolving f with the step function $\psi(t) = [1, -1]$, at stride 2:

$$d_i = f(2i) - f(2i+1)$$

2. Compute a set of *sum* or *smoothing* coefficients $\mathbf{s}$ by repeating the same operation, but using the averaging function $\phi(t) = \frac{1}{2}[1, 1]$.

$$s_i = \frac{f(2i) + f(2i+1)}{2}$$

For a signal of length n, this procedure yields $\frac{n}{2}$ difference coefficients $\mathbf{d}$ and $\frac{n}{2}$ sum coefficients $\mathbf{s}$. These coefficients represent f in an implicit basis corresponding to shifted copies of the

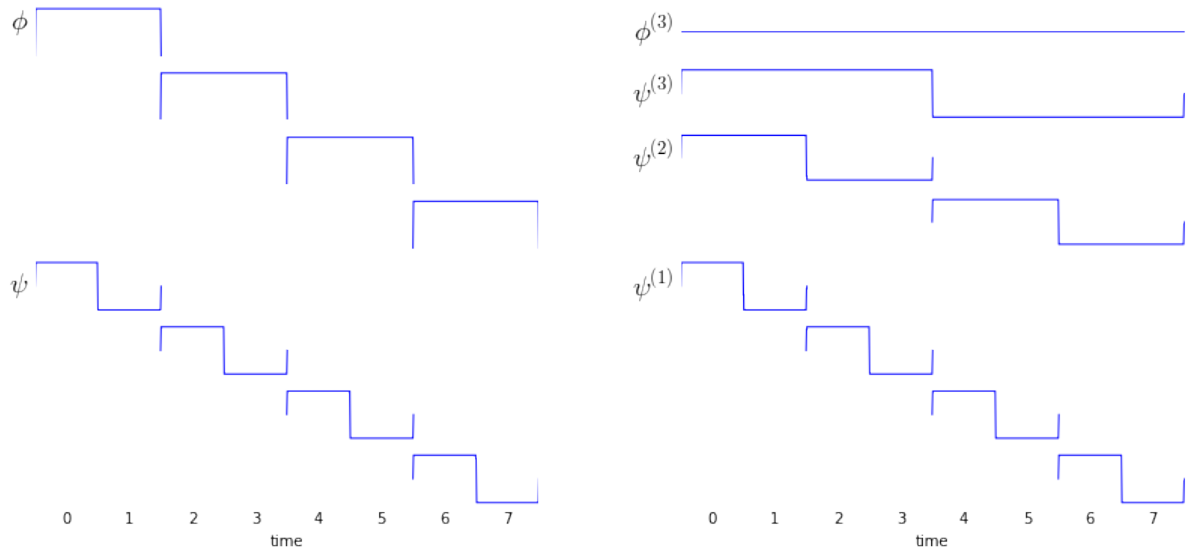


Figure 3.6: Basis functions implied by Haar transforms of a length-8 signal. Left: a single-level transform ($k = 1$) projects onto a basis consisting of shifted copies of the father "sum" wavelet ϕ and mother "difference" wavelet ψ. Right: a recursive transform ($k = 3$) implies a basis consisting of shifted and scaled copies of the difference wavelet ψ, with a single global copy of ϕ.

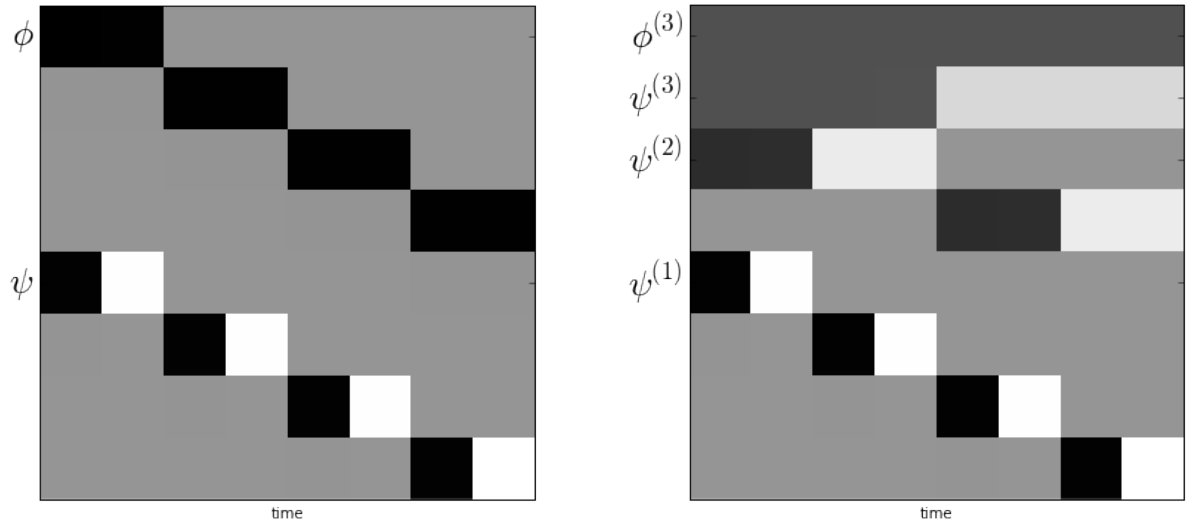


Figure 3.7: Wavelet transform matrices $\mathbf{A}^{(1)}$ for the single-level Haar transform (left) and $\mathbf{A}^{(3)}$ for a recursive transform (right).

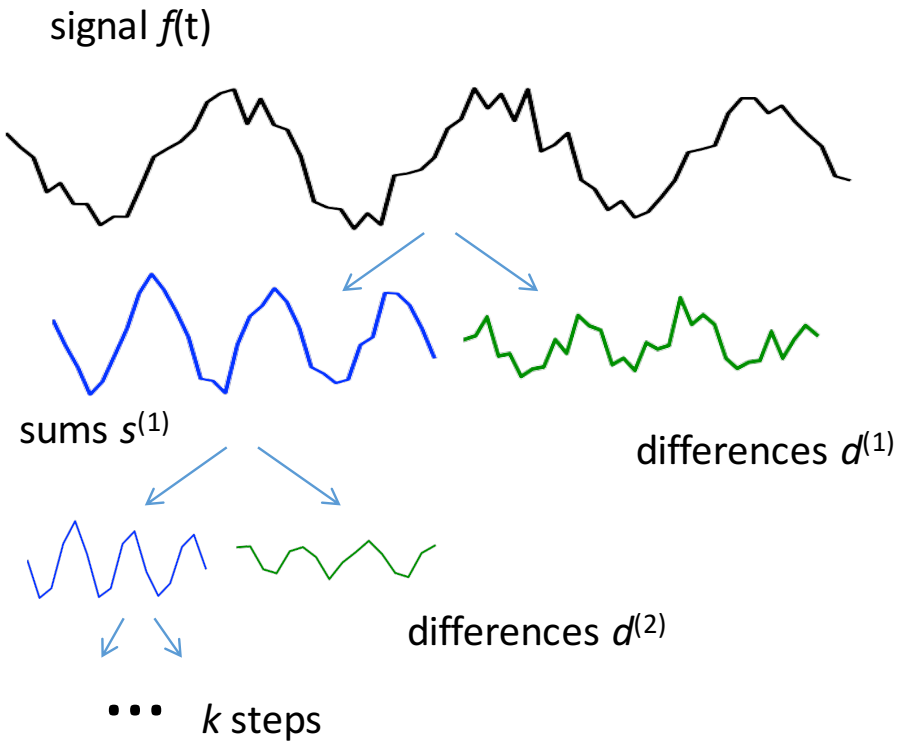


Figure 3.8: Multiresolution Haar wavelet decomposition of a signal $f(t)$ into recursive difference coefficients $\mathbf{d}^{(i)}$.

step function ψ, which we refer to as the *mother* wavelet, and the bump ϕ, the *father* wavelet (Figure 3.6). The difference coefficients perform a sort of edge detection, capturing high-frequency information, while the sum coefficients represent a downsampling of the original signal which preserves low-frequency content.

If desired, we can apply this procedure recursively to generative a *multiresolution analysis* (Figure 3.8). That is, we relabel $\mathbf{d}$ and $\mathbf{s}$ as *first-level* coefficients $\mathbf{d}^{(1)}$ and $\mathbf{s}^{(1)}$, set aside the difference coefficients $\mathbf{d}^{(1)}$, and recursively transform $\mathbf{s}^{(1)}$ by again convolving with the mother and father wavelets (i.e., taking local differences and averages) to yield a new set of $\frac{n}{4}$ difference coefficients $\mathbf{d}^{(2)}$ and $\frac{n}{4}$ sum coefficients $\mathbf{s}^{(2)}$. Repeating this decomposition k times represents the signal by a hierarchical set of difference coefficients $\mathbf{d}^{(1)}, \mathbf{d}^{(2)}, \ldots, \mathbf{d}^{(k)}$ along with the final sum coefficients $\mathbf{s}^{(k)}$, which we concatenate to form the *wavelet coefficients*

$$\mathbf{w}^{(k)} = (\mathbf{d}^{(1)}, \mathbf{d}^{(2)}, \ldots, \mathbf{d}^{(k)}, \mathbf{s}^{(k)}).$$

Since the number of coefficients is halved at each step, we can recurse at most $\log_2 n$ times, but we are also free to perform a partial decomposition by stopping early, i.e., choosing $k < \log_2 n$. The choice of k serves to trade off time- and frequency-domain resolution: large values of k generate basis functions at multiple scales corresponding to different frequency ranges, while small values (e.g., $k = 1$) localize each basis function within the time domain.

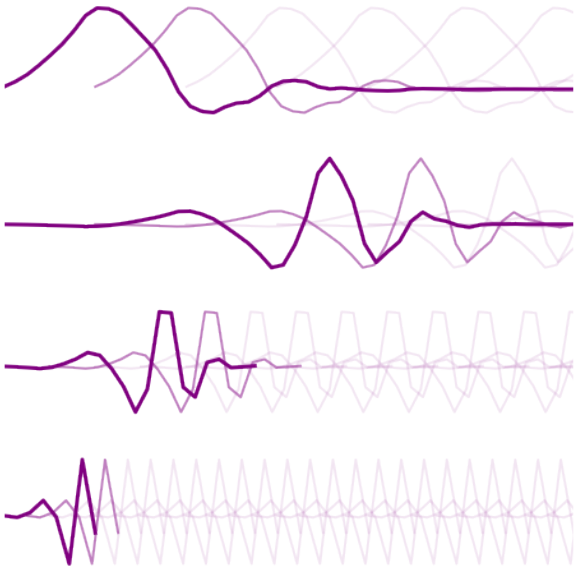


Figure 3.9: Illustration of 4th-order Daubechies wavelet basis (db4), arising from a three-level multiresolution decomposition ($k = 3$), consisting of shifted copies of a father wavelet ϕ_4 (top) and shifted and scaled copies of a mother wavelet ψ_4 (bottom three lines).

It is not hard to see that the coefficients generated by the recursive Haar transform implicitly define a basis consisting of shifted and *scaled* copies of the mother wavelet ψ, where the scaling is determined by ϕ (Figure 3.8). (For this reason the mother and father are sometimes referred to as "wavelet" and "scaling" functions respectively). This basis could in principle be written as an $n \times n$ matrix $\mathbf{A}^{(k)}$ (Figure 3.7), so that the wavelet transform of a signal $\mathbf{f}$ is just the linear projection

$$\mathbf{w}^{(k)} = \mathbf{A}^{(k)}\mathbf{f}.$$

Importantly, although general matrix-vector multiplication would require $O(n^2)$ time, the recursive algorithm just described requires only $O(n/2^k)$ work at iteration k, so it runs in $O(n)$ time.

3.7.2 Daubechies wavelets

Although the Haar transform is simply described as a recursive process of taking differences and averages, it does not always provide the most appropriate representation of real signals. The Haar basis functions (Figure 3.6) are not smooth and thus cannot faithfully represent smooth signals. *Daubechies wavelets* generalize the Haar transform by defining alternative mother and father wavelets ψ and ϕ, which implicitly define a set of multiresolution basis functions (Figure 3.9) following an analogous (and similarly efficient) procedure to the Haar decomposition detailed above.

The Daubechies wavelets of order r are defined as the length-$2r$ vectors ψ_r, ϕ_r that maximize a set of moment conditions (Daubechies, 1992). The special case $r = 1$ corresponds to the length-2 Haar wavelets $\psi_1 = [1, -1]$ and $\phi_1 = [1, 1]$. Although we will not derive the general case here, intuitively, an order-r Daubechies basis is able to compactly represent functions that are locally polynomial of degree $r - 1$. That is, a Haar basis represents functions that are locally constant (so their difference coefficients are uniformly zero), an order-2 basis represents functions that are locally linear, and so on. The seismic models in this thesis use an order-4 Daubechies basis, though this has not been rigorously tuned.

Chapter 4

Generative Signal Model

This chapter describes the SIGVISA probability model, i.e., the framework by which SIGVISA assigns probabilities to possible worlds. A "possible world" in this context consists of a set of seismic events, the signals they generate across a network of stations, and some additional latent variables describing the process by which the signals are generated, for example the arrival times of each seismic phase at each station.

Conceptually, SIGVISA defines a joint probability distribution on the Earth's entire seismic history, including all events and signals past, present, and future. The exposition in this section will generally take this perspective. In practice, many aspects of the model are tuned from historical data, which corresponds to *conditioning* on past observations:

$$p(\text{world}_{\text{future}}|\text{world}_{\text{past}}) \propto p(\text{world}_{\text{future}}, \text{world}_{\text{past}}).$$

That is, we imagine sampling many worlds from the model, and throw out all of the samples that are not consistent with historical observations. What remains are worlds in which the model's tunable parameters — which describe the relationship between events and the signals they generate — correspond well to reality. Upon observing new signals, we further throw out all worlds that do not match those signals, and the set of remaining worlds (all of which are consistent with our past and current observations) defines a *posterior* distribution on the unobserved events. If the model is good, this posterior distribution will be heavily concentrated around real events.

In reality, there are infinitely many possible worlds, and the rejection sampling approach just described is not practical to implement. Instead we will describe model-specific procedures for inferring event bulletins from observed signals (Chapter 5) and for training the model from past data (Chapter 6).

4.1 Overview

The goal of a generative model is to capture enough of the relevant causal processes to enable reliable inference of the quantities we care about. For seismic monitoring, we would

Physical phenomenon	Classical technique	SIGVISA model
Predictable travel times (1D)	Traditional pick-based monitoring	IASPEI 91 travel time model
Spatial continuity of waveforms	Waveform matching / cross-correlation methods for sub-threshold detections	Gaussian process (kriging) model of wavelet coefficients describing signal modulation
Spatial continuity of travel-time residuals	Double-differencing	Gaussian process model of travel-time residuals
Other predictable regularities (attenuation, coda decay rates, spectral content, etc.)	(Not exploited by existing techniques)	GP models of envelope shape parameters

Table 4.1: Correspondence between traditional monitoring systems and the SIGVISA forward model.

like to infer events, and the better the model captures the physical processes by which events generate signals, the better our inferences will be. In particular, our model will attempt to capture the following phenomena:

- **Predictable travel times**: given the location and origin time of an event, we can predict the times at which each phase will arrive at a given station. Given detections from multiple stations, this modeling assumption can be inverted to associate and locate events via multilateration, as in traditional detection-based monitoring systems (Section 2.5.2).
- **Repeatable waveforms**: the waveforms produced at a given station by events with similar origin locations will tend to correlate with each other. Inverting this assumption enables sub-threshold event detections and location from a single station, in regions where historical waveform data are available (Section 2.6).
- **Repeatable travel times**: the deviations, or *residuals*, from a simple (e.g., 1D) travel-time model will tend to be correlated for events with similar origin locations. This is because unmodeled variation in the local slowness field affects all events similarly. Inverting this assumption (and exploiting accurate relative arrival times obtained by waveform cross-correlation) yields joint relocation methods such as double-differencing (Section 2.7).

Each of these physical processes can be inverted to yield an existing technique for event detection and relocation; Table 4.1 summarizes the relationships. By modeling them all jointly as part of a unified forward model, inverted via Bayesian inference, SIGVISA attempts to recover the advantages of each of these individual techniques.

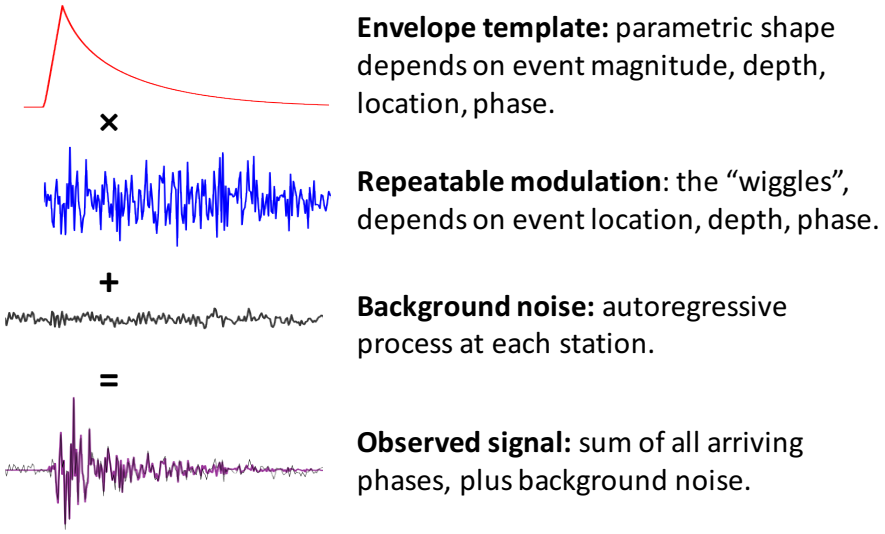


Figure 4.1: Composing sampled model components into a generated signal.

4.2 Generative story

This section outlines the structure of the SIGVISA probability model, in the form of a procedure to sample a possible world. This generative story fully specifies the probability model, given definitions of its component parts, which are explained in subsequent sections.

Event history: We first sample a set of events $\mathbf{E} = (\mathbf{e}_i)_{i=1}^{N}$ from an event prior $p(\mathbf{E})$. Note that the number of events N is itself a random quantity. Each event $\mathbf{e}_i$ is represented as a vector $\mathbf{e}_i$ with five components giving its latitude, longitude, depth in kilometers, origin time in seconds[1], and magnitude. (Future work could extend the event representation to include additional properties of the source mechanism.)

Signals and intermediate variables. Given the event history as a shared global variable, we independently generate signals $s_j(t)$ for each station $j \in (1, \ldots, M)$.[2] The signals are functions of time t, which we discretize at a fixed sampling rate (typically 10Hz). The event–signal dependence is mediated by a a number of intermediate variables, which are also station-specific. For clarity we describe the signal generation process for a generic station, suppressing the station index j. Figure 4.1 illustrates the special case of sampling a signal for a single event with a single phase.

[1]We follow the Unix convention of representing time by seconds elapsed since 00:00 UTC, January 1, 1970.

[2]This independence assumption is computationally advantageous, and reasonable when stations are far apart. It would need to be relaxed, e.g., to model multiple elements within an array station.

1. For each event $\mathbf{e}_i$ and each potential phase k, we sample a phase activation indicator $h_i^{(k)}$ from a distribution $p(h_i^{(k)}|\mathbf{e}_i)$ conditioned on the event depth and event–station distance. The phase activation indicators determine the set of phases observed at the current station from the origin $\mathbf{e}_i$ (Sections 2.2 and 4.4).

2. For each active phase k, we sample the waveform produced by the arrival of that phase. To model correlations from path-dependent effects, all quantities involved are sampled *jointly* for all events from Gaussian process priors. Specifically, we jointly sample:

   a) a set of *arrival times* $\tau = (\tau_i)_{i=1}^N$, from a distribution $p(\tau|\mathbf{E})$ conditioned on the event origin times, locations, and event–station distance.

   b) a set of *envelope shape parameters* $\theta = (\theta_i)_{i=1}^N$, also from a joint distribution $p(\theta|\mathbf{E})$ conditioned on the event locations, magnitudes, and event–station distance. These parameters determine an envelope shape template $g(t;\theta_i)$ (Figure 4.5). Details of the parameterization θ and envelope function g are presented below in Section 4.5.

   c) a set of *wavelet coefficients* $\mathbf{W} = (\mathbf{w}_i)_{i=1}^N$, which are passed through a discrete wavelet transform $\mathbf{A}$ to define a set of *modulation signals*

   $$\mathbf{M} = \mathbf{A}\mathbf{W},$$

   or equivalently for each event i,

   $$\mathbf{m}_i = \mathbf{A}\mathbf{w}_i.$$

   These modulations represent the "wiggles" in the observed signal that are not captured by the simple envelope shape g, but nonetheless represent important structure that we assume to be repeatable for events in nearby locations. Our model encodes this assumption by sampling the coefficients $\mathbf{W}$ jointly for all events from a Gaussian process $p(\mathbf{W}|\mathbf{E})$ conditioned on the event locations.

3. We additionally sample a set of *unassociated arrivals* at each station. These represent unmodeled phase arrivals as well as signals from small events that cannot be localized. As with events, the count R of unassociated arrivals is itself a random quantity. Each unassociated arrival, indexed by r, is represented similarly to an event phase: by an arrival time τ_r, shape parameters θ_r, and modulation signal $\mathbf{m}_r$, with these quantities sampled from a prior specific to unassociated arrivals (Section 4.7).

4. The *predicted signal* $\bar{s}_j(t)$ is generated at each station by summing the contributions from each phase (i,k) and unassociated arrival r, with modulation signals scaled mul-

tiplicatively by corresponding envelope shapes (Figure 4.1).

$$\begin{aligned}\bar{s}_j(t) = &\sum_{i=1}^{N} \sum_{k \in \mathbf{h}_i} m_{i,j}^{(k)}(t - \tau_{i,j}^{(k)}) \cdot g(t - \tau_{i,j}^{(k)}; \theta_{i,j}^{(k)}) \\ &+ \sum_{r=1}^{R} m_{r,j}(t - \tau_{r,j}) \cdot g(t - \tau_{r,j}^{UA}; \theta_{r,j}).\end{aligned} \tag{4.1}$$

Note that the envelope shape g and modulation signal m are shifted by the arrival time τ; formally we define both g and m to equal zero when their argument is negative. In principle each arrival continues contributing to the signal forever, though in practice we cap the length of each arrival at 300 seconds, so that a small number of phases are active at any particular time.

5. We finally sample autoregressive noise process parameters $\psi_j = (\mu_j, \sigma_j^2, \phi_j)$ (Section 3.6) from a station-specific prior $p(\psi_j)$, and then sample a noise realization $\mathbf{z}_j$ from this process. Note that we typically work with signals in short blocks of one or two hours, so modeling the noise coefficients as randomly sampled from a prior allows the model to adapt during inference to the actual noise characteristics observed in each period. Fitting the AR process priors is discussed in Section 6.4.

   Given the noise process, we generate the *observed signal* $s_j(t)$ at each station as the sum of signal and noise,

$$s_j(t) = \bar{s}_j(t) + z_j(t). \tag{4.2}$$

The outcome of this sampling process is a possible world, consisting of an event history $\mathbf{E}$, signals $(\mathbf{s}_j)_{j=1}^M$ at each station, and latent variables $(\mathbf{h}, \tau, \theta, \mathbf{W}, R, \psi, \mathbf{z})$ providing an interpretable description of the mechanism by which each signal was generated from the events. Table 4.2 summarizes the variables included in the model.

The sections that follow flesh out the details of this story, including the event prior, envelope shape parameterization, and priors on the repeatable modulation signals and unassociated arrivals.

4.3 Event prior

The prior distributions on the number of events, event time, depth, location, and magnitude are tuned from historical data via standard estimation techniques ("empirical Bayes"). Most of the prior formulation is shared with NETVISA (Arora et al., 2013); one advantage of the Bayesian formulation is that improvements to the priors of one system can be easily

Quantity	Notation	Indexed By
Event count	N	n/a
Event source	$\mathbf{e}$	event i
Active phases	h	station j, event i, phase k
Arrival time	τ	station j, arrival (i,k) or r
Envelope (collectively)	θ	station j, arrival (i,k) or r
Envelope rise time	ρ	station j, arrival (i,k) or r
Envelope amplitude	α	station j, arrival (i,k) or r
Envelope decay (initial)	γ	station j, arrival (i,k) or r
Envelope decay (coda)	β	station j, arrival (i,k) or r
Wavelet coefficients	w	station j, event i, phase k, coef c
Modulation signal	$m(t)$	station j, arrival (i,k) or r
Unassociated count	R	station j
AR noise parameters	ψ	station j
Background noise process	$z(t)$	station j
Observed signal	$s(t)$	station j

Table 4.2: Random variables in the SIGVISA generative model.

incorporated into the other. Events are sampled independently under the prior,

$$p(\mathbf{E}) = p(N) \prod_{i=1}^{N} p(\mathbf{e_i}) \cdot N! \tag{4.3}$$

$$p(\mathbf{e}_i) = p(\mathbf{e}_i^{\text{loc}}) p(\mathbf{e}_i^{\text{depth}}) p(\mathbf{e}_i^{\text{time}}) p(\mathbf{e}_i^{\text{mb}}), \tag{4.4}$$

where the component priors on location, depth, etc. are as discussed below. Note that events are subject to a labeling symmetry, since permuting the event indices does not change the model; we correct this by summing the joint density over all $N!$ permutations.

Event occurrence is modeled as a time-homogenous Poisson process,[3] corresponding to an assumption that there is some constant probability $\lambda\epsilon$ of generating an event in each infinitesimal time period of duration ϵ. This induces a Poisson distribution on the event count,

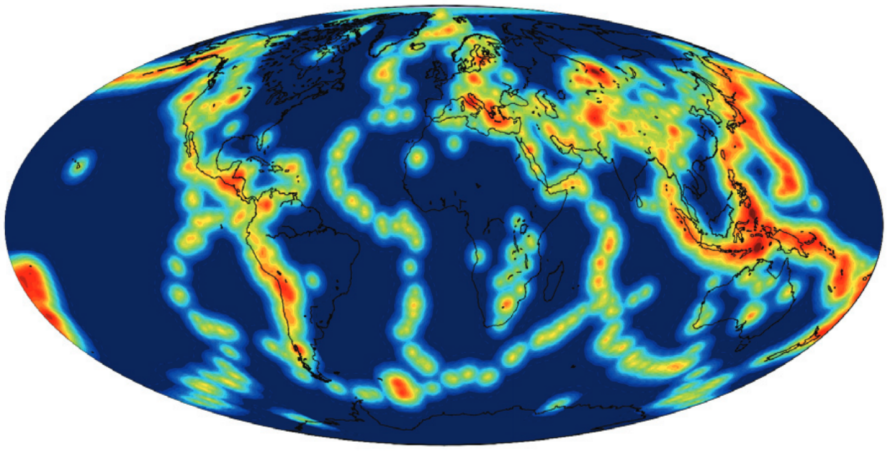
$$p(N) = \text{Poisson}(\lambda T),$$

and a uniform distribution on event times,

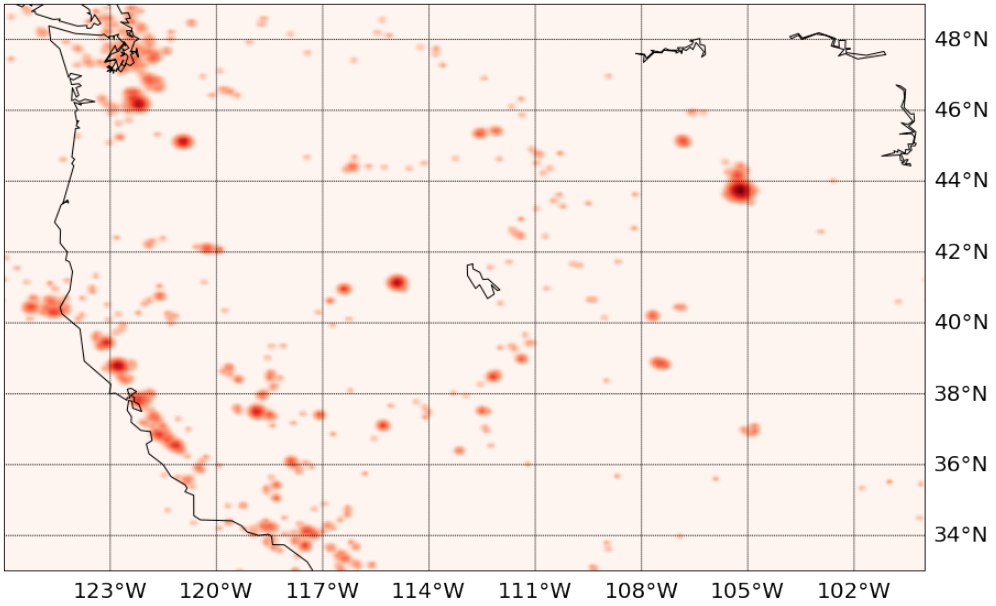
$$p(\mathbf{e}_i^{\text{time}}) = 1/T,$$

where T is the length of the period being modeled. The rate parameter λ is estimated from historical data via maximum likelihood. For the global IMS network and LEB training data, this yields approximately 4.5 events per hour; for the Western US dataset evaluated in Chapter 7, covering a much small area, the rate is 0.1 events per hour.

[3] A more sophisticated model could use a heterogenous process to account for, e.g., the increased probability of aftershocks following a large event.



(a) Global prior from Arora et al. (2013) (tuned bandwidth 0.7°).



(b) Regional prior on western US events (tuned bandwidth 0.05°).

Figure 4.2: Event location density estimates learned from historical data.

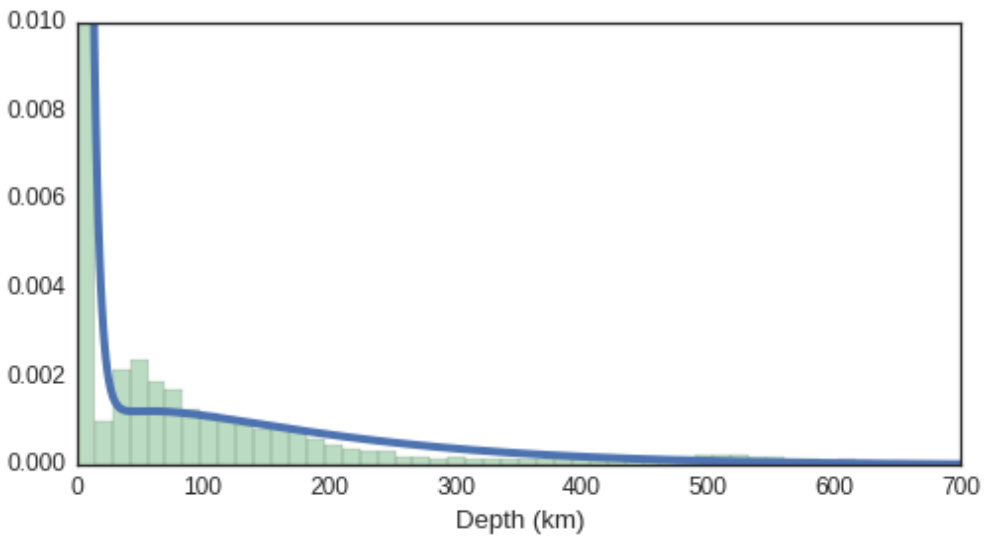


Figure 4.3: Event depth prior, with histogram of LEB depths.

The event location prior is represented by a kernel density estimate of historical events, with Gaussian kernel bandwidth set by leave-one-out cross validation, mixed with a uniform distribution to account for *de novo* events. In our experiments the weight of the uniform component was fixed at 0.01. Figure 4.2 visualizes a global prior distribution learned from LEB data as well as the prior learned for our Western US dataset.

For event depths, we use a mixture of exponential and gamma densities, with the exponential modeling the concentration of near–surface events while the gamma fits the long tail of deeper events. Specifically, we use

$$p(\mathbf{e}_i^{\text{depth}} = d) \sim .7 \cdot \text{Exp}(d; \lambda = 0.2) + .3 \cdot \text{Gamma}(d - 6.27; \alpha = 1.42, \beta = .0079),$$

with d in kilometers, and we impose a hard maximum depth of 700km. This produces the density shown in Figure 4.3. This is a significant improvement on the uniform prior used by Arora et al. (2013), although the fit is still rough; a more sophisticated model would also represent a joint distribution over surface location and depth (Arora et al., 2015).

Magnitudes are modeled by the Gutenberg–Richter law (Gutenberg and Richter, 1954), which posits that the number of events with magnitude m or more is 10 times the number of events with magnitude $m+1$ or more. This is represented by an exponential distribution, Figure 4.4. In principle the law implies an arbitrarily large number of very low magnitude events, many of which are undetectable. In practice we impose a minimum magnitude of 2.0 (and use only events above this threshold to fit other prior components such as event rate and location density).

Note that, unlike traditional body- and surface-wave magnitudes, which are defined in terms of observations, magnitude in SIGVISA is a latent variable that stochastically determines the amplitudes of body and surface waves at all stations. In this sense SIGVISA

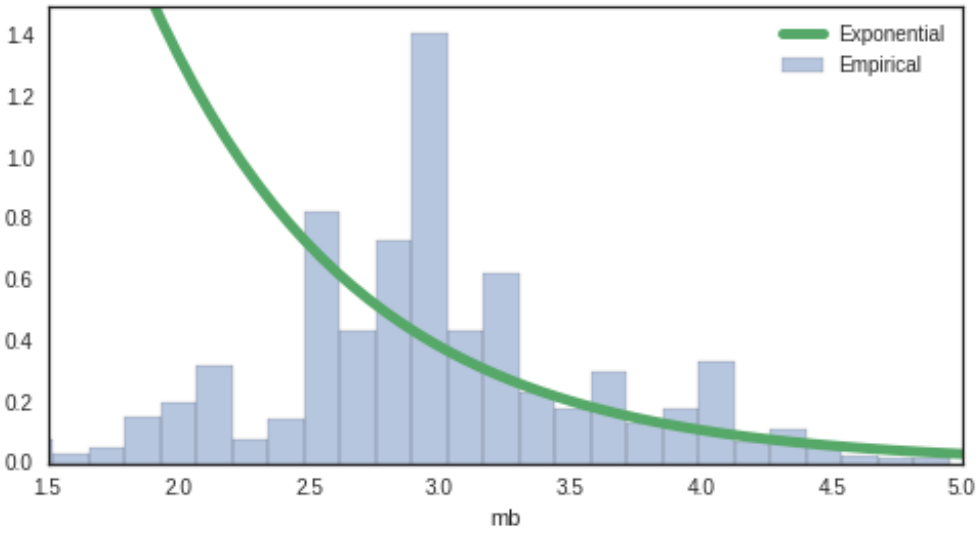


Figure 4.4: Event magnitude prior, with histogram of empirical magnitudes from the ISC training bulletin (Section 7.1). The gap between the model and empirical data is largely due to the inability to detect very low-magnitude events.

Phase	$r_{\min}$	$r_{\max}$	$d_{\min}$	$d_{\max}$
P	0 (deg)	98	40 (km)	700
P	17	98	0	40
Pn	2	20	0	40
Pg	0	20	0	40
Sn	2	18	0	40
Lg	0	18	0	40

Table 4.3: Distance and depth ranges for phases in the SIGVISA model. A phase is considered active if any of its definitions are satisfied.

magnitudes are somewhat analogous to moment magnitudes (Section 2.1), though we do not (yet) include source parameters such as rupture area, displacement, shear modulus, and seismic efficiency explicitly in our model.

4.4 Arriving phases

For the experiments in this thesis, we consider a small set of regional phases with fixed distance/depth ranges, given in Table 4.3 and taken originally from the GA system definition (Le Bras et al., 2002). A phase is considered legal for a particular event and station if any one of its (potentially multiple) distance–depth conditions is satisfied. However, for inference purposes it is convenient to model phase activations as random, so that a phase may

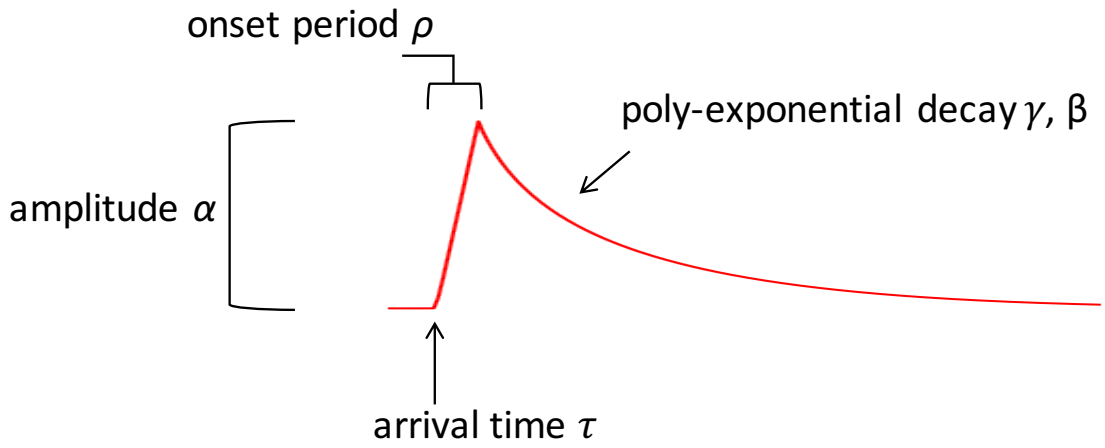


Figure 4.5: Parameterized envelope shape for an arriving phase.

sometimes fail to appear even if we are within its legal range (this allows an MCMC sampler to traverse the boundary regions without requiring a phase birth/death move whenever it crosses a hard boundary).

We therefore model the set of phases arriving at a particular station as a nearly-deterministic function of the event–station distance r and event depth d. Given a source location $\mathbf{e}_i$, the set of phases $\mathbf{h}_{i,j}$ arriving at station j is generated by sampling independently an indicator for each phase,

$$p(\mathbf{h}_{i,j}|\mathbf{e}_i) = \prod_{k \in \text{phases}} p(h_{i,j}^{(k)}|\mathbf{e}_i),$$

where the phase indicators $h_{i,j}^{(k)}$ are Boolean random variables. The activation probability for each phase depends on the signed distance ξ to the boundary of its legal region, defined such that positive values correspond to interior (legal) points. Concretely, we define

$$p(h_{i,j}^{(k)} = 1|\mathbf{e}_i) = \begin{cases} 0 & \text{if } \xi \leq 0 \\ 1/(1 + e^{-a(\xi - b/2)}) & \text{if } 0 < \xi < b \\ 1 & \text{otherwise} \end{cases} \tag{4.5}$$

such that illegal phases are given zero probability, and within the boundaries the activation probability increases following a logistic sigmoid curve up to some distance $2b$, after which it is 1. We arbitrarily set $b = 100$km for surface distance and $b = 10$km for depth, and define a so that the boundary probability at $\xi = 0$ is 0.01.

4.5 Parametric envelope shapes

The shape of each arriving phase is governed by a simple function $g(t; \theta)$, with parameters $\theta = (\tau, \rho, \alpha, \gamma, \beta)$ (note that here we are including as an envelope parameter the arrival time

Parameter	**Features** $\phi(\mathbf{e})$	$p(\sigma_n^2)$	$p(\sigma_f^2)$	$p(\ell)$
Arrival time (τ)	n/a	LN(0, 0.5)	LN(2.2, 0.5)	LN(4,1)
Amplitude (log α)	$\left(1, \delta, \sin(\frac{\delta}{15000}), \cos(\frac{\delta}{15000})\right)$	LN(-1.75, 0.5)	LN(-.75, 0.5)	LN(3,1)
Onset (log ρ)	(1, mb)	LN(-3, 1)	LN(-2, 1)	LN(3,1)
Peak decay (log γ)	(1, mb, δ)	LN(-4, 1)	LN(-3, 1)	LN(3,1)
Coda decay (log β)	(1, mb, δ)	LN(-3.5, 1)	LN(-2.5, 1)	LN(3,1)
Wavelet coefs (w)	n/a	Beta(5, 2)	$\mathbb{I}[1 - \sigma_n^2]$	LN(3,1)

Table 4.4: Feature parameterizations and hyperpriors for Gaussian process models. δ is event–station distance in km. LN(μ, σ) is a lognormal distribution with location μ and scale σ, having mean $e^{\mu+\sigma^2/2}$.

τ, which was separate above). We use the functional form

$$g(t;\theta) = \begin{cases} \alpha(t-\tau)/\rho & \text{if } t-\tau < \rho \\ \alpha(t-\tau+1)^{-\gamma} e^{-\beta(t-\tau)} & \text{otherwise,} \end{cases} \tag{4.6}$$

which corresponds to an initial linear onset of duration $\rho > 0$, peaking at an amplitude $\alpha > 0$, then decaying according to a polynomial rate $\gamma > 0$ and exponential rate $\beta > 0$ (Figure 4.5). The polynomial decay term allows for a sharp fall from the initial peak, which then leads into a stable coda modeled by the exponential term. The decay formulation is inspired by models of seismic coda (Mayeda et al., 2003), while the linear onset follows the form used by Cua (2005). We also considered combinations of an exponential onset, purely exponential decay, and a peak "plateau", as in Cua (2005), but settled on this parameterization as the best overall fit.

The goal of this envelope representation is to encode some simple prior knowledge about the signals generated by seismic phase arrivals — they have arrival times and amplitudes, they peak quickly, and then decay — even in the absence of historical waveform data from which we might be able to predict more detailed structure. For example, we expect a magnitude 6.0 event in a novel location to produce large arrivals at nearby stations, though we may not know exactly what those arrivals will look like.

We enforce the positivity constraints on ρ, α, γ, and β by representing those parameters in the log domain. Each parameter is sampled jointly across a set of events, and independently of the other parameters,

$$p(\theta|\mathbf{E}) = p(\tau|\mathbf{E})p(\log\rho|\mathbf{E})p(\log\alpha|\mathbf{E})p(\log\gamma|\mathbf{E})p(\log\beta|\mathbf{E}),$$

though sampling all shape parameters jointly would be a useful direction of future work.

The individual shape parameters are modeled by Gaussian processes (Section 3.4) conditioned on the event location and depth. That is, for each station j and phase k, the parameter is assumed to follow an unknown function of the event origin $f_j^{(k)}(\mathbf{e})$, and we model these functions as draws from a GP with noisy observations. For example, arrival

times are modeled as

$$\tau_i = f_\tau(\mathbf{e}_i) + \varepsilon, \tag{4.7}$$

$$f_\tau \sim GP(\mu_{\tau,j}^{(k)}, k_{\tau,j}^{(k)}) \tag{4.8}$$

and similarly for the other parameters, where ε represents Gaussian noise for each observation and is incorporated into the GP covariance following Section 3.4.

For the arrival time τ and amplitude α, we use mean functions $\mu_\tau(\mathbf{e})$ given by a travel time model, and $\mu_\alpha(\mathbf{e})$ given by a source amplitude model, so that those GPs model a travel-time residual and a (log-)amplitude transfer function respectively. Other parameters are modeled with $\mu = 0$. We currently use the IASPEI-91 travel time model (Kennett and Engdahl, 1991) and a Brune source model (Brune, 1970); our system also implements the Mueller–Murphy explosion model (Mueller and Murphy, 1971), though this was not used for the experiments in this thesis. A more sophisticated travel time model such as LLNL-G3D (Simmons et al., 2012) would likely improve performance, although the deficiencies of the 1D model are mitigated somewhat by our use of GPs to model location-dependent residuals.

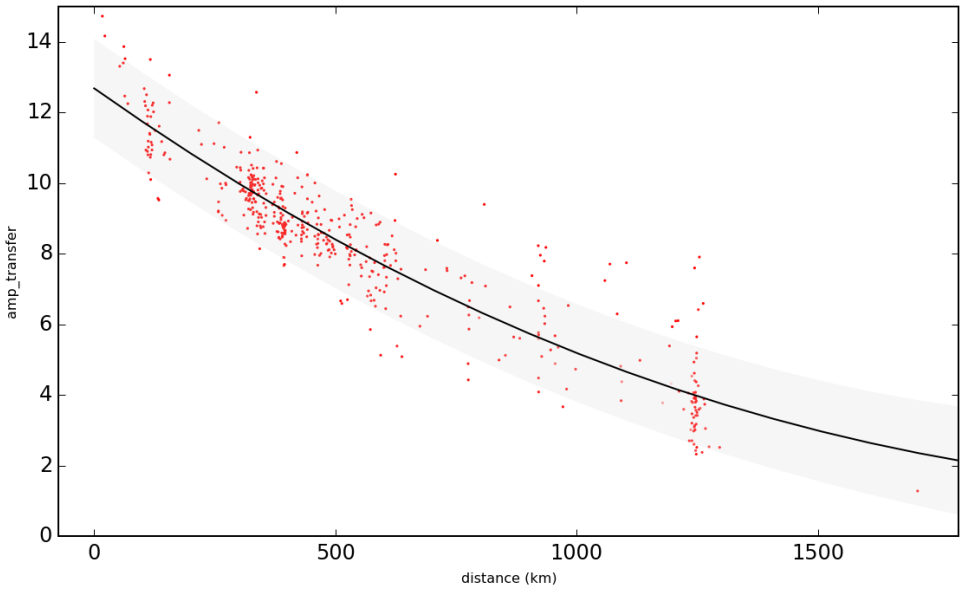
The GP covariance k is taken to be a Matérn kernel with hyperparameters, consisting of a noise variance σ_n^2, marginal variance σ_f^2, and characteristic lengthscale ℓ, sampled from log-normal hyperpriors, listed in Table 4.4. The training procedure (Chapter 6) selects hyperparameters for each station, phase, and shape parameter by maximizing the marginal likelihood of historical data, penalized by the relevant hyperprior.

In addition to the nonparametric Matérn component, we use the machinery of semiparametric GPs (Section 3.4.1) to incorporate an additional term that is linear in a set of event features. This is intended to represent predictable regularities, such as amplitude decay with distance, that allow the model to generalize to *de novo* events. Table 4.4 gives the features used for each shape parameter, as functions of event magnitude mb as well as the event–station distance δ (in km). Figure 4.6 shows an example of a learned relationship between event–station distance and the (log-) amplitude transfer function; note that the use of sinusoidal features allows our model to learn a nonlinear relationship.

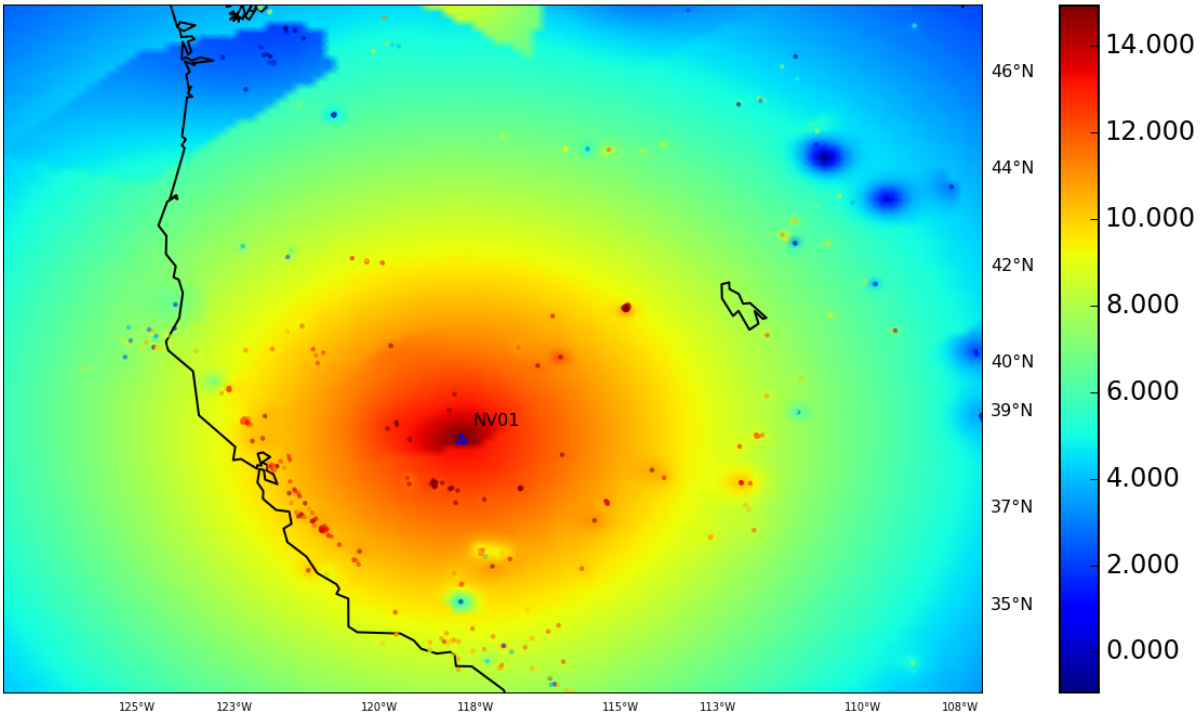
The use of GP models imposes (log) Gaussianity on the envelope parameters; in practice, we cut off the Gaussian tails to eliminate physically unrealistic hypotheses. For example, we impose a maximum travel time residual of 25 seconds, so that the joint density of arrival times τ (for a particular station and phase, indices omitted) is given by

$$p(\tau|\mathbf{E}) = \begin{cases} \mathcal{N}(\bar{g}(\mathbf{E}), \Sigma_g(\mathbf{E})) & \text{if } |\tau_i - \mu_\tau(\mathbf{e}_i)| \le 25, \forall i \\ -\infty & \text{otherwise} \end{cases}, \tag{4.9}$$

where $\bar{g}$ and Σ_g are given by the semiparametric GP posterior evaluated at test points $\mathbf{E}$, following eqs. (3.14) and (3.15). Other parameters are evaluated analogously, with tails truncated at four standard deviations from the GP mean prediction.

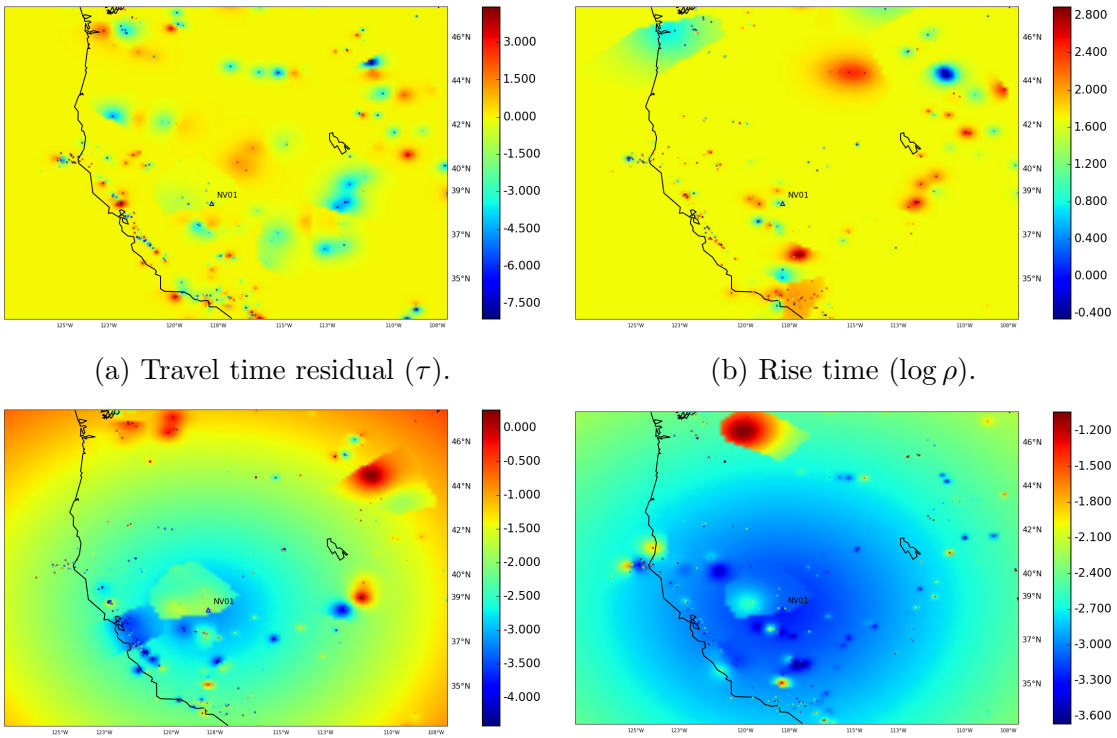


(a) Parametric component showing nonlinear dependence on event–station distance (shaded $\pm 2\sigma$).



(b) Full GP model including local adjustments to the parametric distance dependence.

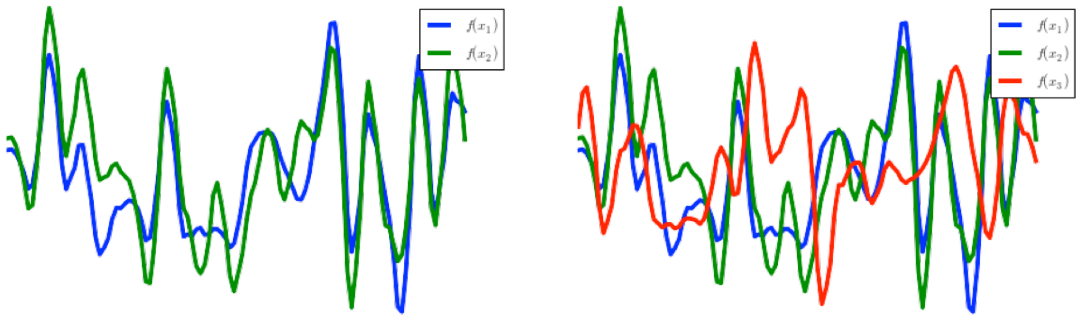
Figure 4.6: Learned model of Lg phase amplitude transfer (envelope log-amplitude minus source log-amplitude) for IMS station NV01.



(a) Travel time residual (τ). (b) Rise time ($\log \rho$).

(c) Peak decay rate ($\log \gamma$). (d) Coda decay rate ($\log \beta$).

Figure 4.7: Learned models of remaining parameters for Lg phases at IMS station NV01. Discontinuities correspond to Voronoi cell boundaries for local GPs (Section 6.5).



(a) From doublet source locations $\mathbf{x}_1, \mathbf{x}_2$. (b) Adding a distant location $\mathbf{x}_3$.

Figure 4.8: Modulation signals sampled from a GP prior on Daubechies (db4) wavelets. Signals for nearby events $\mathbf{x}_1$, $\mathbf{x}_2$ are highly correlated.

4.6 Wavelet coefficients and modulation signals

The envelope shape template $g(t;\theta)$ modeled in the previous section determines only a broad outline of the arriving signal. The vast majority of information regarding the waveform's fluctuations is contained in the modulation signal, $m(t)$, which we multiply by the envelope shape to generate the final waveform.

As specified above, we represent each modulation signal using a vector of wavelet coefficients $\mathbf{w}$. Specifically, for each phase arrival we use a multiresolution ($k = 3$) Daubechies db4 wavelet basis (Figure 3.9) to model 20 seconds of repeatable signal at 10Hz, so that

$$\mathbf{w} = \left(\mathbf{d}^{(1)}, \mathbf{d}^{(2)}, \mathbf{d}^{(3)}, \mathbf{s}^{(3)}\right)$$

consists of 220 coefficients $(w_c)_{c=1}^{220}$, partitioned into four blocks: 31 sum coefficients $\mathbf{s}^{(3)}$, 31 third-level difference coefficients $\mathbf{d}^{(3)}$, 55 second-level difference coefficients $\mathbf{d}^{(2)}$, and 103 first-level difference coefficients $\mathbf{d}^{(1)}$.

We sample wavelet coefficients jointly for all events from a Matérn GP prior, so that nearby events will receive similar coefficients and generate similar modulation signals (Figure 4.8). Each of the four blocks of coefficients is given its own hyperparameters (sampled independently for each station and phase), allowing the model to choose, e.g., a different characteristic lengthscale for the high-frequency versus lower-frequency components, though all coefficients within a block use the same hyperparameters. As with the envelope shape GPs, the wavelet GP hyperparameters are modeled as arising from a hyperprior (Table 4.4), which is used during training to tune the hyperparameters from historical data for each station and phase (and coefficient block). Since the scale of the modulation signal is unidentifiable due to multiplication by the envelope amplitude α, we fix the prior on the modulation signal to have unit marginal variance by imposing the hyperparameter constraint $\sigma_f^2 = 1 - \sigma_n^2$.

For each station j and phase k, we define the coefficient matrix $\mathbf{W}_j^{(k)}$ so that the ith column gives the coefficients for event i; rows $\mathbf{w}_c$ correspond to coefficients c. Given the hyperparameters, the coefficients are sampled from independent GPs,

$$\begin{aligned} p(\mathbf{W}|\mathbf{E}) &= \prod_c p(\mathbf{w}_c|\mathbf{E}) \\ &= \prod_c \mathcal{N}\left(\mathbf{w}_c; \bar{f}_c(\mathbf{E}), \Sigma_{f,c}(\mathbf{E})\right), \end{aligned} \tag{4.10}$$

so that $\mathbf{W}$ is correlated across events (columns), with values for each coefficient sampled according to the GP posterior at $\mathbf{E}$, following eqs. (3.9) and (3.10).

For computational reasons we do not attempt to represent a repeatable modulation signal beyond the first 20 seconds of each phase arrival. Instead we model the ongoing modulation as (nonrepeatable) white noise, so that the full modulation signal is defined piecewise

$$m(t) = \begin{cases} (\mathbf{A}\mathbf{w})(t) & \text{if } 0 \le t < 20s \\ \varepsilon(t) & \text{otherwise} \end{cases}$$

where $\mathbf{A}\mathbf{w}$ is the transformed wavelet signal and $\varepsilon(t) \sim \mathcal{N}(0,1)$ is a white noise process.

4.7 Unassociated arrivals

In addition to arrivals associated with specific events, we also find it useful to model bursts of signal energy not generated by any particular event as *unassociated* arrivals. Of course, anything that generates seismic waves is an event in some sense; however, many events are sufficiently small and localized that they register only at a single station and cannot be localized. For such events it is useful to have a lightweight representation that avoids the need to instantiate parameters at every station in the network.

Unassociated arrivals also play a useful role during inference, by providing a level of explanation intermediate between background noise and a fully localized event. They are useful as a source of event birth proposals, allowing us to propose new events according that coherently they explain some currently unassociated arrivals (Section 5.4.2), and in event death proposals, allowing us to discard incoherent event hypotheses without being forced to immediately propose a better alternative. In this role as intermediate explanations they are analogous to detections in a traditional monitoring pipeline (Section 2.5.1), although they are generated dynamically throughout the inference process rather than through fixed bottom-up processing.

Our choice of prior for unassociated arrivals governs the role they play in modeling and inference. At each station, the occurrence of unassociated arrivals is modeled by a Poisson process,

$$\begin{aligned} p(R_j) &= \text{Poisson}(R_j; \lambda_j T) \\ p(\tau_{j,r}) &= 1/T, \end{aligned}$$

similarly to the prior on events. A high prior rate λ_j causes the model to generate many unassociated arrivals, and indeed to prefer unassociated explanations to genuine events, while a low rate causes the model to birth potentially spurious events that it is then unable to localize. We manually set the prior rate at all stations to expect one unassociated arrival every 1000 seconds.

Unlike the envelopes for event arrivals, unassociated arrivals have no origin location for us to condition on, so the GP models of Section 4.5 are not applicable. Instead we sample the envelope shape parameters for each arrival r from a manually chosen, station-independent prior, given by

$$\begin{aligned} \log \rho_{j,r} &\sim \mathcal{N}(0.3, 1) \\ \log \alpha_{j,r} &\sim \mathcal{N}(3, 1) \\ \log \gamma_{j,r} &\sim \mathcal{N}(-2.5, 1) \\ \log \beta_{j,r} &\sim \mathcal{N}(-2.5, 1.5) \end{aligned}$$

with the main consideration being that the unassociated prior prefers smaller amplitudes α (relative to the scale of our signals), so that larger arrivals will prefer to be associated with localizable events.

The wavelet-GP models of modulation signals are similarly not applicable to unassociated arrivals, so we simply model the modulation as a white noise process,

$$m_{j,r}(t) \sim \mathcal{N}(0, 1).$$

As with events, unassociated arrivals are subject to a labeling symmetry, so at each station the joint density of unassociated arrivals

$$p(R_j, \theta_j^{(UA)}, \mathbf{M}_j^{(UA)}) = p(R_j) \cdot R_j! \cdot \prod_{r=1}^{R_j} p(\tau_{j,r})p(\rho_{j,r})p(\alpha_{j,r})p(\gamma_{j,r})p(\beta_{j,r})p(\mathbf{m}_{j,r}), \tag{4.11}$$

includes a correction factor $R_j!$ from summing over all permutations.

4.8 Joint density

We have described our model in terms of a sampling process; to score hypotheses during inference we will also need to compute the model's probability density. This is given by the product of the event prior and the signal forward model at each station,

$$p(\mathbf{E}, \mathbf{h}, \theta, \mathbf{W}, \mathbf{M}, \mathbf{R}, \psi, \mathbf{S}) = p(\mathbf{E}) \prod_j p(\mathbf{s}_j, \mathbf{h}_j, R_j, \theta_j, \mathbf{W}_j, \mathbf{M}_j, \psi_j | \mathbf{E}), \tag{4.12}$$

where the per-station forward model density is obtained by multiplying the conditional densities arising from each step of the sampling process described in this chapter,

$$\begin{aligned} p(\mathbf{s}_j, \mathbf{h}_j, R_j, \theta_j, \mathbf{W}_j, \mathbf{M}_j, \psi_j | \mathbf{E}) = &\left(\prod_i p(\mathbf{h}_{i,j} | \mathbf{e}_i) \right) \left(\prod_k p(\theta_j^{(k)} | \mathbf{E}) p(\mathbf{W}_j^{(k)} | \mathbf{E}) p(\mathbf{M}_j^{(k)} | \mathbf{W}_j^{(k)}) \right) \\ &p(R_j, \theta_j^{(UA)}, \mathbf{M}_j^{(UA)}) p(\psi_j) p(\mathbf{z}_j = \mathbf{s}_j - \bar{\mathbf{s}}_j | \mathbf{E}, \theta_j, \mathbf{M}_j, \psi_j). \end{aligned} \tag{4.13}$$

Note that evaluating these densities requires not just the observed signals and a hypothesized event history, but also hypotheses regarding latent quantities including the set of arriving phases, locations and shapes of their envelope templates, parameters describing the noise process, and the decomposition of the observed signal into noise and (scaled) modulation signals. The MCMC inference described in Chapter 5 is a method for searching over such hypotheses, but it turns out we can avoid the need to explicitly represent a signal decomposition by *marginalizing out* the modulation signals, i.e., computing the density

$$p(\mathbf{s}_j | \mathbf{E}, \mathbf{h}_j, R_j, \theta_j, \psi_j) = \int p(\mathbf{z}_j = \mathbf{s}_j - \bar{\mathbf{s}}_j | \mathbf{E}, \theta_j, \psi_j, \mathbf{M}_j) \prod_{k \in \mathbf{h}_j} p\left(\mathbf{M}_j^{(k)} | \mathbf{E}\right) p\left(\mathbf{M}_j^{(UA)}\right) d\mathbf{M} \tag{4.14}$$

that sums over all possible modulation signals $\mathbf{M}$. The next section describes an algorithm for efficiently computing this marginalized density.

4.9 Efficient marginalization of linear Gaussian signal models

We efficiently compute the *marginal* likelihood of seismic signals under our model by formulating the marginal density (4.14) as the likelihood of a linear Gaussian state space model (Section 3.5), which is efficiently evaluated using the Kalman filter recursion (3.23).

Recall from Section 3.6.1 that autoregressive noise can be formulated as a linear Gaussian state space model. In this section we additionally describe a state space representation of the wavelet modulation signals $\mathbf{m}$, and then show that the seismic signal model (4.14) can be expressed as the composition of models describing the noise and modulation processes. In the process we develop a Bayesian fast wavelet transform, i.e., an algorithm for efficiently computing the posterior distribution on the wavelet coefficients of a noisily observed signal; to our knowledge this problem has not been considered in previous literature.

4.9.1 Bayesian wavelet transforms

We first consider a simple model of random signals, in which we draw a set of m wavelet coefficients from a Gaussian prior

$$\mathbf{w} \sim \mathcal{N}(\mu_{\mathbf{w}}, \Sigma_{\mathbf{w}}),$$

and then construct the random signal $\mathbf{m}$ by applying an inverse wavelet transform $\mathbf{A}^T$,

$$\mathbf{m} = \mathbf{A}^T\mathbf{w}.$$

Since the wavelet transform is a linear operator, the resulting signal is also Gaussian,

$$\mathbf{m} \sim \mathcal{N}(\mathbf{A}^T\mu_{\mathbf{w}}, \mathbf{A}^T\Sigma_{\mathbf{w}}\mathbf{A}), \tag{4.15}$$

following eq. (A.2).

We refer to this model as a *Bayesian wavelet transform*: while the standard wavelet transform is only defined with respect to fully-observed signals, adding a Gaussian prior allows us to compute posterior distributions (using eqn. A.5) over wavelet coefficients when some of the signal observations are missing, or when the signal is observed with added noise, as it will be in our case. We can also compute the marginal likelihood of $\mathbf{m}$ by evaluating the density (4.15). Bayesian wavelet transforms have been considered in previous literature, e.g., Abramovich et al. (1998), Ruggeri and Vidakovic (2005), though they are typically seen as a source of "shrinkage" rules for obtaining sparse representations, rather than as building blocks in larger probability models, and computational complexity is a major concern. We will see below how the computations for our model can be done in an efficient way.

It is not strictly necessary that $\mathbf{A}$ represents a wavelet transform; the Gaussian distribution would hold for any linear transformation of the coefficients $\mathbf{w}$. However, the sparse

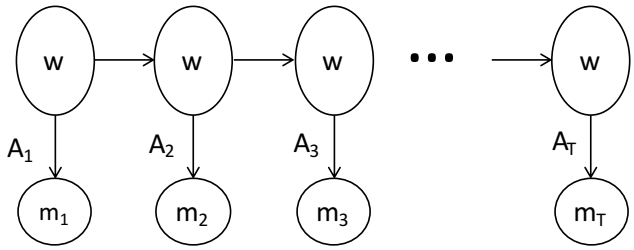


Figure 4.9: Wavelet transform represented as a trivial state space model, in which the latent state is a static representation of the full coefficient vector.

structure of wavelet transform matrices will turn out to be computationally convenient. Figure 3.7 shows a visualization of a wavelet transform matrix $\mathbf{A}$; note that most of the entries are zero, and that the nonzero entries in each row are *compactly supported*, i.e., only nonzero during a finite interval; this is the property we will take advantage of. See Section 3.7 for more on families of wavelet transforms.

4.9.2 State space representation

A naïve evaluation of a Bayesian wavelet transform requires cubic time ($O(m^3)$), which becomes prohibitive for longer signals. This is in contrast to the $O(m)$ time achieved by the deterministic transform that exploits the sparse structure of the wavelet basis (Section 3.7). It turns out that we can achieve a similar speedup by formulating the Bayesian wavelet transform as a state space model.

We first describe a naïve version of the model to demonstrate the principle. Let the latent state $\mathbf{x}$ be the unknown wavelet coefficients $\mathbf{w}$, with prior distribution $\mathcal{N}(\mu_\mathbf{w}, \mathbf{\Sigma}_\mathbf{w})$. The transition model is simply the identity $\mathbf{F} = \mathbf{I}$, with zero transition noise ($\mathbf{Q} = 0$) so that our model has the same hidden state $\mathbf{w}$ at every timestep. The observation model $\mathbf{H}_t$ at timestep t is the tth row of the inverse transform matrix $\mathbf{A}^T$, with no observation noise ($\mathbf{r}_t = 0$). It should be easy to see that this model, shown in Figure 4.9, yields the appropriate Gaussian distribution on outputs.

We do not seem to have gained much by switching to this representation, since we now have to update an m-dimensional Gaussian distribution at every timestep. However, further optimization is possible by exploiting the compact support property of wavelet bases. Let α_t denote the indices of the "active" basis vectors at time t, that is,

$$\alpha_t = \{k | \mathbf{A}_{k,t} \neq 0\}.$$

Then let $\mathbf{w}^{(t)} = \mathbf{w}_{\alpha_t}$ denote the sub-vector containing only those active coefficients. We construct a state space model in which the hidden state at time t is $\mathbf{w}^{(t)}$, using the natural specialization of the observation matrix $\mathbf{H}_{\alpha_t}$ to include only the entries corresponding to active coefficients, i.e., the nonzero entries. To maintain the hidden state over time, we use the transition model to drop coefficients that are no longer needed from the state vector,

and to sample new coefficients as required from their prior distributions. This is simple conceptually but messy in notation.

Let $\alpha_t(i)$ denote the ith active component at time t (defined in sorted order), and conversely let $\texttt{idx}(\alpha_t, k)$ denote the index in α_t of coefficient k, so that $\texttt{idx}(\alpha_t, \alpha_t(i)) = i$. For example, if $\alpha_1 = (0, 3, 7)$, meaning that the signal at time 1 is a linear combination of basis vectors 0, 3, and 7, then $\alpha_1(2) = 3$, and $\texttt{idx}(\alpha_1, 3) = 2$. If coefficient k is not active at time t, then $\texttt{idx}(\alpha_t, k) = \texttt{None}$. We construct $(\mathbf{F}_t)_i$, the ith row of the transition matrix at time t, so that the transition model will copy the required coefficient from the previous timestep if available, and otherwise sample it from its prior. Let $\mathbf{1}(i) = (0, \ldots, 0, 1, 0, \ldots, 0)$ denote a vector of zeros with a 1 in the ith position, and define $\mathbf{1}(\texttt{None})$ as simply a vector of zeros. Then

$$
\begin{aligned}
(\mathbf{F}_t)_i &= \mathbf{1}(\texttt{idx}(\alpha_{t-1}, \alpha_t(i))), \\
Q_{i,i} &= \begin{pmatrix} (\Sigma_{\mathbf{w}})_{\alpha_t(i),\alpha_t(i)} & \text{if } \texttt{idx}(\alpha_{t-1}, \alpha_t(i)) = \texttt{None} \\ 0 & \text{otherwise} \end{pmatrix}.
\end{aligned}
$$

We assume that the prior covariance $\Sigma_{\mathbf{w}}$ is diagonal, so that we can sample each coefficient from its prior independently of the others. Non-diagonal covariances can also be handled in this framework, somewhat more messily, if the conditional distribution for each coefficient depends only on other coefficients present in the active set when that coefficient is sampled. Full covariance matrices that introduce arbitrary long-range dependencies between coefficients cannot be efficiently handled in this setting. However, the restriction to diagonal covariances may seem reasonable if we view the wavelet representation as an attempt to "whiten" the dependence structure of the generated time-domain signal, so that the wavelet coefficients are statistically independent.[4]

Representing only the active components of the wavelet basis leads to a significant computational advantage. Instead of a hidden state of size m, containing all wavelet coefficients, the active sets is of size $O(\log m)$, which is significantly smaller. Applying the Kalman filter recursions (Section 3.5) to this model, we compute the signal likelihood, along with a filtering posterior on wavelet coefficients, in time $O(n \log^2 m)$.

4.9.3 Seismic model

The Bayesian wavelet transform described above is a thinly veiled gloss of our seismic signal model, Section 4.6, in which the (diagonal) Gaussian priors on wavelet coefficients are provided by Gaussian processes conditioned on historical signals, eq. (4.10).

[4]This is only a loose analogy, since wavelet bases are not explicitly constructed with this goal in mind. Learning a basis using methods such as probabilistic PCA, which assumes an independent prior on the latent coefficients, or ICA, which explicitly optimizes a measure of independence on the latent representation, would lend more weight to this argument; however, those methods do not in general yield bases with the compact support structure that we rely upon for computational efficiency. Learning a basis under such support constraints is an interesting subject for future work.

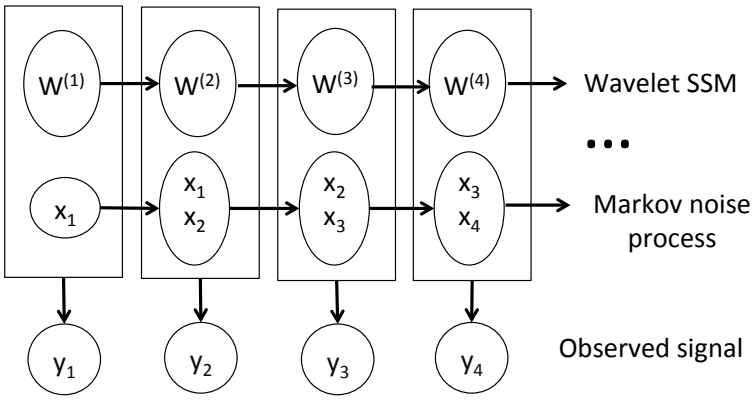


Figure 4.10: Bayesian network structure of a state space model tracking the sum of an AR(1) noise process $\mathbf{x}$ and a signal generated by a Gaussian prior on wavelet coefficients, with coefficients $\mathbf{w}^{(t)}$ active at time t.

In the the full SIGVISA model, modulation signals are scaled by an envelope shape template $g(t;\theta)$. We write this template as a vector $\mathbf{g}_\theta$, and scaling by the envelope template corresponds to multiplication by the diagonal matrix $\mathbf{G}_\theta = \text{diag}(\mathbf{g}_\theta)$. That is, the predicted signal for a given arrival is just an additional linear transformation,

$$\bar{\mathbf{s}} = \mathbf{G}_\theta \mathbf{m} = \mathbf{G}_\theta \mathbf{A}^T \mathbf{w},$$

so that $\bar{\mathbf{s}}$ is also Gaussian distributed, conditioned on θ. We incorporate this into a state space representation by extending the observation model $\mathbf{H}$, i.e., taking $\mathbf{H}_i$ to be the ith column of the *scaled* transform matrix $\mathbf{G}_\theta \mathbf{A}^T$.

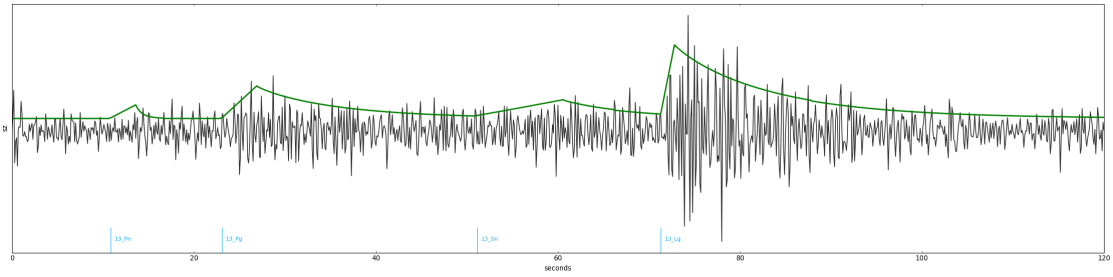
We also trivially form state space representations for envelopes with i.i.d. modulation signals, covering the cases of unassociated arrivals as well as the nonrepeatable portions of arriving phases following the initial repeatable signal. These processes do not require any state at all, but are represented by i.i.d. observation noise, with variance $\mathbf{r}_t = (\mathbf{G}_\theta)^2_{t,t}$.

4.9.4 Composing state space models

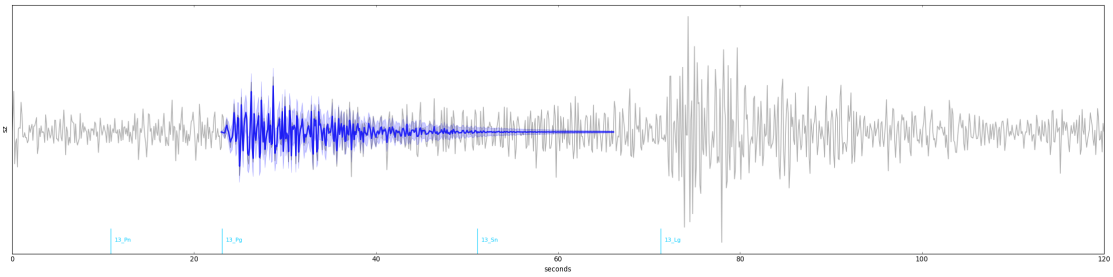
The SIGVISA signal model (eqs. (4.1) and (4.2)) is a linear combination of signals from all arriving phases, plus an autoregressive noise model. We represent this process as a unified state space model, in which the latent state jointly tracks the wavelet coefficients of all currently active modulation signals, along with the noise process, and the observation model takes the sum of these components, with the modulation signals scaled by the appropriate envelope templates as described above.

In general, it is simple to compose state space models by stacking their state vectors,

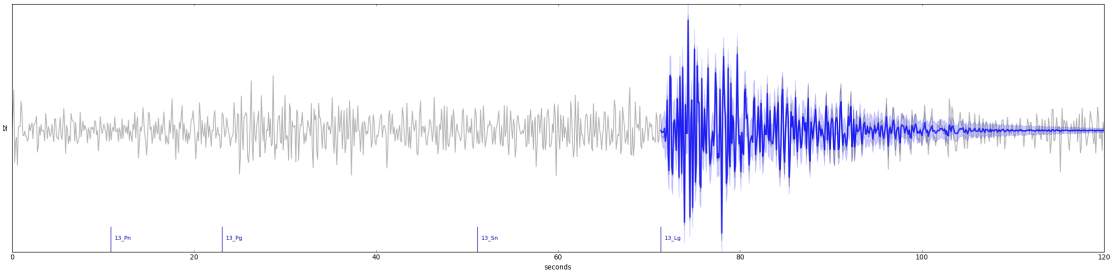
$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix},$$



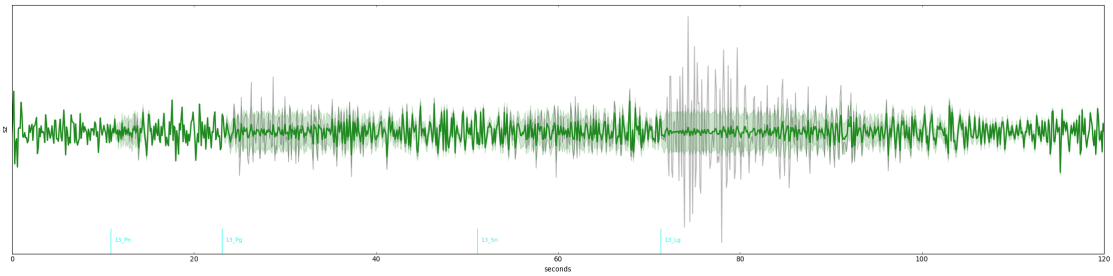
(a) Observed signal (black) with phase arrival times (bottom) and envelope shape $\mathbf{g}_\theta$ (green) inferred by MCMC.



(b) Posterior mean signal for Pg arrival $\mathbf{G}_\theta \mathbf{m}^{(Pg)}$, in blue, shading $\pm$ two standard deviations.



(c) Posterior mean signal for Lg arrival $\mathbf{G}_\theta \mathbf{m}^{(Lg)}$, in blue, shading $\pm$ two standard deviations.



(d) Posterior mean of autoregressive background noise, in green, shading $\pm$ two standard deviations. Note the noise process continues during phase arrivals, with higher posterior variance due to signal/noise uncertainty.

Figure 4.11: Filtering posterior on components of a signal observed at NVAR.

allowing the transition model to act separately on the two components,

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_1 & 0 \\ 0 & \mathbf{F}_2 \end{pmatrix}, \qquad \mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 & 0 \\ 0 & \mathbf{Q}_2 \end{pmatrix},$$

and merging the observation models to yield the sum of the individual observations

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_1 & \mathbf{H}_2 \end{pmatrix}, \qquad \mathbf{R} = \mathbf{R}_1 + \mathbf{R}_2.$$

This procedure generalizes straightforwardly to combinations of three or more state space models.

Using state space representations of the AR noise process $\mathbf{z}$ (Section 3.6.1), the scaled wavelet modulation signals $\mathbf{G}_\theta\mathbf{M}$ (Section 4.9.3), and (trivially) the i.i.d. modulation signals, we represent the SIGVISA signal model conditioned on events and envelope shapes, $p(\mathbf{s}_j|\mathbf{E}, \theta_j, \psi_j)$, as a state space model summing the individual processes. Figure 4.10 illustrates a simple example of a model combining a wavelet modulation signal with autoregressive noise. The filtering posterior tracks the uncertainty in the hidden state for each component process (the diagonal blocks of the covariance matrices), as well as the the dependence between the processes (represented in the off-diagonal blocks). By propagating individual component posteriors through the observation model, we can visualize the model's decomposition of a observed signal into arriving phases and noise (Figure 4.11).

Note that the combined model will have a state space of varying size. When no arrivals are active, the hidden state represents only the AR noise process, so it is of size p. When a single arrival is active, the hidden state is of size $p + \log m$, including the wavelet coefficients describing that arrival. When J arrivals overlap, the hidden state contains all of their wavelet coefficients and is of size $p + J \log m$. Since inference time scales with the square of the hidden state size, the vast majority of the computation may be spent on a relatively small fraction of the timesteps: those where multiple overlapping arrivals create a complex inference problem.

4.9.5 Discussion

Formulating the marginal signal density (4.14) as a state space model allows us to compute it relatively efficiently. It does, however, introduce a subtle approximation: because we prune inactive coefficients from the state space, the priors on wavelet coefficients for different events are assumed to be independent. In reality, this is false because coefficients are coupled by the GP prior (4.10). This independence assumption is perhaps not a big deal at test time when using models conditioned on historical data, since we expect those data to already provide a good picture of the signals in a region: if we are predicting the signal at a location with 100 nearby training events, increasing the effective training size to 101 by *also* conditioning on a nearby test event is unlikely to make a big difference.[5]

[5]The main exception would be an aftershock sequence in a novel location, for which we have no previous training data. We bypass this issue in our evaluation (Chapter 7) by including the first six hours of the Wells aftershock sequence in our training set. Enhancing our model to perform online inference, learning as it goes, is an interesting subject for future work.

Matters are different at *training* time, when we do not yet have any historical data to condition on, and our inference goal is to find a high-likelihood *alignment* of signals from nearby events so that we can extract waveforms that correlate according to the GP prior. In this setting the independence assumption becomes a fatal flaw.

Computing a density that takes the dependence between multiple events into account turns out to be nontrivial. We could, of course, construct an explicit multivariate Gaussian covariance matrix, by evaluating the joint GP prior on wavelet coefficients, propagating this covariance through a wavelet transform and envelope scaling, then adding in the explicit covariance matrix of the autoregressive background noise. But evaluating this density would require time cubic in the length of the signal, which is prohibitive in practice. Instead, Section 6.2 describes an approach that maintains the linear-time efficiency of state space models but accounts for dependencies approximately through a form of graphical model message passing.

Chapter 5

Inference

Given a probability model on a set of random variables, and observed values for some of those variables, *inference* is the task of computing a representation of the conditional distribution on the unobserved variables. In Bayesian terms we refer to this as the posterior distribution. In our case, the observations are seismic signals, the unobserved variables the events, and our goal is to compute the posterior

$$p(\text{events}|\text{signals}) \propto p(\text{signals}|\text{events})p(\text{events})$$

where the event prior $p(\text{events})$ and signal likelihood $p(\text{signals}|\text{events})$ are as defined by the model in Chapter 4.

Previous work on NETVISA (Arora et al., 2013) uses hill-climbing search to find an event history maximizing $p(\text{events}|\text{signals})$, i.e., the maximum a posteriori (MAP) estimate. This is not well defined for our model, because MAP estimates are not invariant to parametrization; for example, expressing event depth in meters rather than kilometers does not change the distribution over possible worlds implied by our model, but would divide the corresponding densities by a factor of 1000 and therefore "penalize" hypotheses containing larger numbers of events under a naive density-based optimization.

As an alternative to hill-climbing search, we apply reversible jump Metropolis Hastings (Section 3.2.1) to draw approximate samples from the posterior on event bulletins. This implements a form of stochastic hill-climbing, in a way that correctly accounts for the model parameterization, integrates over the uncertainty on latent variables, and provides posterior samples representing our uncertainty. The price of these advantages is significant additional implementation complexity, since each move must be formulated as a proposal distribution from which we can both sample and compute a density.

This chapter presents the basic structure of our RJMCMC algorithm, and describes the proposals used, including birth and death proposal moves for events and unassociated templates. Note that we do not apply MCMC to the modulation wavelet coefficients $\mathbf{w}$, which are instead marginalized out exactly as described in Section 4.9.

5.1 Envelopes and unexplained signals

We introduce some notation that will be broadly useful in the proposals that we define. The *envelope* $\mathbf{v}_j = \text{env}(\mathbf{s}_j)$ of a signal is an approximation of the signal amplitude at each point in time, which we compute using standard methods as a low-pass filtering of the signal's Hilbert transform (Kanasewich, 1981). The envelope of a signal is always nonnegative.

The *expected envelope* $\bar{\mathbf{v}}_j$ is given by the sum of envelope shapes g for currently modeled arrivals, plus a noise mean μ_j,

$$\bar{\mathbf{v}}_j = \mu_j + \sum_{i=1}^{N} \sum_{k \in \mathbf{h}_i} g(t - \tau_{i,j}^{(k)}; \theta_{i,j}^{(k)}) + \sum_{r=1}^{R} g(t - \tau_{r,j}^{UA}; \theta_{r,j}).$$

Subtracting the expected envelope from the observed envelope yields the *unexplained envelope* $\hat{\mathbf{v}}_j = \mathbf{v}_j - \bar{\mathbf{v}}_j$.

In addition to the envelope and unexplained envelope, we also define the *unexplained signal* $\hat{\mathbf{z}}_j$ (Figure 4.11d), as the posterior mean of the noise process under the linear Gaussian signal model of Section 4.9,

$$\hat{\mathbf{z}}_j = E\left[\mathbf{z}_j | \mathbf{s}_j, \mathbf{E}, \theta_{\mathbf{j}}\right].$$

That is, the unexplained signal contains all variance in the observed signal that could not be explained by currently modeled arrivals. If there are no arrivals at a station (or if all arrivals have near-zero amplitude), the entire signal will be explained under the noise model, in which case the unexplained signal is equal to the observed signal.

5.2 Algorithm overview

Our inference algorithm is structured as a cyclic sweep that performs Metropolis–Hastings steps to update all currently instantiated model variables in turn, while also proposing dimension-changing moves that birth new event hypotheses, kill existing events, and birth and kill unassociated arrivals. We first present the high-level algorithmic structure. Unless otherwise specified, all "updates" are Gaussian random walk proposals (eq. (3.3)) with manually tuned step sizes.

1. **Event attribute moves**: For each instantiated event $\mathbf{e}_i$, update, in turn, its latitude and longitude (jointly), depth, origin time, and magnitude. In some cases these proposals may also jointly propose new phase arrival times at some stations, birth new phases that would naturally be generated from the new location, or kill existing phases that are no longer plausible, as described in Section 5.6.

2. **Event birth/death moves**: Propose creating a new event or destroying an existing event, using one of the moves in Section 5.4 below.

3. **Event reproposal move**: Choose an event at random, and propose killing that event and re-birthing a new event from the resulting state, as a single joint move. This has the effect of allowing events to escape local modes by jumping to new locations.

4. **Event merge/split moves**: We implement a merge move by first choosing a pair of events with probability inversely proportional to the space-time distance between them,[1] then jointly proposing to kill both events and re-birth a new event from the resulting state. Similarly a split move chooses an event uniformly at random, and proposes killing that event while jointly proposing to birth two new events from the resulting state.

5. **Station-local moves**: For each station j in the network:

   a) **Station noise parameters**: propose updating the mean μ, variance σ^2, and AR parameters ϕ of the background noise process.

   b) **Unassociated arrival birth/death**: Propose a new unassociated arrival at this station, or kill an existing unassociated arrival (Section 5.3).

   c) **Swap associations**: Choose a pair of consecutive arrivals uniformly at random, and propose swapping their associations (if both arrivals are unassociated, this is a no-op).

   d) **Phase birth/death**: For each event with arrivals at this station, propose killing an existing phase or birthing a new phase for this event. The phase births use the same proposal distribution as event births, described in Section 5.4.4.

   e) **Shape parameters**: For each arrival at this station (associated or not), update its time and shape parameters θ (Section 4.5) using a random-walk proposal for each parameter. We also perform the following custom proposals:

      - **Onset length move**: jointly propose a new arrival time τ and onset length ρ, so as to change the onset length while leaving the peak time $\tau + \rho$ unchanged.
      - **Mode-jumping move**: propose a new peak time $(\tau + \rho)$ from a distribution proportional to the unexplained signal envelope $\hat{\mathbf{v}}_j$.
      - **Waveform alignment move**: propose a new arrival time from a distribution proportional to the Bayesian cross-correlation (B.4) of the signal predicted for this phase using historical data (expected modulation signal under the GP model, multiplied by the current envelope shape) against the current unexplained signal $\hat{\mathbf{z}}_j$. For events with strong historical waveform information, this allows inference to quickly snap to the correct alignment, which might otherwise be difficult to find by random walk moves.

[1] We arbitrarily equate a one-second difference in origin times with a 10km distance between surface locations.

5.3 Unassociated arrival birth and death moves

The birth proposal for unassociated arrivals proceeds according to the chain rule: first we propose an arrival time, then a peak time (which determines the onset period ρ), then (jointly) the amplitude and decay parameters:

$$q(\theta|\mathbf{s}_j) = q(\tau|\mathbf{s}_j)q(\rho|\tau, \mathbf{s}_k)q(\alpha, \gamma, \beta|\tau, \rho, \mathbf{s}_j).$$

The component proposals are given by:

1. **Arrival time**: proposed with probability proportional to the cube of an STA/LTA detector (Section 2.5.1).

2. **Peak time**: proposed with probability proportional to the positive part of the exponentiated unexplained envelope $\exp(\hat{\mathbf{v}}_j)$ within a short period (20s) following the arrival time, restricted to timesteps at which the envelope is increasing,

$$q(\rho = \tau + t|\tau, \mathbf{s}_j) \propto \exp(\hat{v}_j(t))^+ \cdot \mathbb{I}[0 < t - \tau < 20] \cdot \mathbb{I}\left[\frac{d\hat{v}_j(t)}{dt} > 0\right].$$

3. **Amplitude and decay parameters**: given the arrival and peak times, we run a gradient-based optimizer to minimize a surrogate signal likelihood given by considering the unexplained envelope under an iid Gaussian noise model,

$$\mathcal{L}(\alpha, \gamma, \beta) = \log \mathcal{N}(\hat{\mathbf{v}}_j(\alpha, \gamma, \beta), \mathbf{0}, \mathbf{I}).$$

   We then propose from a (three-dimensional) multivariate Gaussian with mean centered at the optimum and covariance given by the inverse Hessian of the surrogate log-likelihood, i.e., a Laplace approximation (MacKay, 2003).

The death proposal chooses an arrival to kill with probability inversely proportional to the amplitude α of each existing unassociated arrival. Thus we are more likely to propose killing small arrivals, for which the death proposals are likely to be accepted, than large arrivals, whose deaths would leave significant signal energy unexplained.

5.4 Event birth moves

An event birth proposal contains three steps: a *origin* proposal for the event $\mathbf{e}$, including surface location, depth, time, and magnitude, an *association* proposal at each station that decides whether to instantiate the event by birthing new phase arrivals from scratch or co-opting existing unassociated arrivals, and a *shape* proposal for the parameters θ of all newly generated phase arrivals. Because our phase existence model is near-deterministic (Section 4.4), when proposing an event it is necessary to jointly propose the parameters governing all of its phase arrivals along with the event origin itself.

Note that the associations themselves are not, formally speaking, variables in the probability model; "associating" an existing arrival is equivalent to killing that arrival and birthing a new phase with the same shape parameters. Formally speaking we construct a joint proposal by the chain rule,

$$q(\mathbf{e}, \theta^{(UA)}, \theta) = q(\mathbf{e})q(\theta^{(UA)}|\mathbf{e})q(\theta|\mathbf{e}, \theta^{(UA)}).$$

where $\theta^{(UA)}$ denotes the set of unassociated arrivals which may shrink as existing arrivals are associated with the newly birthed event.

In this section we describe each part of the proposal in turn. First we consider the event origin proposal $q(\mathbf{e})$, of which we implement three different variants:

- **Bayesian correlation birth**: proposes new events in the vicinity of training events that correlate well with the observed signals. This enables low-threshold detections of repeated events for which historical waveform data is available.
- **Hough transform birth**: proposes new events in locations that coherently explain some set of currently unassociated arrivals. This is essentially a form of multilateration, as in detection-based systems (using the unassociated arrivals as "detections"), and allows for the construction of *de novo* events.
- **Prior birth**: we also include a "dumb" proposal that simply generates events from the prior (Section 4.3). This serves to guarantee ergodicity, since in principle it can propose any event, and to increase the acceptance probability of death moves (cf. Smart-Dumb/Dumb-Smart MCMC, Wang and Russell, 2015).

The constructions of the correlation and Hough transform proposals are rather intricate; the following Sections 5.4.1 and 5.4.2 give details. In both cases we construct a proposal by defining a *surrogate* probability model that captures essential elements of the full SIGVISA model, but for which we can efficiently compute the posterior distribution. The surrogate posteriors are then used as a source of proposals that, we hope, will also be plausible under the full model.

Given a proposed origin location $\mathbf{e}$, Section 5.4.3 describes the proposal $q(R^{(UA)}|\mathbf{e})$ for associating existing arrivals. Finally, Section 5.4.4 describes our proposal $q(\theta|\mathbf{e}, R^{(UA)})$ for the envelope shapes of newly birthed phase arrivals.

5.4.1 Bayesian correlation proposal

We first describe our origin proposal based on waveform correlation, which generates event hypotheses near training events whose signals correlate with the observed data. This proposal makes use of a novel probabilistic extension of cross-correlation, described in Appendix B.

Before constructing the proposal, we pre-compute, for each station j, each phase k, and each *training event* i at location $\mathbf{x}_i$, our model's *predicted signal* $\bar{\mathbf{s}}_i$ for an event in that training location. That is, we construct the envelope shape $g(t; \bar{\theta}_i)$ generated by the posterior mean

parameters $\bar{\theta}_i$ under the GP model (4.9), and the expected modulation signal $\bar{\mathbf{m}}_i = \mathbf{A}\bar{\mathbf{w}}_i$, where $\bar{\mathbf{w}}_i$ are similarly the expected wavelet coefficients. These are multiplied to yield the predicted signal,

$$\bar{s}_i(t) = g(t; \bar{\theta}_i) \cdot \bar{m}_i(t).$$

The effect of this procedure is to query the model for its memory of the signal from training event i. Note that the model's reconstruction $\bar{\mathbf{s}}_i$ will differ from the signal originally observed, in that the reconstruction discards noise and, depending on learned noise and lengthscale hyperparameters, may also be influenced by the signals observed for other nearby training events.[2]

Given a signal prototype $\bar{\mathbf{s}}_i$ for each training event (for each phase at each station), we define a *surrogate probability model* for the currently unexplained signals $\hat{\mathbf{z}}_j$. By modeling the unexplained signal, we avoid re-proposing events that already exist. Our surrogate model corresponds to the following generative story:

1. First, we select a training event index i and sample an origin time $\mathbf{e}^{\text{time}}$ uniformly at random.

2. At each station j, for each arriving phase k, we sample an arrival time $\tau_j^{(k)}$ from the travel time model, conditioned on the origin time and the training location $\mathbf{x}_i$. We also sample an amplitude $\alpha_j^{(k)}$ from a flat prior.[3]

3. Each unexplained signal $\hat{\mathbf{z}}_j$ is generated as autoregressive noise ζ_j, plus the scaled predicted signals from our chosen training event,

$$\bar{z}_j(t) = \zeta_j(t) + \sum_k \alpha_j^{(k)} \bar{s}_{j,i}^{(k)}(t - \tau_j^{(k)}).$$

   By construction of this model, the likelihood of $\hat{\mathbf{z}}_j$ under a particular hypothesis for $\tau_j^{(k)}$ (optimizing over $\alpha_j^{(k)}$) is equal to the Bayesian correlation statistic (B.8).

Our proposal proceeds in two steps. We first compute an (approximate) *posterior* on the training event index i and origin time $\mathbf{e}^{\text{time}}$, given the unexplained signals $\hat{\mathbf{z}}_j$. The model is constructed so that this can be done efficiently. We then propose a new event $\mathbf{e}$ with time sampled from the surrogate posterior, and location (and depth) sampled from a mixture of Gaussian distributions centered at the training events, with mixture weights given by the posterior on training indices i.

The approximate posterior computation proceeds as follows. First, for each station and phase, and for each training event i, we consider the hypothesis that an event exists during our test period at location $\mathbf{x}_i$ generating the signal $\bar{\mathbf{s}}_{j,i}^{(k)}$, arriving at time $\tau_{j,i}^{(k)}$. For each time

[2] More practically, since at inference time we are guaranteed to have trained GP models in hand, relying on their reconstructions saves the burden of preserving a separate database of historical training signals.

[3] The use of a flat prior weakens the generative story, but allows $\alpha_j^{(k)}$ to be optimized analytically (B.6).

step t we compute the Bayesian correlation (B.7), yielding the *signal likelihood* $L(\tau = t)$ given by eq. (B.9).

We next marginalize out the travel time to yield an origin time likelihood for each station and phase. Let $p(\tau = t|\mathbf{x}_i, \mathbf{e}^{\text{time}})$ be a model of the travel time from location $\mathbf{x}_i$ to the current station (defined by a GP, Section 4.5). Then the likelihood of a signal given an origin time $\mathbf{e}^{\text{time}}$ is

$$p(\hat{\mathbf{z}}_j|\mathbf{x}_i, \mathbf{e}^{\text{time}}, \bar{\mathbf{s}}_{j,i}^{(k)}) = \int_{-\infty}^{\infty} L(\tau = t)p(\tau|x, \mathbf{e}^{\text{time}})d\tau. \tag{5.1}$$

In practice, we truncate the tails of the travel time model past a certain point (25s), so given a vector of values for the arrival time likelihood $\mathbf{L}$, the origin time likelihood is computed efficiently by convolution with a finite-width travel time distribution.

We assume that the signal at each station is independent of other stations and that each phase acts independently, so the likelihood of a hypothesized origin time is the product over stations and phases,

$$p(\hat{\mathbf{z}}|\mathbf{x}_i, \mathbf{e}^{\text{time}}, \bar{\mathbf{s}}_i) = \prod_j \prod_k p(\hat{\mathbf{z}}_j|\mathbf{x}_i, \mathbf{e}^{\text{time}}, \bar{\mathbf{s}}_{j,i}^{(k)}).$$

Since we assume a uniform $\frac{1}{T}$ prior on origin times, this likelihood is proportional to the posterior for any given training event i. The normalizing constant is the marginal likelihood $\ell(\mathbf{x}_i)$ of an event at the training location $\mathbf{x}_i$,

$$\ell(\mathbf{x}_i) = \frac{1}{T} \sum_{t=1}^{T} p(\hat{\mathbf{z}}|\mathbf{x}_i, \mathbf{e}^{\text{time}} = t, \bar{\mathbf{s}}_i).$$

Since we treat time as discrete, this marginalization is computed as an explicit sum. The marginal likelihood will be large if the predicted signals from an event at the training location correlate well with the observed signals across many stations, at the times predicted by the travel time model. Thus, like the full SIGVISA model, our proposal is sensitive not just to correlations at a single station, but to *coherent* correlations across multiple stations.

Given the marginal likelihood $\ell(\mathbf{x}_i)$ for each historical training event i, we use these as weights in a Gaussian mixture model proposal. That is,

1. For each training event i, we define a proposal distribution q_i that samples an origin location from a small Gaussian around $\mathbf{x}_i$ and an origin time from the approximate posterior $p(\mathbf{e}^{\text{time}}|\hat{\mathbf{z}}, \mathbf{x}_i, \bar{\mathbf{s}}_i) \propto p(\hat{\mathbf{z}}|\mathbf{x}_i, \mathbf{e}^{\text{time}}, \bar{\mathbf{s}}_i)$.

2. The overall proposal distribution q is a mixture of the q_i, with weights given by normalizing the likelihoods $\ell(\mathbf{x}_i)$:

$$q(\mathbf{e}) = \sum_i \frac{\ell(\mathbf{x}_i)}{\sum_{i'} \ell(\mathbf{x}_{i'})} q_i(\mathbf{e}).$$

Thus our event proposal distribution is a mixture of Gaussians centered at the training event locations, with weights proportional to how well each expected signal correlates (coherently) with the signals we observed.

To compute this proposal quickly in practice, we precompute the origin time likelihoods (5.1) at each station, correlating each signal against all training events, so that proposing a new event requires only evaluating the marginal likelihoods $\ell(\mathbf{x}_I)$ and sampling from a mixture of Gaussians. As new events are born, we do not compute the origin time likelihoods using the updated unexplained signal $\hat{\mathbf{z}}$ that incorporates the new events; instead, we heuristically update the cached origin time likelihoods to zero out any regions corresponding to existing event phases. Since this heuristic is a function only of the current model state, this approximation preserves detailed balance. We also use the i.i.d. version of the Bayesian correlation log odds (B.4) in place of the autoregressive log odds (B.8), as the i.i.d. version is much faster to compute and appears to provide proposals of comparable quality.

5.4.2 Hough location proposal

We next consider another origin proposal, intended to generate events that coherently explain signals observed at multiple stations, even in the absence of historical correlations. This is done using a generalized Hough transform (Duda and Hart, 1972). That is, we define an *accumulator array* in which each bin represents an event hypothesis, and we allow each unassociated arrival to "vote" for those bins that could plausibly have generated it. Bins corresponding to genuine events will tend to receive votes from many stations, and so are more likely to be proposed.

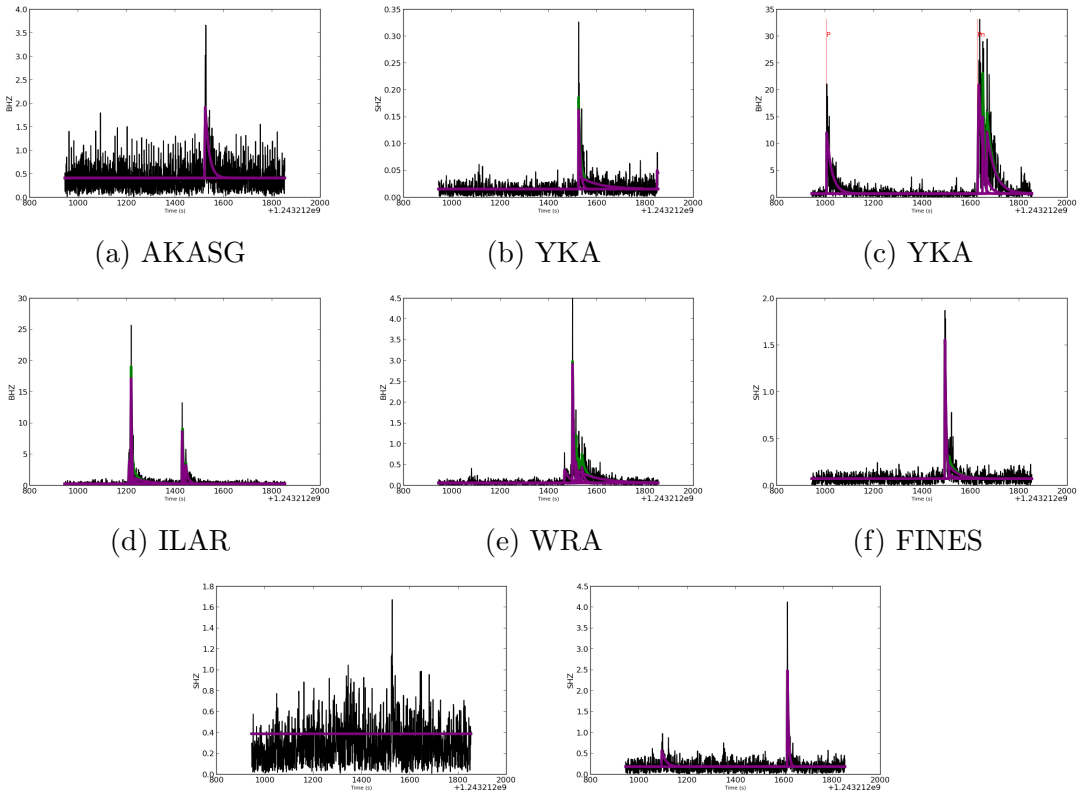
Concretely, our accumulator array is a 5D array with dimensions corresponding to those of the proposed event $\mathbf{e}$, i.e., longitude, latitude, depth, time, and magnitude. The score of each bin is computed via a surrogate probability model, described below, as the likelihood of the observed data (unassociated arrivals) under the hypothesis that an event exists in the bin's region of spacetime and magnitude. This surrogate model has the flavor of a simplified NET-VISA (Arora et al., 2013), in which the unassociated arrivals play the role of detections. It can also be seen simply as a sophisticated Bayesian voting scheme, in which the "votes" cast into different bins by a particular arrival are weighted according to the (log) probability that each bin could have generated it.

Our surrogate probability model is defined by the following generative story:

1. We sample an event $\mathbf{e}$ from the model prior (Section 4.3); this event will fall in some bin b.

2. We also sample a set of unassociated arrivals at each station j, following the prior of Section 4.7.

3. At each station, we compute the set of legal phases $\mathbf{h}_j$ generated by the sampled event $\mathbf{e}$. For each phase $k \in \mathbf{h}_j$, we sample a Boolean *detection* variable d_k with probability δ_k given by a detection model described below. If the phase is detected, we generate



(a) AKASG (b) YKA (c) YKA

(d) ILAR (e) WRA (f) FINES

(g) ASAR (h) NVAR

Figure 5.1: Unassociated arrivals generated by MCMC at several stations recording the 2009 DPRK event.

a new arrival by sampling from the phase model $p(\theta_{j,i}^{(k)}|\mathbf{e})$, eq. (4.9). Each such arrival is added to the set of unassociated arrivals at its station.

Note that although the phase arrivals are sampled conditioned on the event $\mathbf{e}$, they are not "marked" in the final output; a sample from this surrogate model consists of a set of undifferentiated arrivals at each station. To generate an event proposal from this model, we observe those undifferentiated arrivals as the *unassociated* arrivals θ present in the current inference state, under the assumption that some of these unassociated arrivals are in fact generated coherently by some latent event $\mathbf{e}$. As with the surrogate model we defined for waveform correlation proposals, this model is defined so that we can perform an (approximate) posterior calculation efficiently by a sequence of feedforward steps, yielding a closed-form proposal density for the event $\mathbf{e}$ in terms of posterior probabilities $p(b|\theta)$ for each bin.

At a high level, the posterior on bins is given by Bayes' rule,

$$p(b|\theta) \propto p(\theta|b)p(b),$$

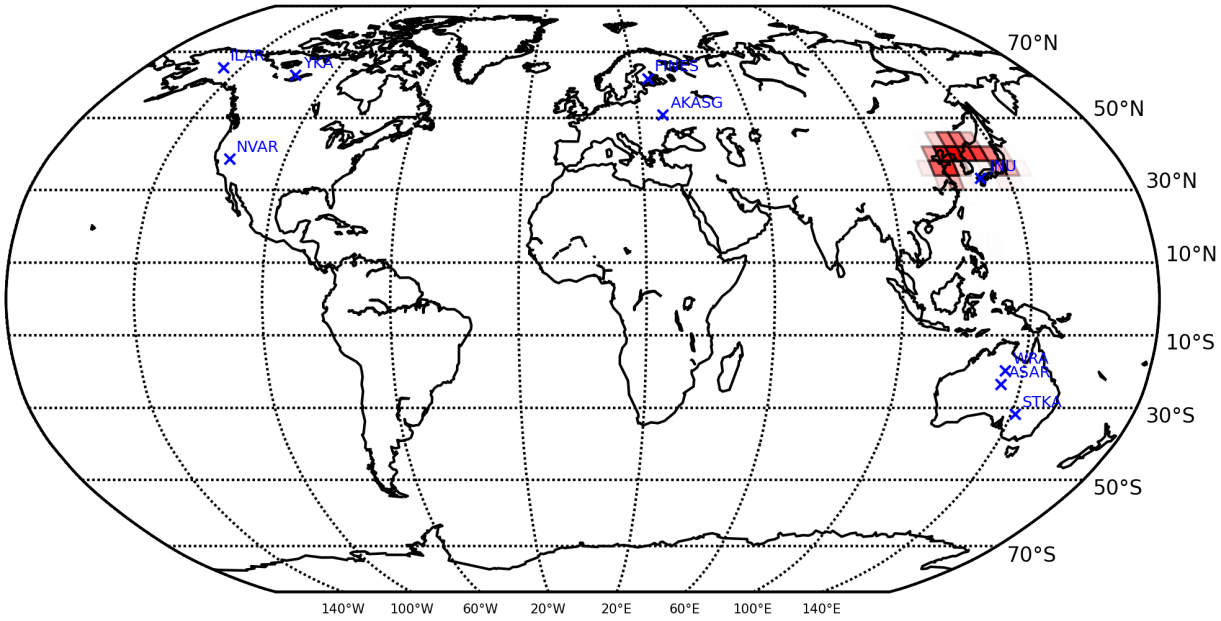


Figure 5.2: Hough transform of the unassociated templates in Figure 5.1.

where the prior probabilities $p(b)$ are precomputed by numerical quadrature over the event prior density (Section 4.3) within each bin. We propose an event $\mathbf{e}$ by computing a posterior probability for each bin in the array, sampling a bin b from this distribution, and proposing from a uniform distribution within the bin,

$$q(\mathbf{e}|\theta) = \sum_b p(b|\theta) \cdot \frac{\mathbb{I}[\mathbf{e} \in b]}{\text{volume}(b)}. \tag{5.2}$$

To do this, we must compute the likelihood $p(\theta|b)$, which is really a *marginal* likelihood since we must sum over the unknown location of an event $\mathbf{e}$ within the bin, and over phase associations, i.e., which of the observed arrivals θ were generated by that event. We treat the former marginalization using a combination of analytic derivation and explicit summation, and the latter by a greedy maximization, as described in the following sections. We simplify the likelihood calculation by modeling *only* the arrival time and amplitude, i.e., we define $\theta = (\tau, \alpha)$ in the following sections, though in principle other shape parameters could be included as well.

5.4.2.1 Marginalizing over location within a bin

Since we do not know where within each bin an event might have occurred, we must integrate over event descriptions $\mathbf{e} \in b$ to yield a probability covering the entire bin. We assume a uniform prior[4] within each bin on the event's surface location and depth (summarized as

[4]Ideally we would use the true event prior (4.4), but a uniform prior simplifies the calculations.

"location"), time and magnitude. Integrating over the specific event yields

$$\begin{aligned} p(\theta_j^{(r)}|\text{bin } b) &= \int p(\theta_j^{(r)}, \mathbf{e}|\text{bin } b)d\mathbf{e} \\ &= \int p(\theta_j^{(r)}|\mathbf{e})p(\mathbf{e}|\text{bin } b)d\mathbf{e} \\ &= \int p_E(\tau|\mathbf{e})p_E(\alpha|\mathbf{e})p(e|\text{bin } b)d\mathbf{e} \\ &= \left(\int p_E(\tau|\mathbf{e})p(e^{\text{time}}|b)p(\mathbf{e}^{\text{loc}}|b)de^{\text{time}}d\mathbf{e}^{\text{loc}}\right)\left(\int p_E(\alpha|\mathbf{e})p(e^{\text{mb}}|b)de^{\text{mb}}\right) \\ &= f(\tau, b)f(\alpha, b). \end{aligned}$$

That is, we exploit independence to decompose the bin likelihood into the product of an arrival time "score" $f(\tau, b)$ and an amplitude score $f(\alpha, b)$. We interpret this as each arrival casting (weighted) votes for all bins whose locations could plausibly have generated it under the travel time model, as well as all bins whose magnitudes could plausibly have generated it under the amplitude model. These scores are computed separately, as described in the following sections, then combined to form the final accumulator array.

5.4.2.2 Amplitude

We write the amplitude score in terms of a source amplitude and a transfer function,

$$\begin{aligned} f(\alpha, b) &= \int p_E(\alpha|\mathbf{e})p(e^{\text{mb}})de^{\text{mb}} \\ &= \int p(\text{transfer} = \log\alpha - S(e^{\text{mb}}))p(e^{\text{mb}})de^{\text{mb}}. \end{aligned}$$

We assume a uniform prior on magnitude within each bin, $p(e^{\text{mb}}) = Unif(mb_1, mb_2)$. The source (log) amplitude $S(e^{\text{mb}})$ is taken to be a deterministic function of the event magnitude, as well as the arriving phase, and frequency band. We use a Brune source model (Brune, 1970), and model the transfer function in log space using a GP, as described in Section 4.5, so that for a given location we obtain a Gaussian $p(\text{transfer}) = N(\mu, \sigma^2)$. Formally speaking, we should integrate the GP model of amplitude transfer over event locations within the bin, but this makes little difference for small bins; we just use the bin center for simplicity. The amplitude score is computed as

$$\begin{aligned} f(\text{amp}, b) &= \frac{1}{\text{mb}_2 - \text{mb}_1}\int_{mb_1}^{mb_2} p(\text{transfer} = \log\alpha - S(e^{\text{mb}}))de^{\text{mb}} \\ &\approx \frac{1}{\text{t}_2 - \text{t}_1}\int_{t_1}^{t_2} p(\text{transfer} = t)dt \\ &= \frac{1}{\text{t}_2 - \text{t}_1}\left(\Phi\left(\frac{t_2 - \mu}{\sigma^2}\right) - \Phi\left(\frac{t_1 - \mu}{\sigma^2}\right)\right), \end{aligned}$$

where Φ is the CDF of a standard Gaussian, and $t_1 = \log\alpha - S(\text{mb}_1)$ and $t_2 = \log\alpha - S(\text{mb}_2)$ are the transfer function values corresponding to the edge-of-bin magnitudes mb_1 and mb_2 respectively. There is an approximation in the second step, where we perform the change of variables assuming that $\frac{dmb}{dt} = \frac{mb_2 - mb_1}{t2 - t1}$, that is, that the source model S is linear over the magnitude range contained in the bin. The Brune source model is defined analytically, so in principle the exact derivative could be used here, but the linear approximation is simple, reduces implementation complexity, and seems reasonable for small bin sizes.

5.4.2.3 Arrival time

The arrival time score for each bin is complicated somewhat by the need to integrate over multiple unknowns: the origin time and the origin location. We treat these two sources of uncertainty separately.

We first consider the integral over the origin time. Here we model travel time residuals using a Laplace(0, β) distribution with fixed width of $\beta = 5$ seconds.[5] For a bin with origin time bounds $[t_1, t_2]$, this gives the marginal travel-time likelihood

$$\begin{aligned} p_E(\tau | e^{\text{loc}}, \text{bin } b) &= \int p_E(\tau | e^{\text{time}}, e^{\text{loc}}) p(e^{\text{time}} | \text{bin } b) de^{\text{time}} \\ &= \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} p_E(\tau | e^{\text{time}}, e^{\text{loc}}) de^{\text{time}} \\ &= \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} p_E(\text{travel time} = \tau - e^{\text{time}}) de^{\text{time}} \\ &= \frac{1}{r_2 - r_1} \int_{r_1}^{r_2} \text{Laplace}(r; 0, \beta) dr, \end{aligned}$$

where we change variables in the final line to work in terms of the travel time residual, so that $r_1 = (\tau - t_1) - E[\text{travel time}]$ is the residual for an event with origin time t_1, and r_2 is defined analogously. This integral defines a partial "score" for the bin b; partial because is still conditioned on a particular location e^{loc}. Letting L denote the CDF of a Laplace$(0, \beta)$ distribution, the location-specific score is

$$f(\tau, \text{bin } b, e^{\text{loc}}) = \frac{1}{r_2 - r_1}(L(r_2) - L(r_1)).$$

We next integrate over locations within the (lon, lat, depth) bin. This is approximated by a finite sum over nine "representative" locations $(x_1, \ldots, x_9)$, namely the bin center and

[5] We could use the full model's location-specific GP residual model, but doing so would complicate the calculations. The heavier Laplace tails are also useful to compensate for location uncertainty within the bin.

the eight corners.[6]

$$f(\tau, \text{bin } b) = E_{p(e^{\text{loc}}|\text{bin } b)}\left[f(\text{arrival time}, \text{bin } b, e^{\text{loc}})\right]$$
$$\approx \frac{1}{9}\sum_{i=1}^{9} f(\text{arrival time}, \text{bin } b, x_i).$$

As an implementation note, both the Laplacian and Gaussian CDFs are computed efficiently via linear interpolation on a precomputed lookup table.

5.4.2.4 Greedy associations

Along with the specific event locations, we are also uncertain about the associations: which templates exactly are arriving phases from this event. Assume for simplicity that we are modeling only P and S phases. Either phase might fail to generate a template; let binary variables d_P and d_S denote whether P and S arrivals respectively are actually detected, following Bernoulli priors with probabilities δ_P, δ_S. These probabilities are given by a logistic regression for each phase and station, with features based on the event magnitude, depth, event–station distance, and fit to data produced by running inference (including unassociated birth and death moves) on signals from training events.

If a P arrival is detected at a station, then let t_P denote the index of that arrival (within the set of all arrivals at that station, sorted by arrival time), and similarly for t_S. The probability of a detection hypothesis (d_P, d_S) is given by a sum over associations t_P, t_S realizing that hypothesis.[7]

Summing over detection and association variables yields the marginal likelihood for all

[6] For larger bins these nine locations may not be sufficient; there is probably room for improvement here.

[7] To be precise, the generative model is that we first sample random variables "arrival time of P phase", etc. from appropriate conditional distributions. We don't observe these variables directly; instead we observe unlabeled order statistics, e.g. "arrival time of first template". In high-level notation, if L represents the "labeled" templates as generated by the model, and O represents the "observed" order statistics, the marginal likelihood is $p(O) = \int p(L)p(O|L)dL$, where $p(O|L)$ is a Dirac delta that activates when O=sorted(L) and is 0 otherwise. This determinism means that only a finite set of hypotheses for L contribute nonzero probability, namely exactly those that simply "label" the observed templates O without actually changing any of the numbers. So the integral over L is equivalent to a sum over labelings. This might be obvious, but I often found myself tempted to treat the associations as explicit model variables with their own prior, which is *not* correct since the generative story does not involve sampling labels from a prior.

station templates θ_j,

$$\begin{aligned} p_E(\theta_j|B) =& p(d_P, d_S) \sum_{t_P \neq t_S} p_E(\theta_j^{t_P}|B) p_E(\theta_j^{t_S}|B) p_{UA}(\theta_j \setminus (t_P, t_S)) \\ &+ p(d_p, \neg d_S) \sum_{t_P} p_E(\theta_j^{t_P}|B) p_{UA}(\theta_j \setminus t_P) \\ &+ p(\neg d_p, d_S) \sum_{t_S} p_E(\theta_j^{t_S}|B) p_{UA}(\theta_j \setminus t_S) \\ &+ p(\neg d_p, \neg d_S) p_{UA}(\theta_j). \end{aligned}$$

Because this sum grows exponentially large with increasing numbers of phases, we prefer not to compute it explicitly. Instead we perform a greedy approximation obtained by iteratively choosing the most likely association for each modeled phase. That is, we compute the likelihoods of all possible P associations, including the null association corresponding to $\neg d_P$, while modeling all other templates as unassociated. We then fix the maximum-likelihood P association, and then compute likelihoods for all possible S associations where $t_S \neq t_P$. This greedy maximization is not guaranteed to give the globally most-likely solution (if the chosen P association would also have been the most likely S association), but it has the advantage of being linear rather than exponential in the number of phases.

5.4.3 Associations

Given the proposed event origin, we propose which currently unassociated arrivals, if any, should be associated with the new event. This is done independently at each station. We describe this proposal here for a generic station, having R currently unassociated arrivals given by parameters $(\theta_r)_{r=1}^R$.

We first compute (deterministically) the set $\mathbf{h}$ of legal phases at this station, given the event location, as described in Section 4.4. As part of the event proposal, each of these phases will be either associated with an existing arrival, or birthed from scratch, though later inference moves may kill them. We then enumerate all possible *joint associations*, where a joint association consists of a function from the legal phases $\mathbf{h}$ to $\{1, \ldots, \mathrm{R}, \texttt{NULL}\}$. That is, a joint association maps each phase either to an existing arrival, or to the `NULL` arrival so that it is birthed from scratch. We require that associations are one-to-one with respect to existing arrivals, so that no arrival is associated with multiple phases.

We compute a score for each joint association ν, consisting of a product of odds ratios for each phase,

$$\mathrm{score}(\nu) = \prod_{k \in \mathbf{h}} \frac{p_{GP}(\theta_{\nu(k)}|\mathbf{e})}{p_{UA}(\theta_{\nu(k)})}, \tag{5.3}$$

where $\theta_{\nu(k)}$ are the parameters of the currently unassociated arrival $\nu(k)$ proposed for association with phase k. Each odds ratio compares the probability of these parameters under a GP model conditioned on the event location (Section 4.5), to the probability under the

prior on unassociated arrivals (Section 4.7), so that existing arrivals that are highly likely to have been generated by the proposed event (because they have a plausible arrival time, amplitude, and other properties) are given high scores. For phases not associated with any existing arrival, $\nu(k) = \texttt{NULL}$, we define the odds ratio to be 1, so that we can increase the score only by associating those phases with arrivals that are better explained under the event hypothesis than as unassociated templates.

Given scores for all possible joint associations at a station, we sample a joint association to propose with probability proportional to its score. Note that the hard constraint on travel time residuals (eq. (4.9)) allows us to discard immediately the vast majority of possible associations, and explicitly compute scores only for the remaining few.

As noted above, the associations themselves are not reified variables within our probability model, instead, the effect of this proposal is structural: the overall event birth proposal will shift the model to a new state in which we delete any existing arrivals that have been 'associated', but re-use their parameters as the parameters of newly birthed phase arrivals, so that the proposal distribution for those phase arrivals is a delta function. The shape parameters for all other phase arrivals must be proposed more explicitly, as we describe in the next section.

5.4.4 Shape parameters

We propose envelope shape parameters θ, conditioned on the event location $\mathbf{e}$ and the unexplained envelope $\hat{\mathbf{v}}_j$, at each station j. Our proposal consists of three stages: initial heuristic proposals $q(\tilde{\theta}_j^{(k)}|\mathbf{e}, \hat{\mathbf{v}}_j)$ for all legal phases k that have not already been associated with an existing arrival, which are then fine tuned by a small number of MCMC steps to yield optima $\theta_j^{(k)*}$, after which we propose the final shape parameters from the neighborhood of these optima.

5.4.4.1 Heuristic proposal

The heuristic proposal $\tilde{q}$ operates independently for each phase k, and begins by sampling the rise time ρ and decay parameters β, γ from the event-conditional Gaussian process prior (Section 4.5). Given these parameters, we propose the arrival time τ, using a proposal similar to that for unassociated arrivals (Section 5.3), proportional to the (exponentiated, positive part of the) unexplained envelope $\hat{\mathbf{v}}_j$ at the peak time $\tau + \rho$,

$$\tilde{q}(\tau|\rho, \hat{\mathbf{v}}_j) \propto \exp(v_j(\tau + \rho))^+ \cdot p_{GP}(\tau|\mathbf{e}),$$

where we also include the event-specific prior $p_{GP}(\tau|\mathbf{e})$ to force the proposed arrival time to be consistent with the travel time model. We then finally propose the amplitude α from a piecewise linear approximation to the posterior density, constructed via a grid search given the other parameters and observed signal. The overall heuristic is thus given by the factored density:

$$\tilde{q}(\theta|\mathbf{e}, \hat{\mathbf{v}}_j) = p_{GP}(\rho|\mathbf{e})p_{GP}(\beta|\mathbf{e})p_{GP}(\gamma|\mathbf{e})\tilde{q}(\tau|\mathbf{e}, \hat{\mathbf{v}}_j, \rho)\tilde{q}(\alpha|\mathbf{e}, \hat{\mathbf{v}}_j, \tau, \rho, \beta, \gamma).$$

5.4.4.2 Fine tuning

The heuristic proposal $\tilde{q}$ often works well, but it can be dangerously myopic. For example, since it samples the arrival time τ before the amplitude α, it might propose an otherwise-unlikely arrival time in order to fit a large spike in the signal, without realizing that the prior on amplitude won't actually allow such a large fit. Since the heuristic proposals are also independent for each phase, we may end up proposing explanations that are poor when considered jointly, e.g., two phases each explaining the same observed spike so that their combined amplitude is double that of the signal. As an improvement, we'd like to allow for more flexible adjustment to the signal before deciding whether to accept or reject the event.

Our approach is to use the heuristic proposal $\tilde{\theta}$ as a starting point for additional optimization, in which we apply a small number (25 epochs) of Metropolis–Hastings steps, including random-walk proposals on each shape parameter as well as the custom peak-invariant, mode-jumping, and alignment moves described in Section 5.2. We perform this MH optimization jointly for *all* new phases proposed at a station, so that the resulting proposals are co-adapted.

This additional optimization is justified within the larger inference framework by an auxiliary-variable construction of Storvik (2011), in which $\tilde{\theta}$ and the intermediate MH steps are treated as auxiliary variables $\mathbf{y}'$ (using Storvik's notation), which are jointly proposed along with a new model state $\mathbf{x}'$ under an extended target distribution $\pi(\mathbf{x}', \mathbf{y}') = \pi(\mathbf{x}')q(\mathbf{y}'|\mathbf{x}')$ to equal their proposal distribution $q(\mathbf{y}'|\mathbf{x}') = \tilde{q}(\mathbf{y}'|\mathbf{e}, \hat{\mathbf{v}}_j)$ (that is, although we propose the parameters conditioned on our event proposal $\mathbf{e}$ and the current unexplained envelope $\hat{\mathbf{v}}_j$, we could in fact derive these quantities and repropose the parameters at any time from the new state $\mathbf{x}'$), so that their values cancel from the acceptance ratio and do not need to be stored.

5.4.4.3 Final proposal

At the conclusion of the MH steps, we treat the resulting parameters $\theta_j^{(k)*}$ for each phase as approximate optima. Our final proposal is then constructed as a product of proposals in the neighborhoods of these optima. That is, we propose each phase k conditioned on the approximate optima as well as the parameters $\theta_j^{(:k)}$ proposed for previous phases,

$$q(\theta_j|\mathbf{e}, \mathbf{s}_j, \theta_j^*) = \prod_k q(\theta_j^{(k)}|\mathbf{e}, \mathbf{s}_j, \theta_j^{(:k)}, \theta_j^{(k:)*}), \tag{5.4}$$

where the per-phase proposals are factored over parameters using the chain rule, so that each parameter is proposed conditioned on the previous ones, and the proposals themselves are from piecewise linear approximations $\tilde{p}$ to the true model posterior density, constructed via a grid search,

$$\begin{aligned} q(\theta_j^{(k)}|\mathbf{e}, \mathbf{s}_j, \theta_j^{(:k)}, \theta_j^{(k:)*}) = &\tilde{p}(\tau|\rho^*, \alpha^*, \gamma^*, \beta^*)\tilde{p}(\rho|\tau, \alpha^*, \gamma^*, \beta^*) \\ &\tilde{p}(\alpha|\tau, \rho, \gamma^*, \beta^*)\tilde{p}(\gamma|\tau, \rho, \alpha, \beta^*)\tilde{p}(\beta|\tau, \rho, \alpha, \gamma) \end{aligned} \tag{5.5}$$

The purpose of this final proposal is to produce values well adapted to the signal and to each other, while maintaining the ability to evaluate the proposal density in closed form to compute an acceptance ratio (this is why we cannot, for example, just propose the result θ^* from the MH optimization).

5.5 Event death moves

Proposing to kill an existing event is much simpler than birthing a new event, in that it involves far fewer decisions. We first sample an event i to kill, with probability

$$q_{\text{kill}}(i|\mathbf{s}, \mathbf{e}_i, \theta_i) \propto \prod_j \prod_k \frac{p_{UA}(\theta_{j,i}^{(k)})}{p_{GP}(\theta_{j,i}^{(k)}|\mathbf{e}, \mathbf{s}_j)} \tag{5.6}$$

proportional to the probability that its phase arrivals could be better explained as unassociated than as being generated by the event (since the prior on unassociated arrivals favors small amplitudes, this will also tend to kill events with no high-amplitude arrivals). We also include a uniform component with probability .5, so that every event has some probability of being killed This is necessary in part because death probabilities appear in the acceptance ratios of birth moves, so our death move must have some probability of proposing to kill even very well-justified events, in order for those events to be born.

Given an event to kill, we could simply propose to delete that event and all of its phase arrivals. However, such proposals will be rejected if they leave unexplained any significant spikes in the signal that were previously modeled as a phase from the deleted event. To improve mixing, we allow a deleted event to *deassociate* some of its phase arrivals instead of deleting them. Similarly to associations during event births (Section 5.4.3), this is formally treated as a joint move in which we delete the event while simultaneously proposing to birth a new set of unassociated arrivals, with shapes matching those of the deleted phases.

As part of the death proposal, we sample independently, for each phase at each station, whether to delete or deassociate that phase. We do this by computing a deletion score,

$$q_{\text{delete}} = \frac{p(\mathbf{s}_j|\mathbf{E}, \theta_j \setminus \theta_{j,i}^{(k)}, \psi_j)}{p(\mathbf{s}_j|\mathbf{E}, \theta_j, \psi_j)},$$

which evaluates the effect on the marginal signal likelihood (4.14) of deleting the phase arrival $\theta_{j,i}^{(k)}$, and a deassociation score,

$$q_{\text{deassociate}} = \frac{p(R_j + 1)}{p(R_j)},$$

corresponding to the penalty under the prior for adding an additional unassociated template. The deassociation proposal probability for each phase is then obtained by normalizing the two scores,

$$q(\text{deassociate}_{i,j,k}) = \frac{q_{\text{deassociate}}}{q_{\text{delete}} + q_{\text{deassociate}}}.$$

5.6 Event location, depth, and time moves

Once an event birth proposal has been accepted, future inference epochs use random-walk proposals to explore the event's surface location, depth, and origin time (we also propose changes to the magnitude, but these are straightforward and do not require the machinery described in this section). At least two complications arise with such proposals:

- **Hard phase constraints**. Although the phase existence model described in Section 4.4 is deliberately "softened", many event-phase combinations are still illegal under our model. For example, the model institutes a hard shadow zone for P wave arrivals further than 98 degrees from a station. If an event is at distance 97.9 degrees from a station where it generates a P arrival, then any attempt to move it 0.1 degrees further will be automatically rejected unless we make a *joint* proposal to move the event and kill the P arrival. To maintain detailed balance, there must also be the possibility of making the reverse move, i.e., birthing a new P arrival when an event moves into the region where such an arrival is legal.

- **Coupled arrival times**. In many cases it is desirable to jointly propose new phase arrival times along with the event location. Consider, for example, a low-magnitude event that generates visible arrivals at only 3 of 100 stations in a large monitoring network. At the other 97 stations, arrivals are still present in the model but with amplitudes below the noise level. Intuitively, these non-detecting stations should pose no constraint on the event's location. We can achieve this by jointly proposing a new event location along with new arrival times at these stations, so that the travel time *residuals* remain constant. However, this is exactly the wrong strategy at the other three stations, where any attempt to change arrival times will arouse the righteous anger of the signal model. Thus we need the structure of our proposal itself to adapt in a way that is informed by the observed signal at each station.

We handle hard phase constraints with a straightforward joint proposal. For each event move, and each station, we compute the set of legal phases at both the current and proposed event location. Any existing phases that are newly illegal are killed. Any phases that are newly legal are birthed with probability given by the phase existence model (Section 4.4). The phase birth and death proposals follow the same machinery as event births and deaths: proposing associations (Section 5.4.3) followed by shape parameters (Section 5.4.4) for phase births, and deassociations (Section 5.5) for phase deaths.

To address coupling between arrival times and the event location, we implement an *adaptive coupling* proposal. Let $\tau_{j,i}^{\text{current}}$ denote the set of arrival times for all phases of event i arriving at station j, and $\tau_{j,i}^{\text{shifted}}$ denote the shifted arrival times that would preserve travel-time residuals under the newly proposed event location $\mathbf{e}'$. Then we propose to use the shifted times with probability proportional to the likelihood (4.14) of the signal $\mathbf{s}_j$ at

that station under the shifted times, weighted by the travel time model $p_{GP}(\tau_{j,i}|\mathbf{e}')$ (4.9),

$$q(\tau_{i,j} = \tau_{j,i}^{\text{shifted}}|\mathbf{e}', \mathbf{s}_j, \theta_j) = \frac{p(\mathbf{s}_j|\mathbf{e}', \tau_{j,i}^{\text{shifted}}, \theta_j)p_{GP}(\tau_{j,i}^{\text{shifted}}|\mathbf{e}')}{p(\mathbf{s}_j|\mathbf{e}', \tau_{j,i}^{\text{current}}, \theta_j)p_{GP}(\tau_{j,i}^{\text{current}}|\mathbf{e}') + p(\mathbf{s}_j|\mathbf{e}', \tau_{j,i}^{\text{shifted}}, \theta_j)p_{GP}(\tau_{j,i}^{\text{shifted}}|\mathbf{e}')}, \tag{5.7}$$

and otherwise our proposal leaves the arrival times unchanged. This has the effect that we jointly propose new, shifted arrival times to satisfy the travel time model at stations where doing so has no effect on the signal likelihood, but at stations where signals are highly informative we leave the arrival times unchanged since they are (presumably) already well adapted to the signal.

5.7 Parallel inference

Running SIGVISA on a large dataset requires significant parallelization; we support parallel inference by partitioning the data set over time. For the two week test set considered in Chapter 7, we partition the test period into 168 blocks of two hours, and run an MCMC chain for each block separately in parallel. Each chain is restricted to inferring events within its assigned time block, but has access to signal data for an additional period following, since phases from an event near the end of one block may not arrive until the following block. In principle, this partitioning modifies the stationary distribution slightly, by preventing "explaining away" effects from propagating in time: arrivals during one block could be explained by events in a previous block, but the inference procedure will not be able to exploit this. This could be corrected but does not appear to have been a major practical issue in these experiments.

As part of the restriction to the western US, we constrain all inference moves to propose only locations within that region (effectively setting the prior probability of outside locations to zero). The signals do contain some evidence of events outside the inference region, but this was not a major problem and appears to be satisfactorily handled by the mechanism of unassociated arrivals.

5.8 Constructing bulletins

Although the SIGVISA model defines a full posterior distribution over event bulletins, for evaluation purposes it is necessary to produce a single bulletin, or at least a continuum of bulletins that trade off precision and recall. While we would perhaps like to report the single *most likely* bulletin, as we discussed above this is not a well-defined quantity. Instead we choose a single sample from each Markov chain, namely the final sample after a fixed (48-hour) runtime, as a representative of the posterior. The assumption is that most events

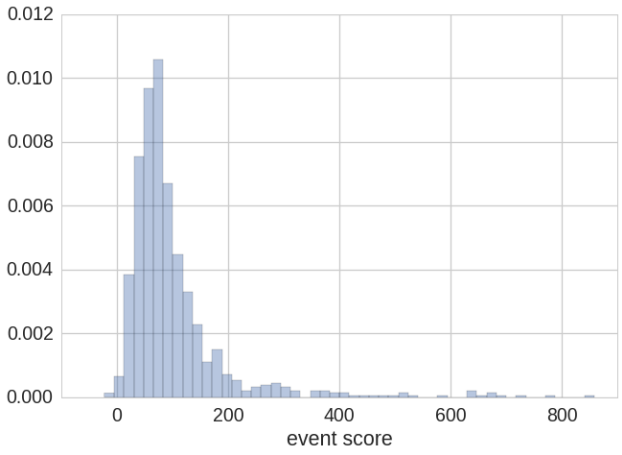


Figure 5.3: Distribution of event scores from an inference run on the two-week test period.

of note have probabilities close to 1 under the SIGVISA model, and so are never killed once born; thus the final sample from a chain contains most of the relevant events.[8]

Given a sample from the SIGVISA chain containing a set of events, we define the *score* of each event as the negative log-probability of accepting a death proposal for that event, considered with respect to a "dumb" prior-based birth proposal. This is essentially a likelihood-ratio test against an intelligently constructed "alternate" hypothesis in which some phase arrivals currently associated with the event may be deleted and others preserved as unassociated arrivals. The effect is to give high scores to events that coherently explain a large number of clearly-visible phase arrivals, and/or that correctly predict observed waveforms based on historical waveforms. By varying the score cutoff, we construct a continuum of bulletins trading off precision and recall. Figure 5.3 shows the distribution of scores from a single test run. Since scores correspond to log-probabilities, any event scoring above 5 has less than a 1% chance of being killed by a death proposal, thus validating the assumption above that most of the events we infer are quite "sticky", i.e., they are the peaks of relatively sharp local maxima.

Inference runs may become stuck in bad local maxima, so it is useful to combine the results of multiple runs by choosing the best events from each. We define a *merged bulletin* by the following simple procedure:

- When adding an event to the bulletin, check to see if a similar event (defined by a distance in time of < 50s and in location of $< 2^\circ$) already exists. If so, tag the events as equivalent.

[8]There is often some uncertainty about the location of a given event, so it's likely that location accuracy could be improved by using multiple samples to compute a posterior mean location for each event, although this is in itself not a well-defined quantity due to identity uncertainty (if event A is killed, and later event B is born at a similar time in a nearby location, should location samples from event A be combined with those from event B?).

- Once events from all runs are added, merge each equivalence class of duplicate events into a single event, with location given by its highest-scoring component, and score given by the sum of its component scores.

Thus the merged bulletin consists of the union of events from all individual bulletins, with duplicates removed by agglomerative clustering. The effect of summing event scores is to give a significant bonus to events found by multiple runs.[9]

Although we do not explore this in this thesis, it would be possible to use a merged bulletin containing events from several inference runs as an initialization for further rounds of inference, effectively allowing the merge procedure to serve as a crossover move in a larger inference procedure. This would enforce consistency on the merged bulletins, which in their naïve form can include mutually incompatible events explaining the same signals.

[9] We also considered using the max over component scores but found that the sum gave better results.

Chapter 6

Training

The SIGVISA model described in Chapter 4 can be viewed as describing a joint distribution over the past, present, and future of seismic events and observed signals. In practice, we are generally interested in the distribution over future activity *conditioned on* past observations. Obtaining this distribution is known as *training* the model. An ideal Bayesian reasoner would approach training by computing the true posterior distribution over all model parameters given available data, and use this as the prior when performing inference on future test data ("today's posterior is tomorrow's prior", Lindley, 1972). Our training procedure is guided by this ideal, with some concessions to practicality, such as performing approximate inference (MCMC) on the training data, and occasionally summarizing the resulting posterior distributions by point estimates.

Training SIGVISA requires estimating parameters for each component of the SIGVISA model. These include

- **Envelope parameters**: for the semiparametric Gaussian process models (Section 4.5) of each envelope parameter, for each phase at each station, we estimate hyperparameters $\ell, \sigma_n^2, \sigma_f^2$ and weight prior means and variances $\mathbf{b}$ and $\mathbf{B}$, as well as extracting historical conditioning data $(\mathbf{X}, \mathbf{y})$ from noisily observed signals.

- **Wavelet coefficients** (Section 4.6): we similarly estimate GP hyperparameters ℓ and σ_n^2 and extract historical conditioning data $(\mathbf{X}, \mathbf{y})$.

- **Background noise**: we learn priors over autoregressive noise parameters $\psi = (\mu, \sigma^2, \phi)$ for each station.

We also fit the event prior distributions from historical bulletins (Section 4.3), though this does not present any special difficulties as it involves only standard maximum-likelihood estimations. This chapter therefore focuses on learning the signal model components listed above.

To estimate these parameters, we use as data:

- A *training bulletin* $\mathbf{E} = (\mathbf{e}_i)_{i=1}^N$ enumerating the "ground truth" events that occur during the training period, including the location, depth, time, and magnitude of each event.[1]

- A set of *training signals* for each station in the network. Recall that the SIGVISA model treats the signals at each station as conditionally independent, given the event bulletin, so we can train models independently at each station. In this chapter we therefore suppress the station indices j from our notation and describe training a model for a generic station. We assume that the training signals are provided as a list of signals, $\mathbf{S} = (\mathbf{s}_i)_{i=1}^N$, each covering the arrival of one training event i.

The main difficulty in training is that most model parameters depend on latent quantities not observed in the data: it would be easy to fit the mapping between event location and, say, a particular P phase wavelet coefficient, if we were given the signals for the P arrival of each training event. In reality, of course, we observe noisy signals which may contain several overlapping phases, with additional uncertainty over the shapes and even arrival times of those phases. This uncertainty is especially problematic when learning models of wavelet coefficients, since even slightly misaligned arrival times for doublet events will lead to very different wavelet coefficients.[2]

The solution to this dilemma is the expectation maximization (EM) algorithm, which alternates between performing (approximate) inference to estimate the latent variables and fitting parameters to the inferred latent values (Dempster et al., 1977). The training procedure outlined in this section can be interpreted within the framework of an approximate EM algorithm.

6.1 Overall structure

The EM algorithm iterates the following two steps:

- **E step**: given estimated model parameters, perform (approximate) inference to obtain an (approximate) posterior on latent variables.

- **M step**: apply some fitting procedure to the approximate posterior to obtain new parameter estimates.

[1] In future work we anticipate relaxing the training bulletin to include uncertainty over event parameters, to accomodate the fact that bulletins such as the LEB typically contain significant error. In principle one could dispense with the bulletin entirely and attempt to induce all model parameters directly from observed signals, performing "unsupervised training" through an iterative process of inferring bulletins, training models, re-inferring bulletins, and so on. This would be interesting as a learning problem, but for practical purposes it is easier to rely on existing bulletins than to bootstrap the entire monitoring problem from data and first principles.

[2] The same would be true if we used a pure time-domain representation of modulation signals. A frequency domain representation would allow for some degree of translation invariance, while introducing other difficulties.

In our case the model parameters consist of GP kernel hyperparameters, parametric weight priors, and conditioning data as mentioned above, along with priors on autoregressive noise parameters. The latent variables of interest are the envelope shapes θ and wavelet coefficients $\mathbf{w}$ for each arriving phase, along with the background noise process $\mathbf{z}$. Thus our EM training process iterates the following two steps:

- **SIGVISA E step**: we run MCMC inference (Chapter 5) in a model with fixed events $\mathbf{E}$ and signals $\mathbf{S}$, to obtain approximate posterior distributions on the latent envelope shapes θ, wavelet coefficients $\mathbf{w}$, and noise processes $\mathbf{z}$.
- **SIGVISA M step**: we use these approximate posteriors to fit GP models and estimate noise process priors.

Because we observe a ground truth event bulletin, inference at training time is simpler than at test time, with no need for complex event birth and death proposals. We do still birth (and kill) unassociated arrivals to explain any signal spikes not associated with a modeled event phase. However, training time inference must account for signal correlations between nearby events, which we ignore at test time (Section 4.9.5). This requires additional calculations, discussed below.

The result of inference is a set of posterior samples for the envelope shapes and AR noise parameters governing each event signal $\mathbf{s}_i$, from which we can also extract posteriors on wavelet coefficients. The specifics of fitting GP models to these posteriors are discussed in Section 6.3, and the priors on AR parameters in Section 6.4.

Within the broad EM framework, the effectiveness of training depends heavily on initialization, as well as on the quality of inference during the E step. In Section 6.5 below we describe specific details regarding our implementation of parallel training, coarse to fine initializations, and heuristics for finding well-correlated signal alignments with which to initialize the MCMC inference.

6.2 Message passing for joint densities

As described in Section 4.9.5, during training it is necessary to account for dependence between nearby events when computing the joint signal likelihood. That is, we cannot factor the marginal signal likelihood (4.14), given by

$$p(\mathbf{S}|\mathbf{E},\theta,\psi) = \int p(\mathbf{S}|\mathbf{W},\theta,\psi)p(\mathbf{W}|\mathbf{E})d\mathbf{W},$$

into a product over event-specific signals $\prod_i p(\mathbf{s}_i|\mathbf{E},\theta,\psi)$, because the GP prior on wavelet coefficients $p(\mathbf{W}|\mathbf{E})$ (eq. (4.10)) introduces dependence between the coefficients of nearby events. This prevents us from naïvely employing the state-space likelihood calculation of Section 4.9.

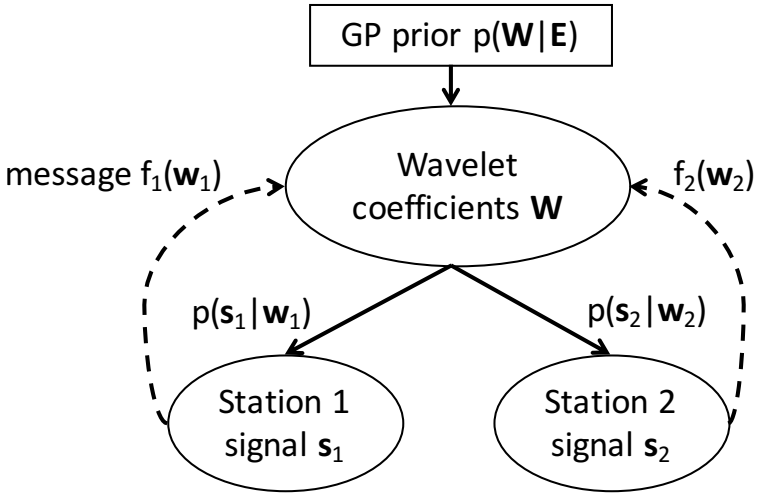


Figure 6.1: Joint distribution over signals for two events, illustrated as a Bayesian network.

However, with a bit more work we can still harness the efficiency of the state-space formulation. The conditional likelihood given wavelet coefficients $p(\mathbf{S}|\mathbf{W}, \theta, \psi)p(\mathbf{W}|\mathbf{E})$, does factor over training events,

$$\begin{aligned} p(\mathbf{S}|\mathbf{E}, \theta, \psi) &= \int p(\mathbf{S}|\mathbf{W}, \theta, \psi)p(\mathbf{W}|\mathbf{E})d\mathbf{W} \\ &= \int \left(\prod_{i=1}^{N} p(\mathbf{s}_i|\mathbf{w}_i, \theta, \psi) \right) p(\mathbf{W}|\mathbf{E})d\mathbf{W} \\ &= \int \left(\prod_{i=1}^{N} f_i(\mathbf{w}_i) \right) p(\mathbf{W}|\mathbf{E})d\mathbf{W}, \end{aligned} \tag{6.1}$$

and we view each factor f_i as a *message* passed from the observed signal towards the GP wavelet model (Figure 6.1), defined by treating the conditional likelihood $p(\mathbf{s}_i|\mathbf{w}_i, \theta, \psi)$ as a function of the wavelet coefficients (Koller and Friedman, 2009).

We approximate these messages by running Kalman filtering in a state space signal model (Section 3.5). The resulting messages $\tilde{f}_i$ are approximate because they are based on the filtering posterior, and therefore represent only a diagonal covariance $\tilde{\Sigma}_i$ that discards dependence between wavelet coefficients.[3] The assumption of diagonal messages is a form of *assumed density filtering* (Maybeck, 1982; Minka, 2001), which allows us to represent the posterior using a separate GP for each coefficient. An extension to model all wavelet coefficients jointly would be an interesting (though likely expensive) avenue of future work.

[3]We could gain a slightly better diagonal approximation by using the smoothing posterior instead, though this would increase complexity while still ignoring off-diagonal covariances.

Concretely, we run Kalman filtering on each signal $\mathbf{s}_i$, using a standard normal prior, $\mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This produces a sequence of filtered posteriors on individual wavelet coefficients c,

$$\tilde{p}(\mathbf{w}_{i,c}|\mathbf{s}_i) \sim \mathcal{N}(\tilde{\mu}_{i,c}, \tilde{\sigma}^2_{i,c}),$$

as well as an (exact) marginal likelihood

$$p(\mathbf{s}_i) = \int_{-\infty}^{\infty} p(\mathbf{s}_i|\mathbf{w}_i)p(\mathbf{w}_i)d\mathbf{w}_i.$$

Considered jointly over coefficients, the filtered posterior $\mathcal{N}\left(\tilde{\mu}_i, \tilde{\mathbf{\Sigma}}_i\right)$ is a diagonal approximation to the true posterior $p(\mathbf{w}_i|\mathbf{s}_i) = \mathcal{N}\left(\mu_i, \mathbf{\Sigma}_i\right)$. For completeness we first derive messages in terms of the true posterior, then specialize to the filtered approximation. Recalling Bayes' rule, eq. (3.1),

$$p(\mathbf{w}_i|\mathbf{s}_i) = \frac{p(\mathbf{s}_i|\mathbf{w}_i)p(\mathbf{w}_i)}{p(\mathbf{s}_i)},$$

we can rearrange to express the message $f_i(\mathbf{w}_i) = p(\mathbf{s}_i|\mathbf{w}_i)$ as the posterior divided by the prior, scaled by the marginal likelihood:

$$f_i(\mathbf{w}) = p(\mathbf{s}_i) \cdot \frac{p(\mathbf{w}_i|\mathbf{s}_i)}{p(\mathbf{w}_i)}.$$

Using eq. (A.7) we see that the messages are unnormalized Gaussian densities,

$$\begin{aligned} f_i(\mathbf{w}_i) &= p(\mathbf{s}_i) \cdot \frac{\mathcal{N}\left(\mathbf{w}_i; \mu_i, \Sigma_i\right)}{\mathcal{N}(\mathbf{w}_i; \mathbf{0}, \mathbf{I})} \\ &= p(\mathbf{s}_i) \cdot \frac{1}{|\mathbf{I} - \mathbf{\Sigma}_i|} \frac{1}{\mathcal{N}\left(\mu_i; \mathbf{0}, \mathbf{I} + \mathbf{\Sigma}_i\right)} \cdot \mathcal{N}\left(\mathbf{w}_i; \left(\mathbf{\Sigma}_i^{-1} - \mathbf{I}\right)^{-1} \mathbf{\Sigma}_i^{-1}\mu_i, \left(\mathbf{\Sigma}_i^{-1} - \mathbf{I}\right)^{-1}\right). \end{aligned}$$

Plugging in the diagonal filtered posterior $\tilde{\mu}_i, \tilde{\mathbf{\Sigma}}_i$ yields approximate messages that factor over wavelet coefficients,

$$\begin{aligned} \tilde{f}_i(\mathbf{w}_i) &= p(\mathbf{s}_i) \prod_c \left[\frac{1}{(1 - \tilde{\sigma}^2_{i,c})\mathcal{N}\left(\tilde{\mu}_{i,c}; 0, 1 + \tilde{\sigma}^2_{i,c}\right)} \cdot \mathcal{N}\left(w_{i,c}; \frac{\tilde{\mu}_{i,c}}{1 - \tilde{\sigma}^2_{i,c}}, \frac{\tilde{\sigma}^2_{i,c}}{1 - \tilde{\sigma}^2_{i,c}}\right) \right] \\ &= p(\mathbf{s}_i) \prod_c \left[\frac{1}{Z_{i,c}} \mathcal{N}(w_{i,c}; \nu_{i,c}, \xi_{i,c}) \right], \end{aligned} \tag{6.2}$$

where the message factor for each wavelet coefficient is itself an unnormalized Gaussian density with mean $\nu_{i,c} = \frac{\tilde{\mu}_{i,c}}{1-\tilde{\sigma}^2_{i,c}}$, variance $\xi_{i,c} = \frac{\tilde{\sigma}^2_{i,c}}{1-\tilde{\sigma}^2_{i,c}}$, and normalizing constant $Z_{i,c} = \frac{1}{(1-\tilde{\sigma}^2_{i,c})\mathcal{N}\left(\tilde{\mu}_{i,c}; 0, 1+\tilde{\sigma}^2_{i,c}\right)}$. Having computed the filtered messages, it remains to combine them

with the GP prior to evaluate the joint density (6.1). Recall that the prior $p(\mathbf{W}|\mathbf{E})$ is joint over events i, but factors over wavelet coefficients c, so we have

$$\begin{aligned} p(\mathbf{S}|\mathbf{E},\theta,\psi) &= \int \left(\prod_{i=1}^{N} f_i(\mathbf{w}_i)\right) \prod_c p(\mathbf{w}_c|\mathbf{E}) d\mathbf{W} \\ &\approx \int \left(\prod_{i=1}^{N} p(\mathbf{s}_i) \prod_c \left[\frac{1}{Z_{i,c}} \mathcal{N}(w_{i,c}; \nu_{i,c}, \xi_{i,c})\right]\right) \prod_c p(\mathbf{w}_c|\mathbf{E}) d\mathbf{W} \\ &= \left(\prod_{i=1}^{N} p(\mathbf{s}_i) \frac{1}{\prod_c Z_{i,c}}\right) \prod_c \left[\int \mathcal{N}(\mathbf{w}_c; \bar{\nu}_c, \xi_c) \mathcal{N}\left(\mathbf{w}_c; \bar{f}_c(\mathbf{E}), \Sigma_{f,c}(\mathbf{E})\right) d\mathbf{w}_c\right] \end{aligned}$$

in which $\bar{\nu}_c, \xi_c$ represent a diagonal Gaussian density collecting the filtered messages for coefficient c across all events, and $\bar{f}_c(\mathbf{E}), \Sigma_{f,c}(\mathbf{E})$ are the prior GP mean and covariance (4.10). By eq. (A.6), the quantity inside the integral is an unnormalized Gaussian density in $\mathbf{w}_c$, so that after integrating we are left with only the normalizing constant

$$\int \mathcal{N}(\mathbf{w}_c; \bar{\nu}_c, \xi_c) \mathcal{N}\left(\mathbf{w}_c; \bar{f}_c(\mathbf{E}), \Sigma_{f,c}(\mathbf{E})\right) d\mathbf{w}_c = \mathcal{N}\left(\bar{\nu}_c; \bar{f}_c(\mathbf{E}), \Sigma_{f,c}(\mathbf{E}) + \xi_c\right).$$

The effect of this derivation is to evaluate each GP prior at the message mean $\bar{\nu}_c$, with heteroskedastic (independent but different for each event) Gaussian noise given by the message variances ξ_c. Putting this all together, we efficiently evaluate the joint signal density (6.1) by computing the filtered posterior $\tilde{\mu}_i, \tilde{\Sigma}_i$ and marginal likelihood $p(\mathbf{s}_i)$ for each signal by Kalman filtering in the state space model of Section 4.9, computing messages (6.2) for each wavelet coefficient c, and treating these messages as describing Gaussian observation noise in evaluating the GP prior, yielding the final density

$$p(\mathbf{S}|\mathbf{E},\theta,\psi) \approx \left(\prod_{i=1}^{N} p(\mathbf{s}_i) \frac{1}{\prod_c Z_{i,c}}\right) \prod_c \mathcal{N}\left(\bar{\nu}_c; \bar{f}_c(\mathbf{E}), \Sigma_{f,c}(\mathbf{E}) + \xi_c\right). \tag{6.3}$$

Although this is still an approximation to the true joint Gaussian density, it is much more faithful to the model than the implicit approximation made in Section 4.9 by assuming independence across events.

6.3 Training GP models

Running MCMC with the joint signal model described in the previous section yields a set of samples from the posterior $p(\theta, \psi|\mathbf{S}, \mathbf{E})$ over envelope shape parameters and AR noise parameters. Note that we do not get explicit samples of the wavelet coefficients, which are marginalized out. This section describes how we fit GP models for envelope shape parameters and wavelet coefficients. Specifically, for each station and phase, and for each shape variable $(\tau, \rho, \alpha, \gamma, \beta)$ and wavelet coefficient w_c, we must fit the GP models described in Sections 4.5

and 4.6, requiring us to specify kernel hyperparameters $\ell, \sigma_n^2, \sigma_f^2$, training inputs $\mathbf{X}, \mathbf{y}$, and for semiparametric models, the parameter prior mean $\mathbf{b}$ and covariance $\mathbf{B}$. The input locations $\mathbf{X}$ are shared across all models and are simply given by the training events $\mathbf{E}$.

We select hyperparameters for all GP models through gradient-based optimization of a penalized marginal likelihood (3.16). Using $\mathbf{y}$ to represent a generic latent variable, the marginal likelihood is just the evaluation of a Gaussian prior density

$$p(\mathbf{y}|\mathbf{X}; \ell, \sigma_n^2, \sigma_f^2) = \mathcal{N}\left(\mathbf{y}; \mu_\mathbf{y}, \mathbf{\Sigma_y}\right)$$

at the observed values $\mathbf{y}$. Unlike the standard GP setting, where $\mathbf{y}$ is observed directly; here we have access only to posterior samples of $\mathbf{y}$ from the E step. We could use a point estimate, for example the posterior mean, to evaluate the marginal likelihood, but this ignores potentially important posterior uncertainty. For example, a low-amplitude phase may not have a well-identified arrival time, so training on any point estimate of the arrival time would cause the model to become falsely confident.

To account for posterior uncertainty, we augment the marginal likelihood to (approximately) marginalize over the latent variable $\mathbf{y}$. Suppose that after performing inference under a Gaussian prior we obtain a Gaussian posterior (this will not in general be the case). We can then divide the posterior by the prior to yield a Gaussian *message*

$$f_y(\mathbf{y}) = p(\mathbf{S}, |\mathbf{y}, \mathbf{E}) \propto \mathcal{N}(\mathbf{y}; \mathbf{a}, \mathbf{A}),$$

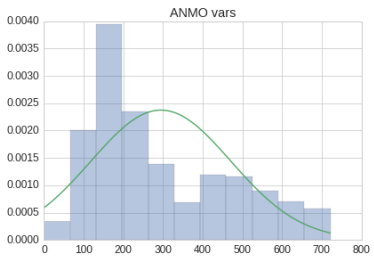
as in the joint density calculations of Section 6.2 above. The augmented marginal likelihood $\mathcal{L}^*$ is then given by marginalizing over $\mathbf{y}$,

$$\begin{aligned}\mathcal{L}^*\left(\ell, \sigma_n^2, \sigma_f^2\right) &= \int f_y(\mathbf{y}) p(\mathbf{y}|\mathbf{X}; \ell, \sigma_n^2, \sigma_f^2) d\mathbf{y} \\ &\propto \mathcal{N}(\mathbf{y}; \mathbf{a}, \mathbf{A}) \mathcal{N}\left(\mathbf{y}; \mu_\mathbf{y}, \mathbf{\Sigma_y}\right) d\mathbf{y} \\ &\propto \mathcal{N}\left(\mathbf{a}; \mu_\mathbf{y}, \mathbf{\Sigma_y} + \mathbf{A}\right), \end{aligned} \tag{6.4}$$

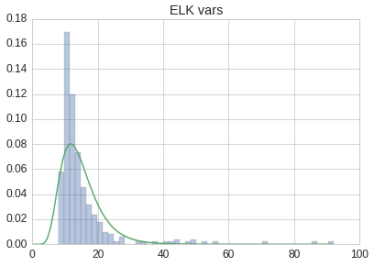
and corresponds to evaluating the GP prior at the message mean $\mathbf{a}$, with added variance $\mathbf{A}$. Given a fixed set of messages from an inference run, we select hyperparameters for each latent variable model by maximizing $\mathcal{L}^*$, penalized by the appropriate hyperprior from table 4.4.

Of course, the true posterior on latent variables is not Gaussian. For the envelope shape variables, we use the empirical means and variances of the MCMC samples for each event phase to construct a (diagonal) Gaussian approximation to the true posterior. For the wavelet coefficients, we condition on the envelope shapes from the most likely posterior sample and extract Gaussian posteriors through Kalman filtering as in Section 6.2 above.

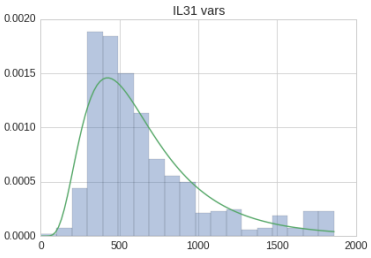
Finally, for the envelope shape variables we must also identify a Gaussian mean $\mathbf{b}$ and covariance $\mathbf{B}$ over the weights for the parametric GP component, governing, e.g., the generalizable relationship between amplitude and event–station distance. Here we take literally the principle that "today's posterior is tomorrow's prior", and take the test-time prior to be the posterior (3.13) obtained analytically by conditioning on training data.



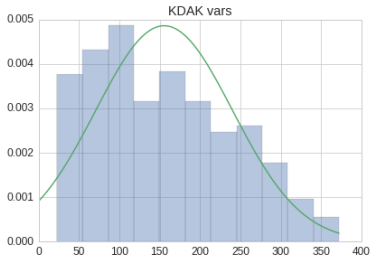
(a) ANMO (truncated Gaussian).



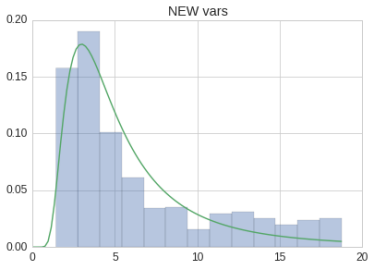
(b) ELK (log-normal).



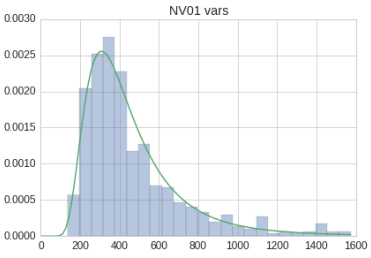
(c) ILAR (log-normal).



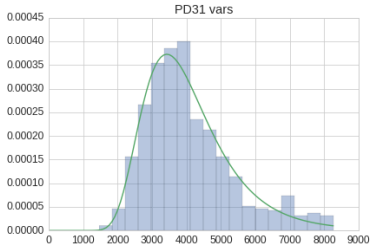
(d) KDAK (truncated Gaussian).



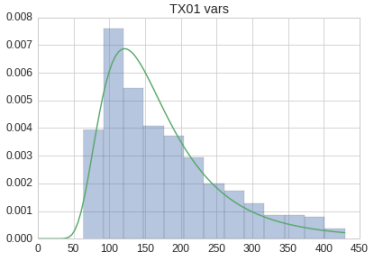
(e) NEW (inverse Gamma).



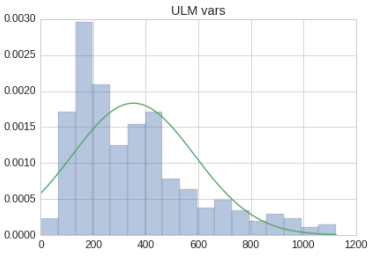
(f) NVAR (inverse Gamma).



(g) PDAR (inverse Gamma).



(h) TXAR (inverse Gamma).



(i) ULM (truncated Gaussian).

Figure 6.2: Posterior distributions of noise model variances at several stations, showing histograms of samples $(\sigma^2)^*$ and corresponding model fits.

6.4 Training station noise models

We also learn priors on the autoregressive noise parameters $\psi = (\mu, \sigma^2, \phi)$, i.e., mean, variance, and process coefficients at each station. For each training signal $\mathbf{s}_i$ we extract the noise parameters ψ_i^* from the highest-likelihood MCMC posterior sample; the collected samples $\psi^* = (\psi_i^*)_{i=1}^N$ represent estimates of the noise process over many different time periods.

For the noise mean μ and process coefficients ϕ, we fit Gaussian priors (multivariate in the latter case) to these samples, with means and (co)variances given by the empirical means and (co)variances of μ^* and ϕ^*. Since we work with band-pass filtered waveforms, we could perhaps get away with just fixing $\mu = 0$; in practice, learning the mean from data has a

similar effect.

The process variance σ^2 determines the overall "noise level" at each station. It is important to model it well; a bad prior may cause inference to get stuck in states that use unassociated arrivals to explain what is genuinely noise, or vice versa. Depending on the station, we use one of three models for σ^2: a truncated Gaussian,

$$p_{TG}(\sigma^2) \propto \mathbb{I}[\sigma^2 > 0] \exp\left(-\frac{(\sigma^2 - \alpha)^2}{2\beta}\right),$$

a log-normal distribution,

$$p_{LN}(\sigma^2) \propto \exp\left(-\frac{(\log \sigma^2 - \alpha)^2}{2\beta}\right),$$

or an inverse Gamma distribution,

$$p_{IG}(\sigma^2) \propto \left(\sigma^2\right)^{-\alpha-1} \exp\left(\frac{-\beta}{\sigma^2}\right).$$

We fit the parameters α, β for each model by maximum likelihood, and then apply maximum likelihood again to select the model itself, essentially fitting a meta-level model that incorporates a discrete choice of distribution family p_{TG}, p_{LN}, or p_{IG}. Figure 6.2 shows examples of the model fits for several IMS stations.

6.5 Large-scale training, initialization, and coarse-to-fine fitting

For large or even moderately sized data sets, joint inference as described in Section 6.2 becomes expensive and often intractable to perform on a single machine. This section describes our approach for effective training in practice, using parallelization, heuristic initializations, and a hierarchy of coarse model structures to speed up the initial training steps.

Because the time to evaluate a Gaussian density scales cubically with dimension, partitioning a large GP model into many smaller models is computationally advantageous. We do this by partitioning the training data into local clusters via k-means clustering. For the western US dataset evaluated in Chapter 7, we partition the training events into 38 clusters, shown in Figure 6.3. The cluster sizes are highly nonuniform; the median cluster contains 18 events, but the smallest clusters contain only 1 event while the largest cluster (corresponding to the Powder River Basin mining region) contains 137 events. We limit the training data to the 50 highest-magnitude events in each cluster. We train an independent GP models for each cluster; this is equivalent to training a single GP with a covariance function that imposes independence between points in different regions. In this dataset the clusters are often (though not always) well separated, so this independence assumption is perhaps justifiable;

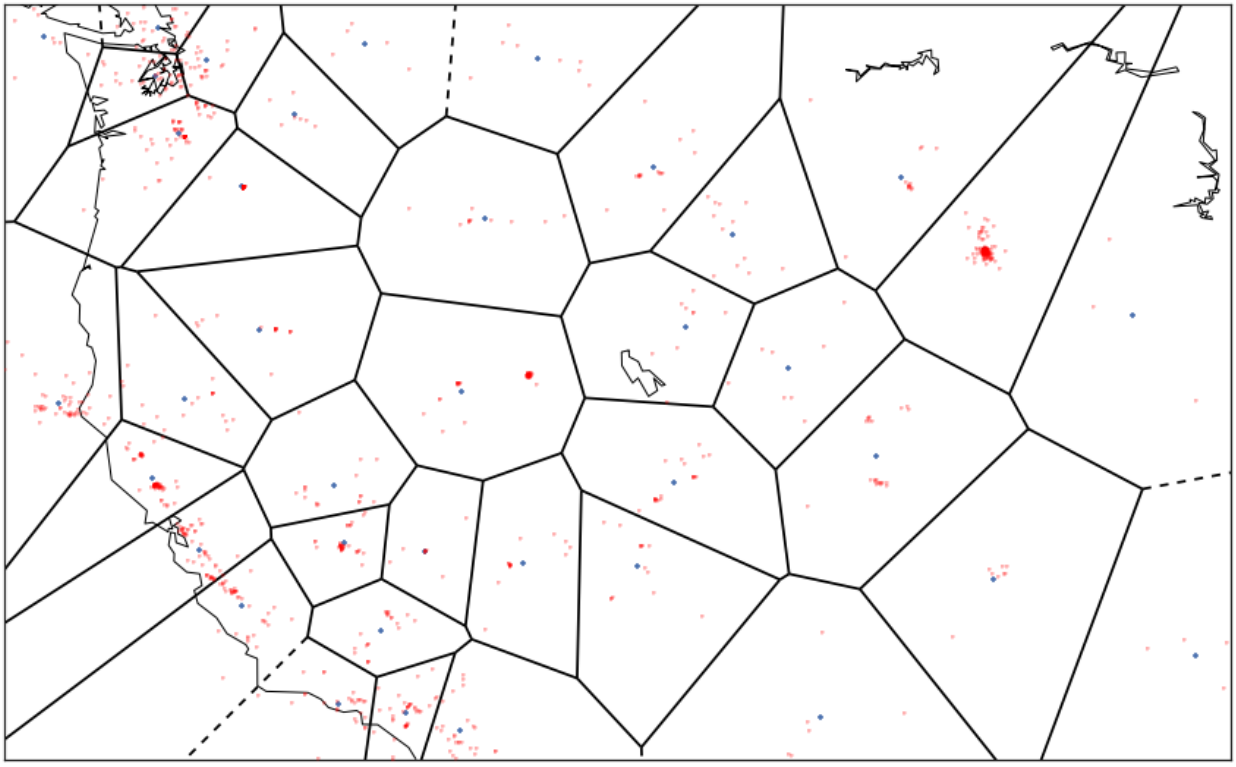


Figure 6.3: Partition into 38 regions found by k-means clustering, with clusters centers marked in blue.

if desired it could be relaxed, at some additional cost, using a Gaussian process random field (Moore and Russell, 2015).

Using a partitioned model allows us to perform training in parallel. Inference in the E step can be performed separately for each station and for each cluster of events; using a 12-station network with 38 clusters this enables up to a 456-fold speedup from parallelism. The resulting messages are then collected for a single joint M step, allowing the model to learn global parametric relationships such as distance-dependence from the entire training set.

To speed up the training, we perform initial rounds of inference using a *coarse* model. The coarse model differs from the full SIGVISA model in that it uses no GPs, just parametric linear-in-features models (eq. (3.6)) of envelope shape parameters, and i.i.d. Gaussian noise for modulation signals. It also models the signal envelope $\mathbf{v}$ instead of the raw waveform $\mathbf{s}$; this allows us to downsample signals from 10Hz to 1Hz, dramatically reducing the quantity of data that must be processed. These changes speed up inference by several orders of magnitude. Envelope shapes inferred in the coarse model are not guaranteed to be coherent for nearby doublet events, but can be used to learn initial parametric models of envelope shapes and station background noise.

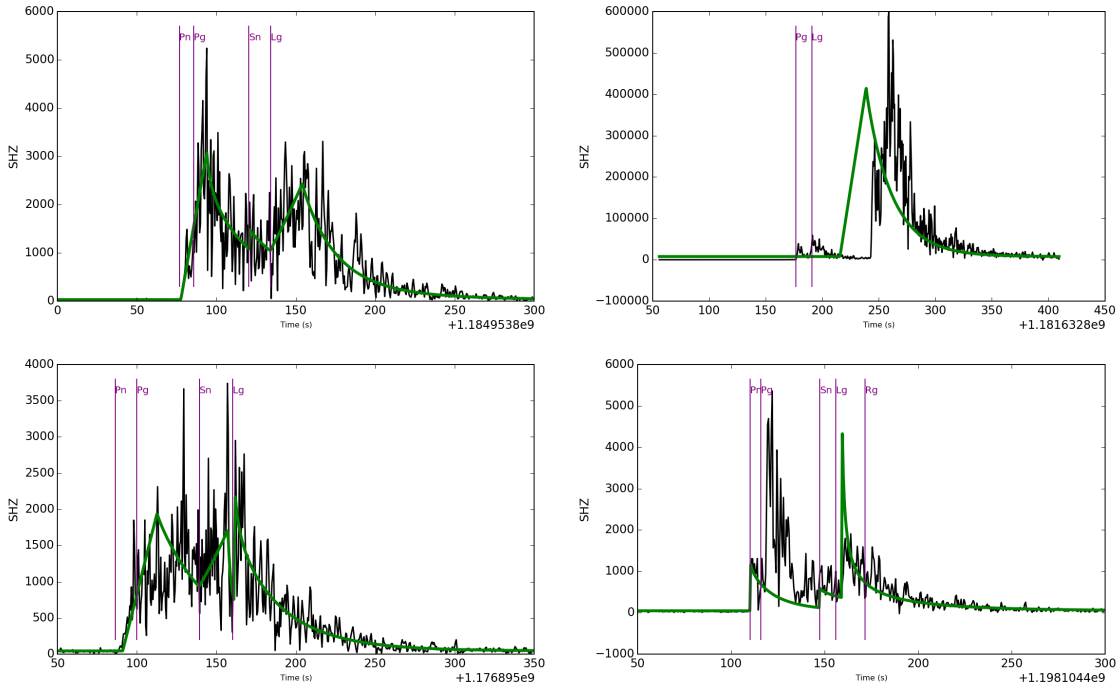


Figure 6.4: Examples of coarse (envelope) model fits manually discarded during first training iteration. Clockwise from top left: missed Pn, distracted by clutter, missed Pg, missed Pg.

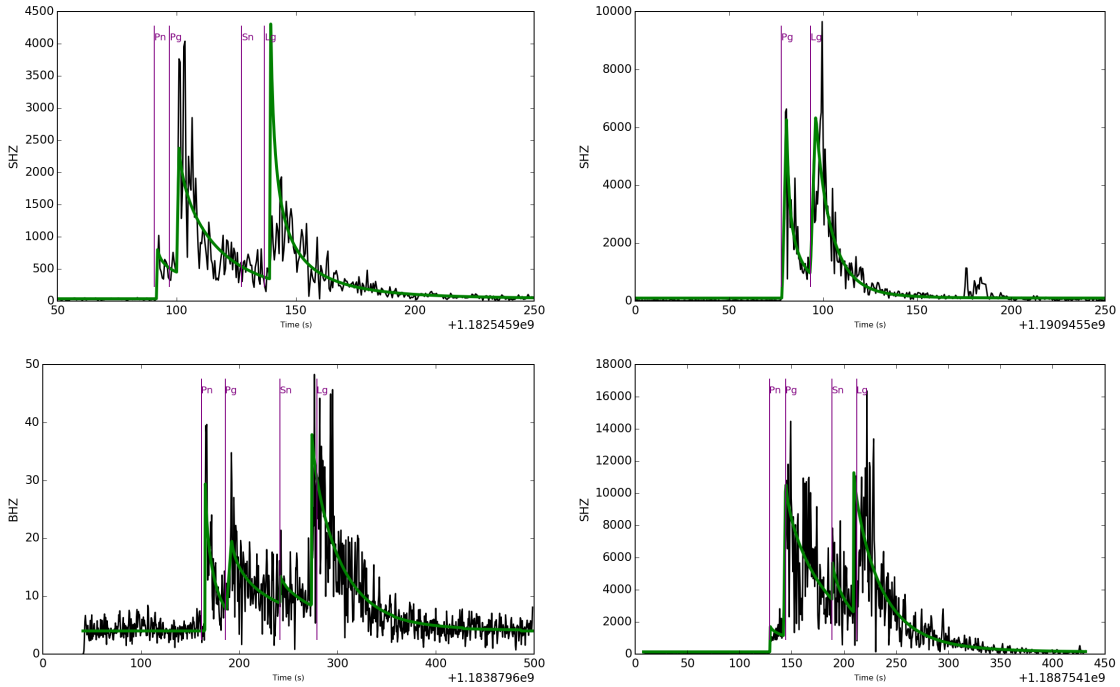
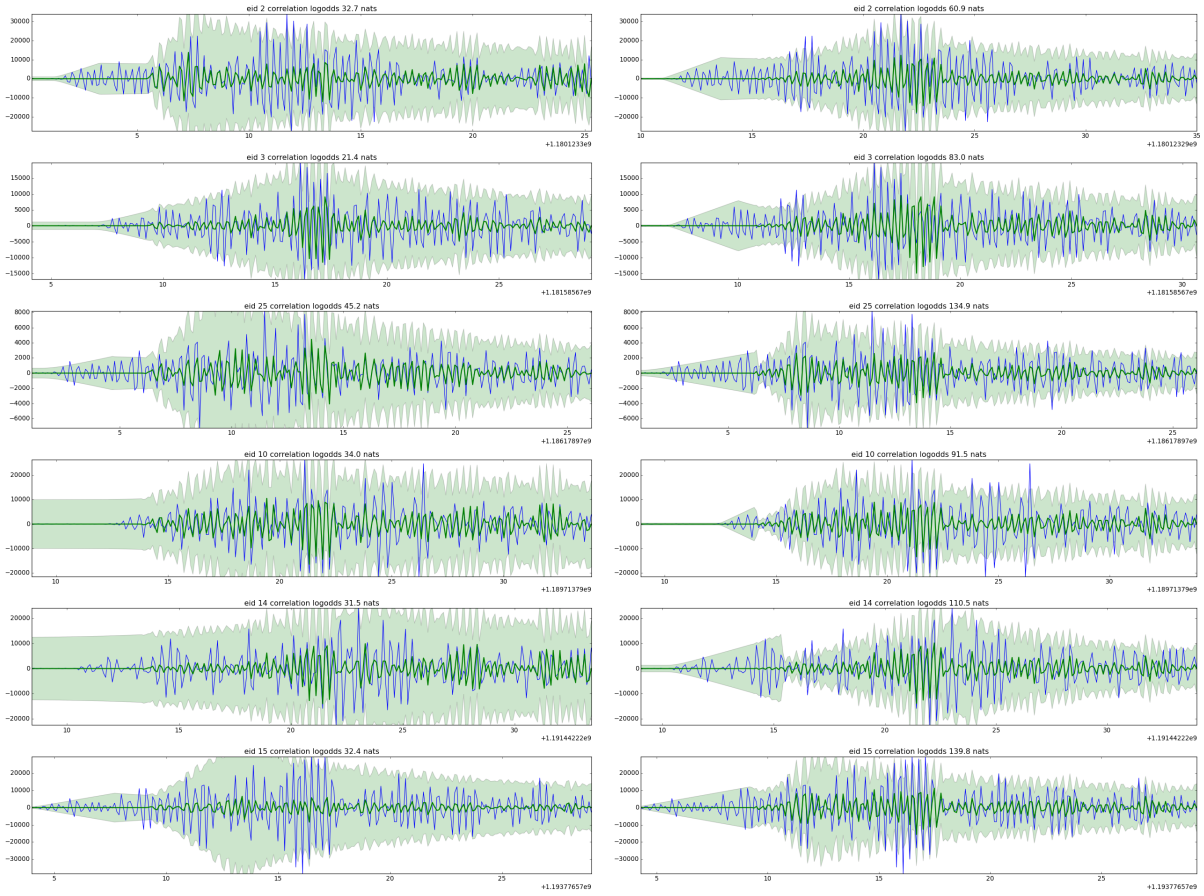


Figure 6.5: Examples of coarse model fits labeled as acceptable during first training iteration.



(a) Heuristic initialization. (b) After 70 epochs of joint MCMC fitting.

Figure 6.6: Observed signals (blue) and leave-one-out GP model predictions (green) for Pg arrivals of six Black Thunder Mine events recorded at the Pinedale array (PDAR). Predictions are conditioned on 50 events in the local cluster. Correlation log odds listed for each event indicate the increase in signal likelihood from the GP prediction relative to an i.i.d. Gaussian (nonrepeatable) modulation signal. MCMC fitting adapts the envelope shapes to fit the aligned signals, leading to higher log odds ratios.

We also use several other heuristics to encourage the training process to find good explanations. Our overall training procedure follows these steps:

1. We run an initial E step using a coarse model on a subset of events chosen to yield clear arrivals. Each event fit is manually examined, and we discard those that appear implausible. Figures 6.4 and 6.5 show examples of fits that are discarded and retained, respectively, at this stage. The retained fits are used in an M step to train linear models of envelope parameters given event features (table 4.4).

2. We then iterate one or more cycles of E and M steps on the full training set, using a coarse model. Each cycle requires roughly two days on a single quad-core machine. Rather than prune bad fits by hand, we use a heuristic: before each M step we discard any outlier fits that are > 2.5 nats more likely under a dummy constant model than under the linear models we fit.

3. We next perform a single E step using the full joint model, parallelized over several dozen machines using an independent submodel for each combination of station and spatial region. To initialize inference for each submodel, we

   a) First run inference in a *partially coarse* model that models envelopes instead of raw waveforms, and treats modulation signals as i.i.d. Gaussian, but does impose GP priors on envelope shapes. This encourages the fits within each cluster to become spatially coherent.

   b) Next, heuristically discard outlier fits. We concatenate the shape parameters of all phases into a vector for each event, map these vectors into $\mathbb{R}^5$ using a random Gaussian projection, fit a multivariate Gaussian to the projected points, and discard the 20% of events with largest Mahalanobis distance (Mahalanobis, 1936) from the mean. These are assumed to be events for which we did not find envelope shapes coherent with their neighbors.

   c) Finally, we search for a heuristic joint alignment of all events within a cluster, so as to maximize correlation between nearby events. For each phase, we perform coordinate ascent on a surrogate objective ω measuring the correlation between the aligned signal for each event $\mathbf{s}_i(\tau_i)$ and the *mean* signal from all other neighboring events $N(i)$ within 25km of event i, $\mathbf{s}_{\setminus i} = \frac{1}{N}\sum_{j \in N(i)} \mathbf{s}_j$. We treat this heuristic objective,

   $$\omega(\tau) = \sum_i \exp\left(\frac{\mathbf{s}_i^T \mathbf{s}_{\setminus i}}{\|\mathbf{s}_i\|\|\mathbf{s}_{\setminus i}\|}\right),$$

   as a function of the arrival time τ_i for each event, and repeatedly adjust the arrival time of each event phase to maximize correlation with the mean signal from its neighbors, under the constraint that arrival times change by no more than 2 seconds from the (hopefully coherent) initial fits obtained previously via inference

using a GP travel-time prior. Like the true model, this heuristic correlation objective is quite nonconvex; we choose the best aligned arrival times after performing several hundred random restarts, which typically takes no more than a minute.

Using this initialization, we run MCMC using the joint density (6.3) to tune the joint envelope shapes and alignments across all events in the cluster. For semiparametric GP models, the prior on the parametric components is taken from the earlier M step on the coarse model. Since (6.3) involves evaluating a GP likelihood inside each MCMC step, there is little extra cost to allowing inference to adapt the GP hyperparameters $\ell, \sigma_n^2, \sigma_f^2$ online, treating them as latent variables governed by hyperpriors, so that each cluster learns to align waveforms according to the appropriate local correlation lengthscale. Figure 6.6 shows examples of envelope fits and predicted waveforms from the heuristic initialization, and from the final step of MCMC tuning, showing that running MCMC yields predictive models significantly better than those from the heuristic initialization. We allow up to three CPU-days for inference on each event cluster at each station; for larger clusters this corresponds to around 70-200 MCMC epochs.

4. Given aligned envelopes at each cluster, we run a final M step, extracting messages and training GP models as described above (Section 6.3), and fitting separate GP hyperparameters for each cluster by optimizing a factored version of the augmented marginal likelihood (6.4). We could instead use the hyperparameters obtained by MCMC during the E step, but running an explicit M step allows us to jointly learn hyperparameters along with new priors on the parametric model components.

The result of this fitting process is a Gaussian process model for each envelope shape parameter and each wavelet coefficient, at each station, for each modeled phase; for the experiments in this thesis this corresponded to 12147 trained GP models. Given the large number of models fit, as well as the nature of latent variable modeling, in which the "training data" for each model are themselves the result of model-dependent approximate inference procedures, it is difficult to manually examine and validate the fit of each individual model. We approached debugging by visualizing subsamples of envelope shape fits, measuring the predictive likelihoods of heldout signals, and ultimately evaluating the quality of end-to-end inference using the trained models, as described in Chapter 7.

We found the early coarse fitting steps to play a crucial role in forming informative priors on envelope shapes. The structural assumptions described in Chapter 4 do not fully constrain the model; they could equally well represent a world in which amplitudes increase with event-station distance, or in which observed amplitudes are subject to extreme variance and essentially unpredictable from the event parameters, as the actual world we live in, in which amplitudes decay with distance and are typically predictable to within an order of magnitude. By fitting simple parametric models to coarse signal representations, we are able to capture this information early in the training process; this helps the later, more expensive joint fitting steps avoid falling into spurious local maxima such as those shown in Figure 6.4.

Finding a coherent joint alignment of all training signals is also crucial to the training process. If a deity were to provide us with precise arrival times for each phase in the training set, it would be relatively straightforward to fit the remaining shape parameters, extract wavelet coefficients, and estimate spatial correlation lengthscales. On the other hand, if even a small fraction of signals from an event cluster are misaligned, the model may be forced to assume a low level of spatial correlation in that cluster, reducing the incentive to correctly align the remaining signals and causing fitting to collapse to a degenerate solution. We attempted to prevent this by using informative priors to enforce a minimal degree of spatial correlation (Table 4.4), but also invested quite a bit of effort in finding well-aligned initializations, using heuristics to discard outlier fits as well as performing hundreds of random restarts on the surrogate objective ω. Despite this effort, there is still no guarantee that the fits found by our training procedure are optimal or even near-optimal, and there is likely room for substantial further improvements.

Chapter 7

Evaluation: Western US

In this chapter we compare the performance of SIGVISA to several existing monitoring systems on the task of monitoring within a restricted region, namely the western United States. Focusing on a specific region significantly reduces the computational burden, because the system need only consider signals from a subset of IMS stations. On this regional monitoring task, SIGVISA demonstrates significant improvements in the number of detected events (recall) and in mean location accuracy. It even detects a large number of events missed by local and regional networks, despite using only data from the IMS network. We are hopeful that these promising results will generalize to improvements in global monitoring, which we defer to future work.

7.1 Dataset

We consider the task of monitoring seismic events in the western United States, using signal data from the IMS network. Specifically, we consider the region bounded in latitude between 33°N and 49°N, and in longitude between 126°W and 100°W. We focus on the western United States because it contains both significant natural seismicity and regular mining explosions — most notably from Wyoming's Powder River Basin, which contains the world's two largest coal mines, Black Thunder Mine and North Antelope Rochelle Mine (US Energy Information Administration, 2016).

We focus in particular on the time period immediately following the magnitude 6.0 earthquake near Wells, NV, on February 21, 2008, which generated a large number of aftershocks. By fortuitous coincidence, the transportable US Array,[1] consisting of 400 seismometers in a regular grid spaced at approximately 70km, was deployed at the time in the surrounding area, providing an unusually close record of the aftershock sequence. The western US is also relatively well covered by regional stations operated by the National Earthquake Information Center (NEIC), which are not part of the IMS but provide a more precise and sensitive picture of regional seismic activity. These additional sensors allow us to form a reference

[1] http://www.usarray.org/researchers/obs/transportable

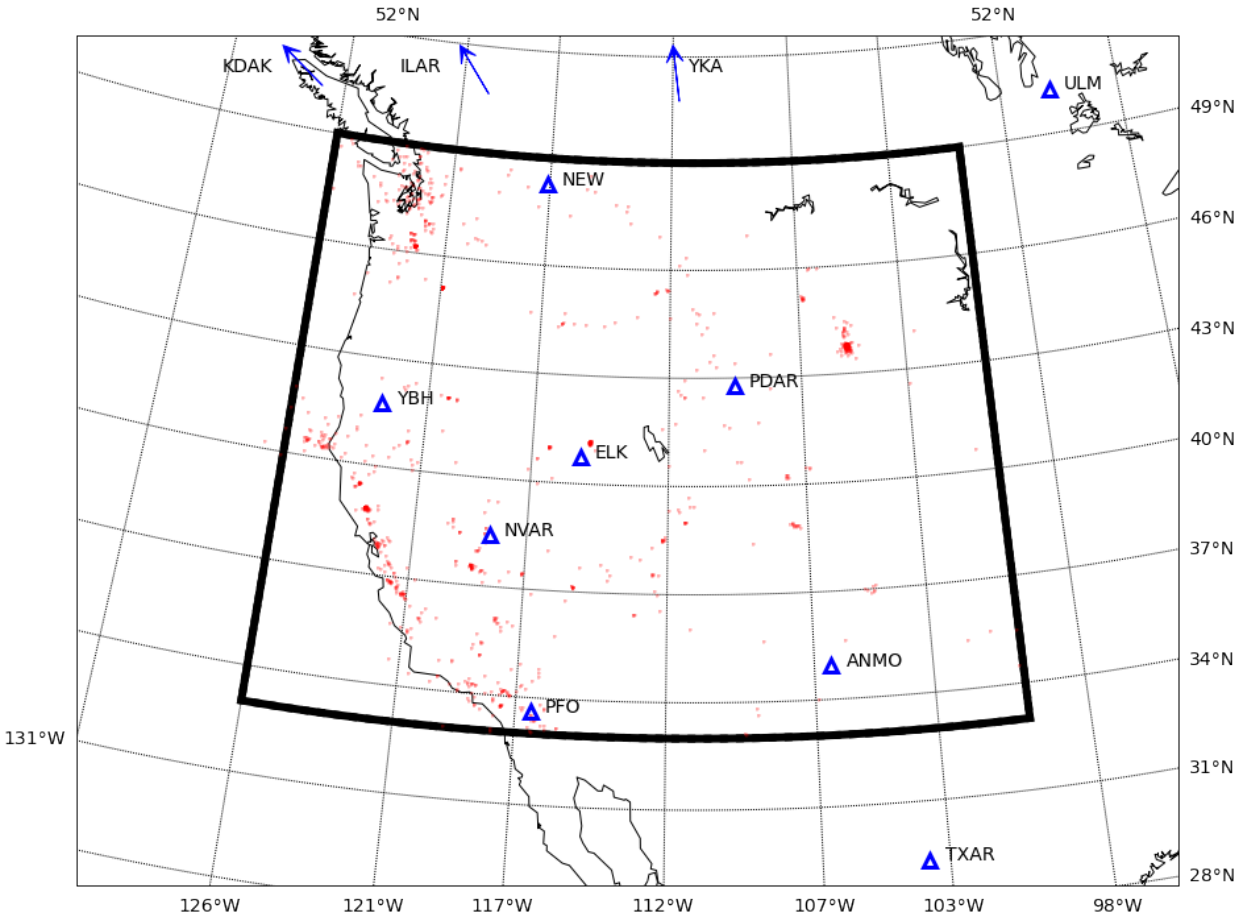


Figure 7.1: Training events from the western US dataset, with region of interest outlined in black. Triangles indicate IMS stations (Table 7.1); note that stations KDAK, ILAR, and YKA are above the north edge of the map.

bulletin, described below, against which to evaluate systems that use only data from the much sparser IMS global network.

As is typical in machine learning, we divide the available data into a training set, a validation set (used during development for model selection and tuning), and a test set on which final results are reported. We use the following split:

- **Training**: one year of historical data, from January 1, 2007, to Dec 31, 2007, as well as the first six hours following the Wells mainshock (February 21, 2008 from 14:16 UTC to 20:16 UTC).
- **Validation**: hours six through twelve following the Wells mainshock: 20:16 UTC on February 21 to 02:16 UTC on February 22.

Code	Location	Lon (°W)	Lat (°N)	Type
ANMO	Albuquerque, NM	106.46	34.95	3C
ELK	Elko, NV	115.24	40.74	3C
ILAR	Eielson, AK	146.89	64.77	Array
KDAK	Kodiak Island, AK	152.58	57.78	3C
NEW	Newport, WA	117.12	48.26	3C
NVAR	Mina, NV	118.30	38.43	Array
PDAR	Pinedale, WY	109.56	42.77	Array
PFO	Pinon Flat, CA	116.45	33.61	3C
TXAR	Lajitas, TX	103.67	29.33	Array
ULM	Lac du Bonnet, Manitoba, Canada	95.87	50.25	3C
YBH	Yreka, CA	112.71	41.73	3C
YKA	Yellowknife, NWT, Canada	114.61	62.49	Array

Table 7.1: IMS stations used by SIGVISA to monitor the western US.

- **Test**: two weeks beginning twelve hours after the Wells mainshock, from 02:16 UTC on February 22 to 02:16 UTC on March 7, 2008.

The data consist of continuous signals from twelve IMS stations, as well as reference hypocenter locations used during training and for evaluation during the test period.

Signal data We used signals from twelve IMS stations located in North America (Table 7.1). From each station we extract a single continuous waveform: for three-component stations we use the vertical component; at array stations we use the vertical component at the reference station.[2] All signals are bandpass filtered (to either 0.8-4.5Hz or 2.0-4.5Hz, see below) and then downsampled to 10 Hz.

Reference hypocenters: We use the bulletin of the International Seismological Centre (ISC), which aggregates regional network data from sources including the National Earthquake Information Center (NEIC) and the US Array Network Facility (ANF). For ISC events with multiple authors and no prime hypocenter, we chose the origin location with the smallest error ellipse; if error ellipses were not available, we preferentially used the ANF and NEIC origins, in that order. Figure 7.1 shows the ISC events for the training period. During the test period, we augment the ISC bulletin of 102 events with an additional 944 events from analysis of the Wells aftershock sequence provided by the Nevada Bureau of Mines and Geology at the University of Nevada, Reno (UNR) (Smith et al., 2011). The UNR events are formed from the US Array as well as 27 temporary instruments deployed approximately one week after the main shock, and relocated with HypoDD; they give our reference bulletin a clearer picture of the "ground truth" for the Wells sequence, which represents a sizable

[2]Extending the SIGVISA model to three-component and array signals is an important subject for future work. It is worth noting that our current results are already competitive with existing systems that do have access to azimuth and slowness estimates from these sources.

portion of the seismicity during our two-week test period. Figure 7.2 shows locations of test events from the combined reference bulletin.

7.2 Evaluation

We compare SIGVISA's performance to several existing monitoring systems that also use the IMS network.

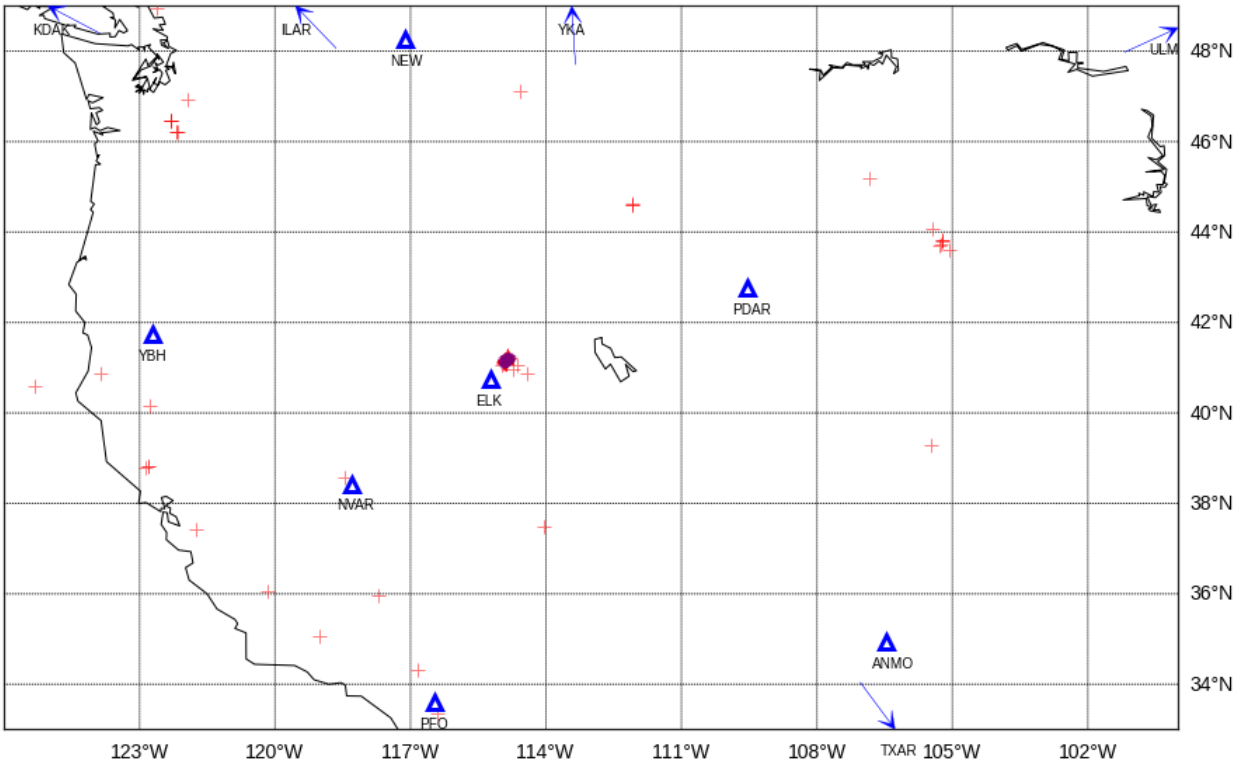
- **SEL3**: final-stage automated bulletin from the CTBTO's existing Global Association (GA) system.
- **LEB**: Late Event Bulletin produced by human analyst review of the SEL3 bulletin.
- **NETVISA**: detection-based Bayesian monitoring (Arora et al., 2013).

Note that SEL3, LEB, and NETVISA generate bulletins using the full IMS network, not limited to the twelve stations used by SIGVISA. All LEB events during our test period were associated to at least three stations within the twelve we considered, so are detectable in principle from those stations alone. However, many events did associate at additional stations, so it is possible that using the full IMS network might further improve SIGVISA's performance (at considerable computational expense).

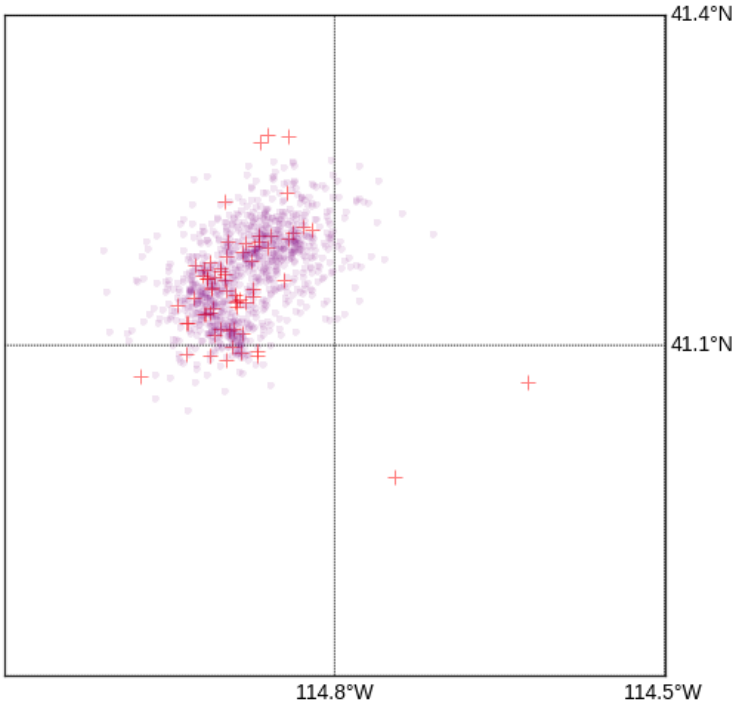
Following the training procedures in Chapter 6, we trained two sets of SIGVISA models, using broadband (0.8-4.5Hz) signals as well as a higher-frequency band (2.0-4.5Hz) intended to provide clearer evidence of regional events. To produce a test bulletin, we ran three MCMC chains on broadband signals and two additional chains on the higher-frequency signals, with each chain parallelized as described in Section 5.7, and merged the results from all five chains as described in Section 5.8. Each chain used 168 cores, one per two-hour block, for 48 hours. We used Microsoft Azure D12v2 virtual machines; as of June 2016 this corresponds to a cost of approximately $750 per chain to perform inference on this two-week test set.

We evaluate each system by comparing its inferred bulletin, computed using only IMS network data, to the ISC/UNR regional bulletin, which we treat as ground truth. We create a bipartite graph from the inferred and true bulletins, with an edge between inferred and true events separated by at most 2° in distance and 50s in time. The weight of the edge is the distance between the two events. Finally, a minimum weight maximum cardinality matching is computed on the graph. Using this matching, we report precision (the percentage of inferred events that are real), recall (the percentage of real events detected by each system), and mean location error of matched events. For NETVISA and SIGVISA, which attach a confidence score to each event, we report a precision-recall curve parameterized by the confidence threshold.

As shown in Figure 7.3, the merged SIGVISA bulletin dominates both NETVISA and SEL3. When operating at the same precision as SEL3 (51%), SIGVISA achieves recall of 19.3% versus SEL3's 6.4%, also eclipsing the 7.3% recall achieved by NETVISA at a slightly higher precision (54.7%). Unsurprisingly, the analyst-reviewed LEB contains very few false



(a) Full region.



(b) Close-up of Wells aftershocks.

Figure 7.2: Reference event locations from the two-week test period. Red crosses indicate ISC events; with the UNR bulletin in purple.

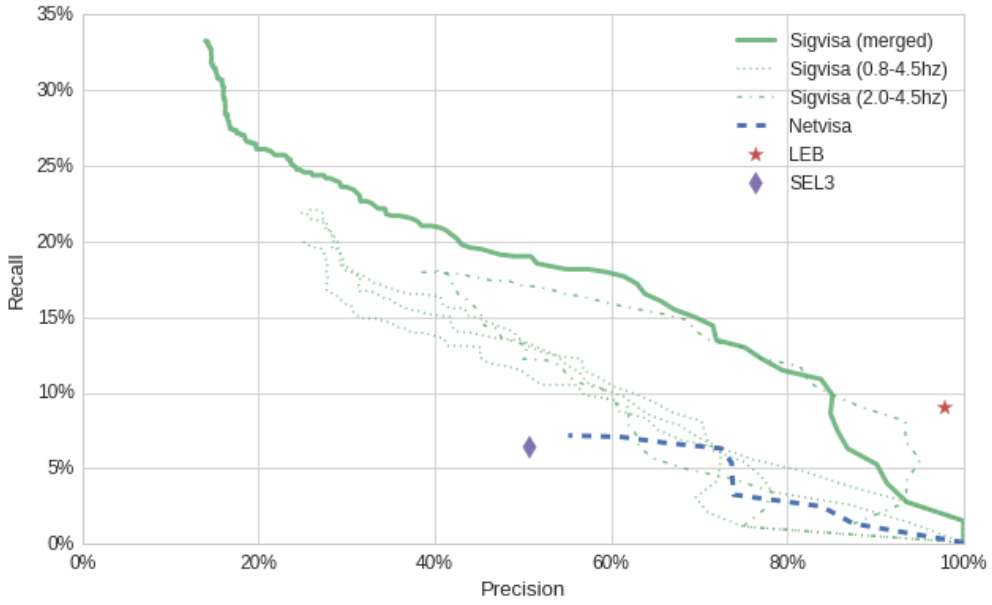


Figure 7.3: Precision-recall performance over the two-week test period, relative to the ISC/UNR reference bulletin.

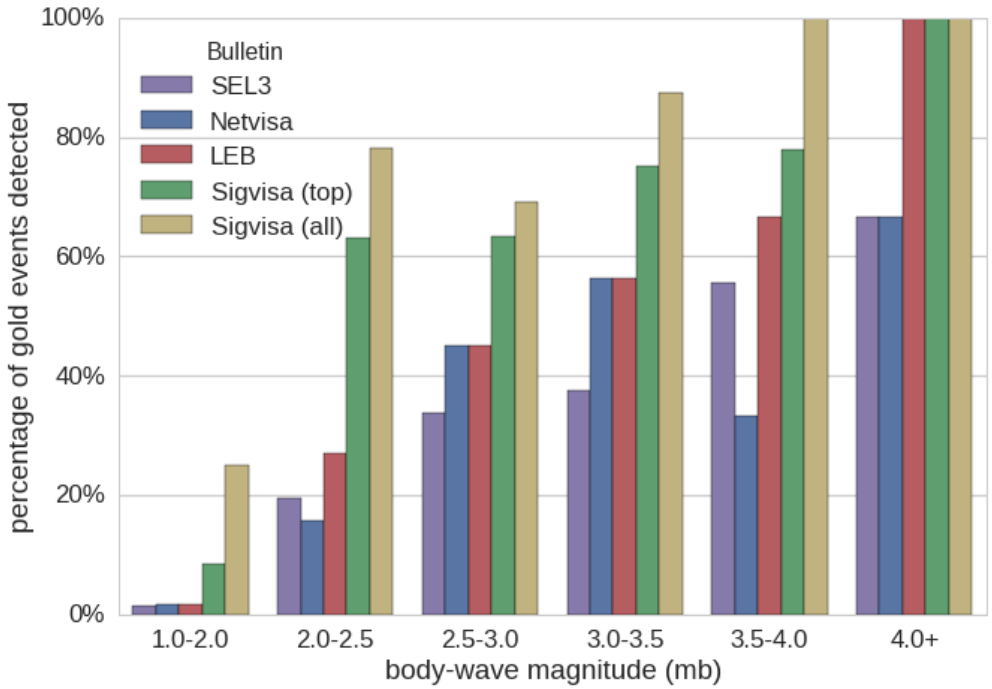


Figure 7.4: Number of reference events detected, by event magnitude. The SIGVISA (top) bulletin is defined to match the precision of SEL3 (51%).

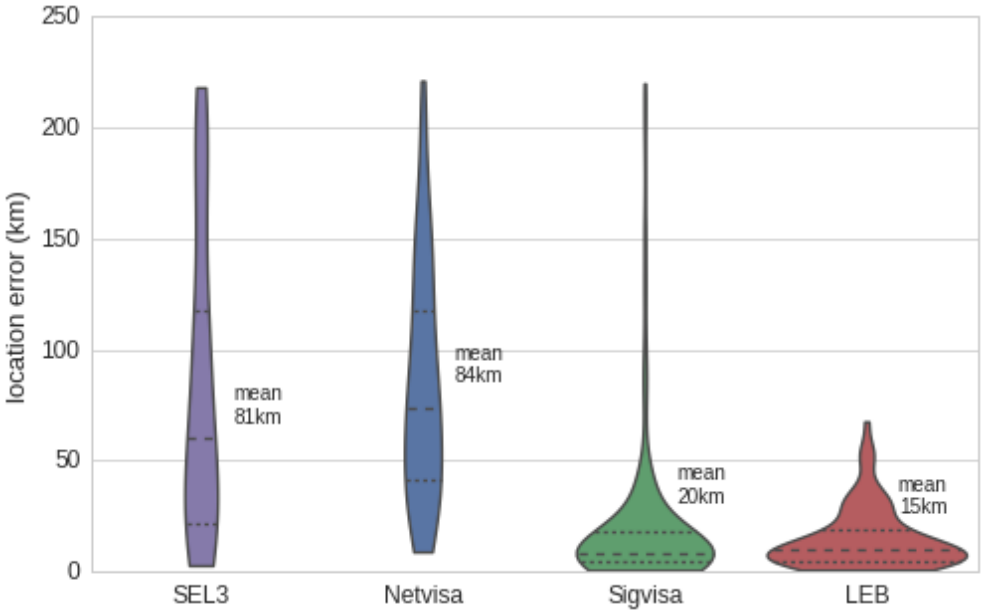


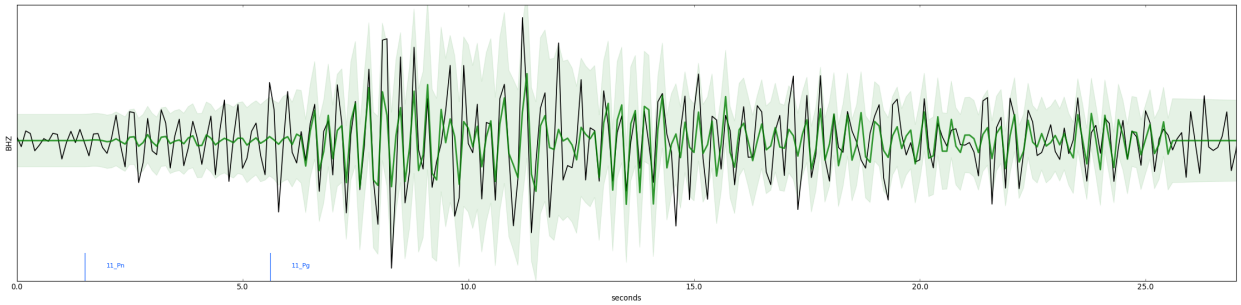
Figure 7.5: Distribution of location errors.

events,[3] achieving 97.9% precision relative to our reference bulletin, at 9.0% recall. At the other extreme, the full, un-thresholded SIGVISA bulletin recovers a full 33.3% of the reference events, though at the cost of generating many false events (14% precision).

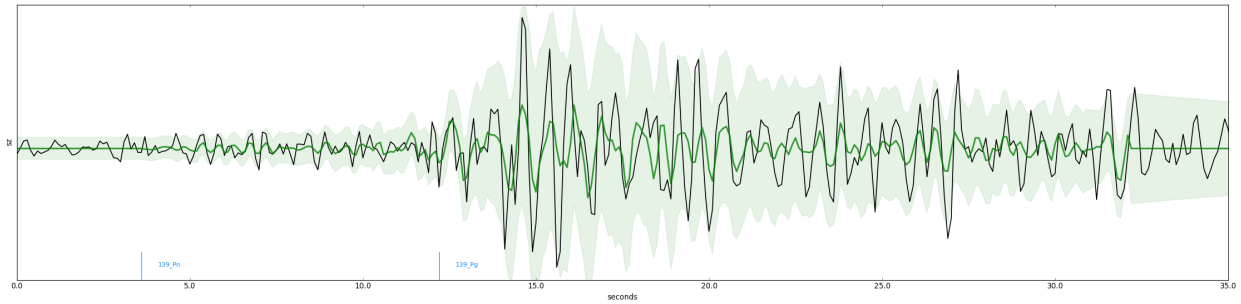
Because our analysis is performed with respect to the ISC/UNR reference bulletin, it may classify as false some genuine events that occur in regions where this bulletin does not have the same strength of coverage as for the Wells aftershocks. Figure 7.6 shows two such events inferred by SIGVISA, which are to our eyes probably genuine, due to strong correspondence between the model-predicted and observed waveforms, but are not present in the reference bulletin. The existence of such events provides reason to believe that SIGVISA's true performance on this dataset is modestly higher than our evaluation suggests.

Much of SIGVISA's performance advantage comes from increased sensitivity to low-magnitude events. Figure 7.4 breaks down each system's recall into event magnitude ranges. Below magnitude 2.5, we recover dramatically more events than the detection-based bulletins, and even detect a number of sub-2.0 events. Such small events are typically visible at no more than one or two IMS stations, and so can only be inferred using signal-based evidence such as waveform correlations. The ability to exploit correlation also improves SIGVISA's location accuracy; as shown in Figure 7.5. For events with historical waveform information available, SIGVISA is able to infer locations to within a few tens of kilometers, as opposed to hundreds for traditional systems. Figures 7.7, 7.8, 7.9, 7.11, 7.10 show the locations of inferred events for SEL3, LEB, NETVISA, and SIGVISA's top-events and full

[3] An alternate interpretation is that, since the IDC is a contributor to the ISC, the ISC bulletin will naturally tend to include LEB events regardless of their actual ground truth.



(a) Likely mining explosion at Black Thunder Mine. Location 105.21° W, 43.75° N, depth 1.9km, origin time 17:15:58 UTC, 2008-02-27, mb 2.6, recorded at PDAR (PD31).



(b) Event near Cloverdale, CA along the Rodgers Creek fault. Location 122.79° W, 38.80° N, depth 1.6km, origin time 05:20:56 UTC, 2008-02-29, mb 2.6, recorded at NVAR (NV01).

Figure 7.6: Waveform correlation evidence for arriving Pn/Pg phases of two example events detected by SIGVISA but not present in the ISC regional bulletin, and thus classified as false detections by our evaluation. Green indicates the model predicted signal (shaded $\pm 2\sigma$), based on historical events at each location, while black is the observed signal (vertical component, filtered 0.8-4.5Hz).

bulletin respectively, with respect to the ISC/UNR reference bulletin (red dots).

7.3 *de novo* events

For nuclear monitoring it is particularly important to detect *de novo* events: those occurring in locations with no historical seismicity. Approaches based purely on waveform correlation are not directly applicable in this setting, although removing the repeated events detected by such systems may ease the task of associating the remaining arrivals with *de novo* events. Since SIGVISA combines elements of correlation-based systems with more traditional multilateration, it is reasonable to hope that it would at least match the performance of more traditional systems in detecting and locating de novo events.

To interrogate this hypothesis, we identify within our test set a subset of de novo events, which we define as any event whose surface location (as given in the ISC bulletin) is at least

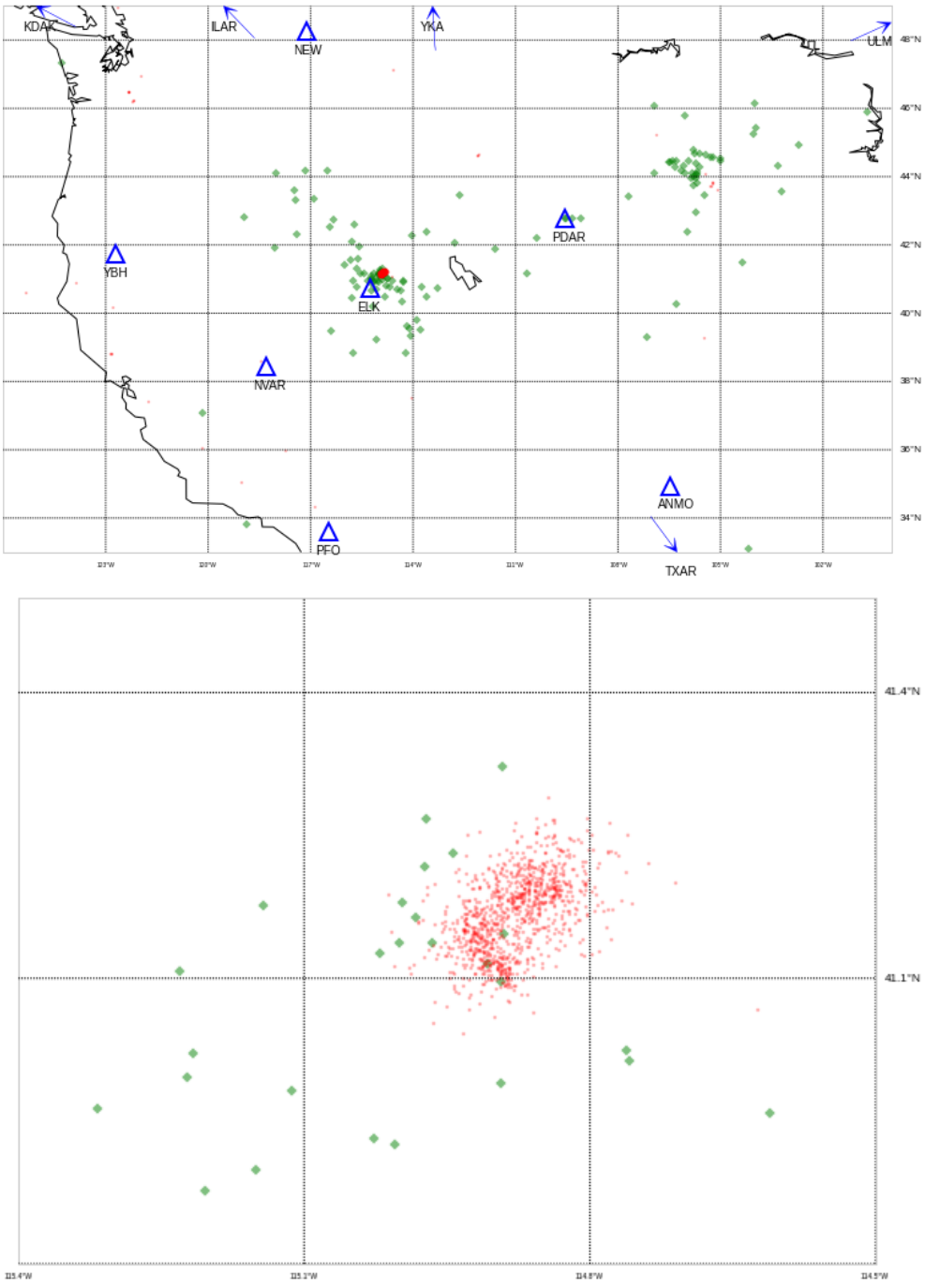


Figure 7.7: SEL3 inferred bulletin (132 events), with close-up of Wells aftershocks.

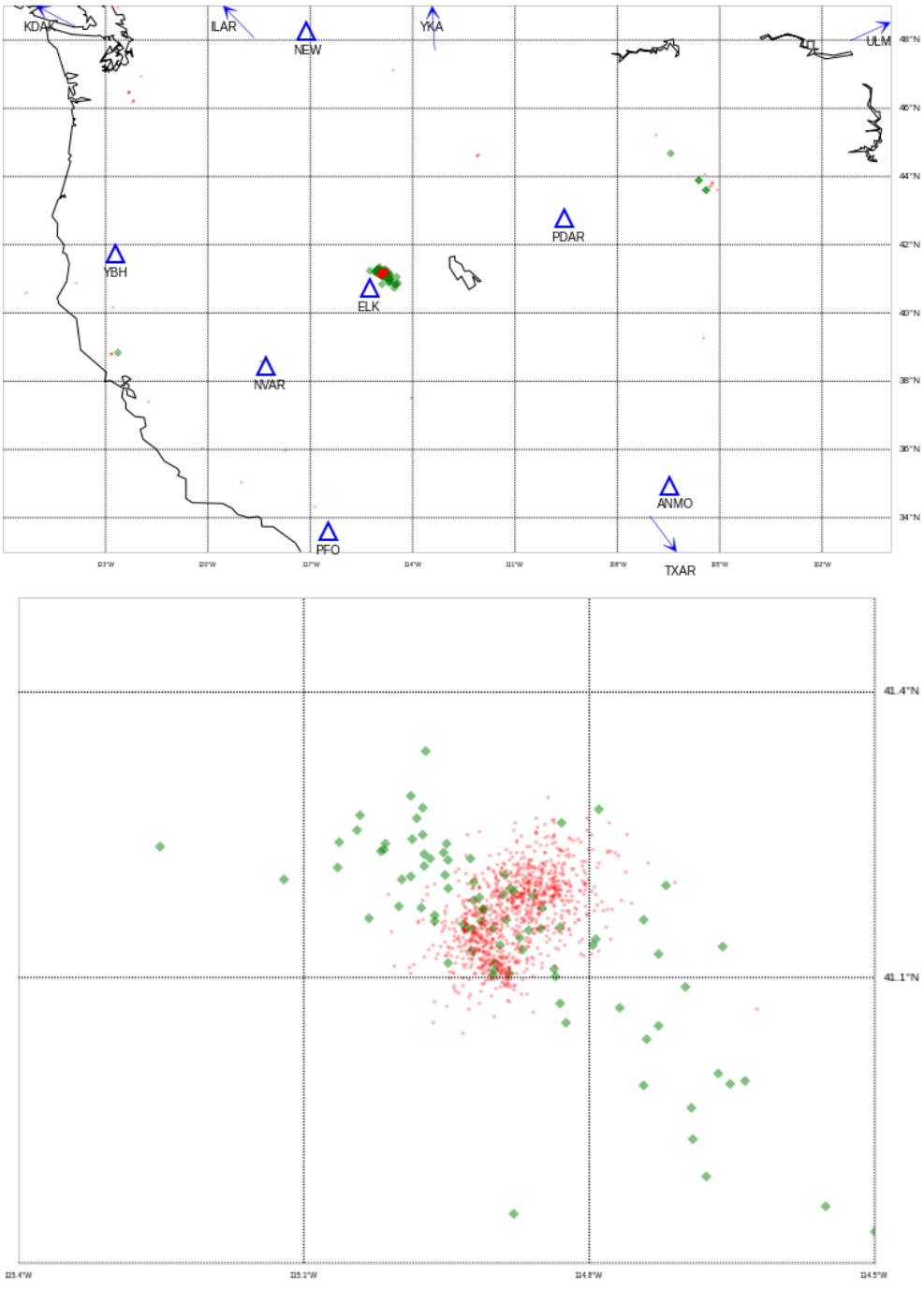


Figure 7.8: LEB inferred bulletin (96 events), with close-up of Wells aftershocks.

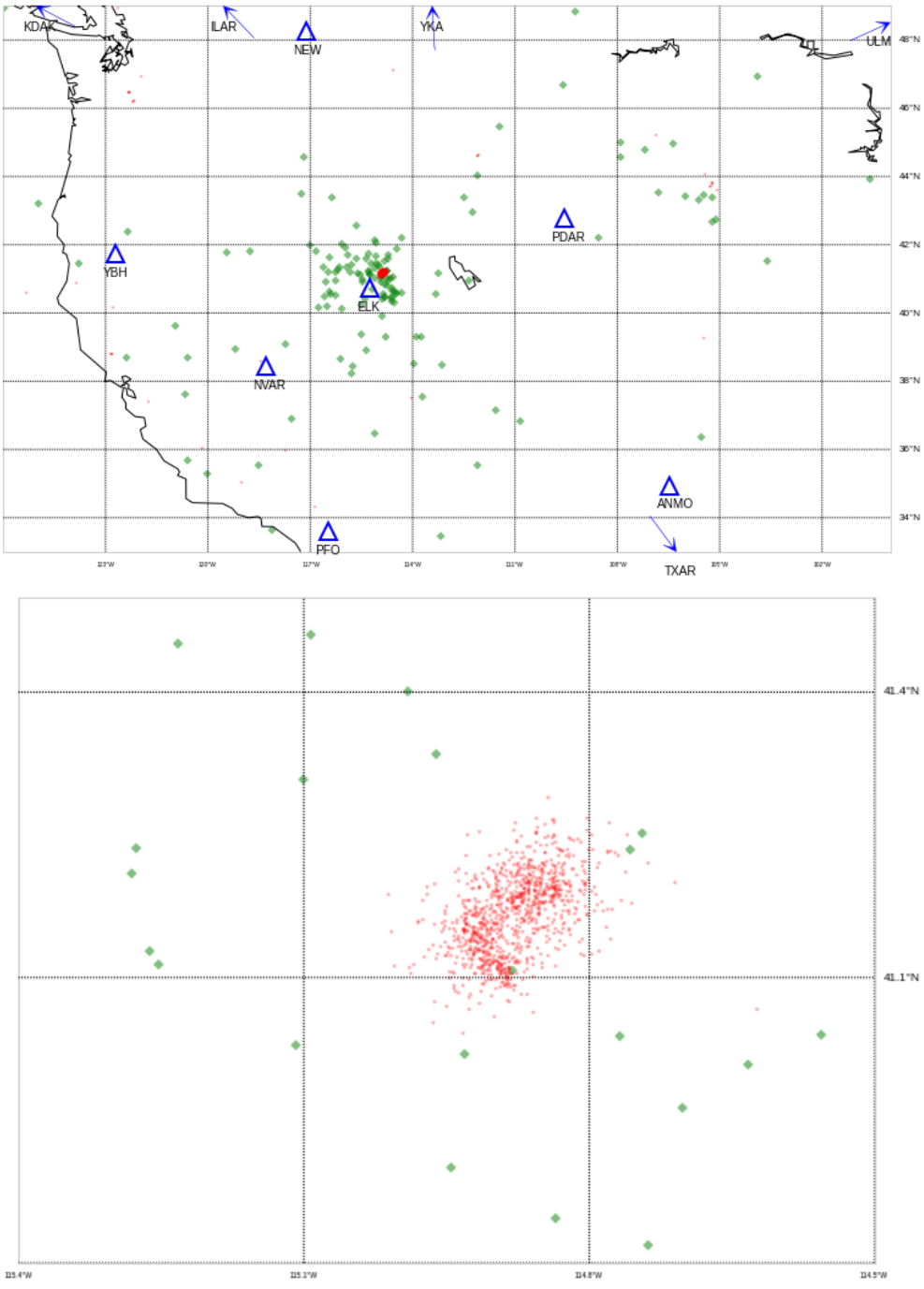


Figure 7.9: NETVISA inferred bulletin (139 events), with close-up of Wells aftershocks.

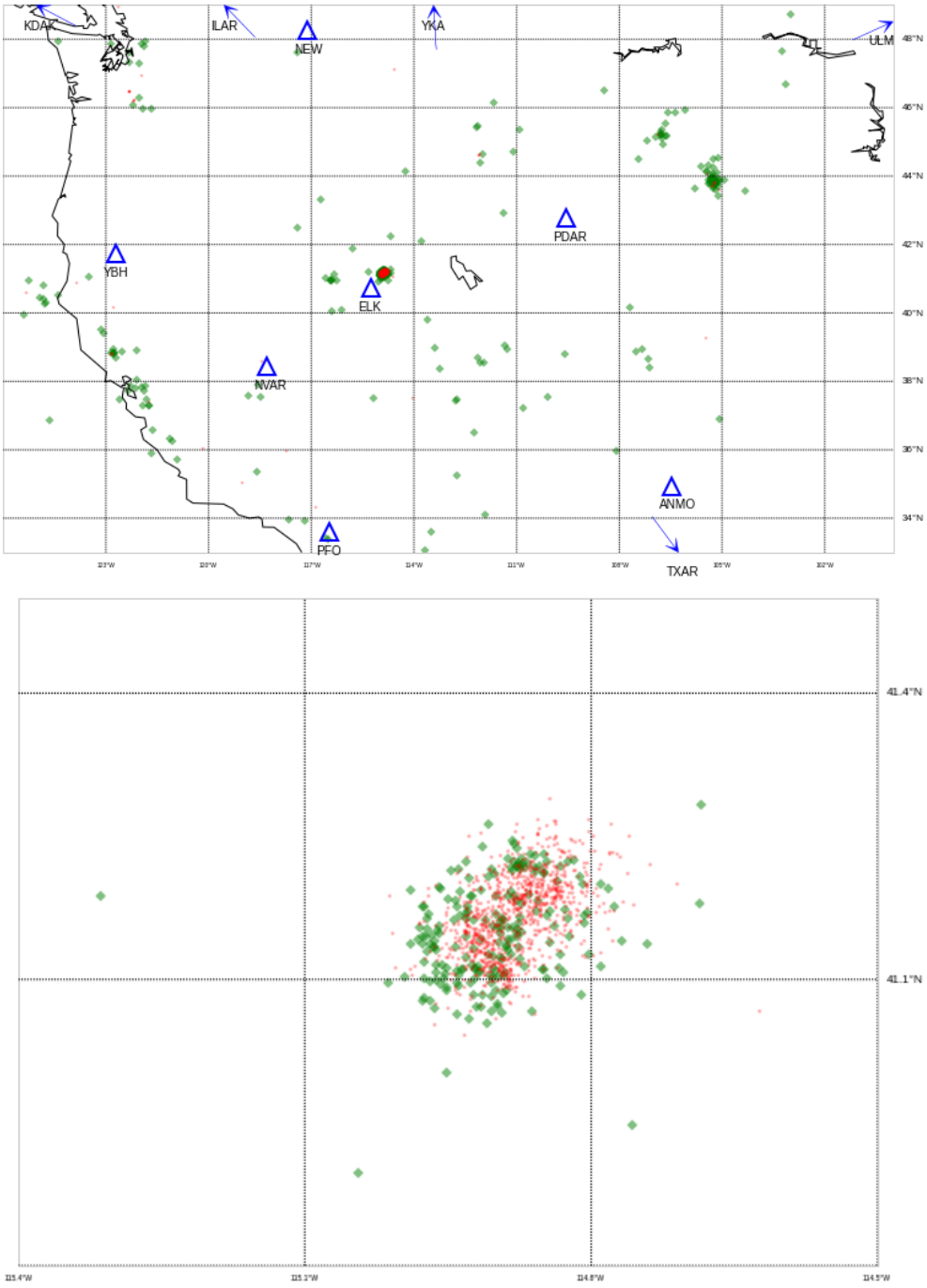


Figure 7.10: SIGVISA top-events bulletin (393 events, 51% precision matching SEL3), with close-up of Wells aftershocks.

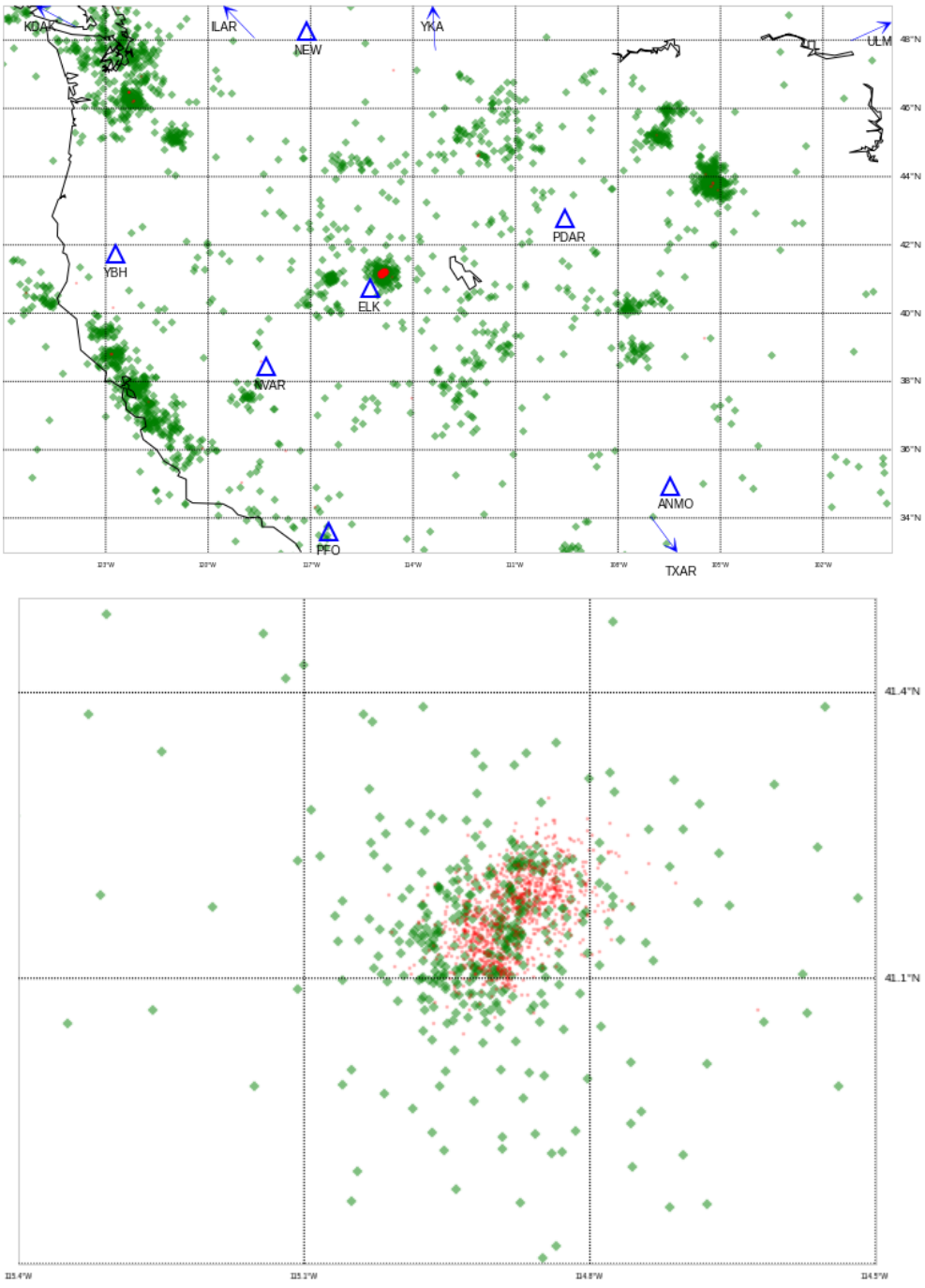


Figure 7.11: SIGVISA full bulletin (2491 events), with close-up of Wells aftershocks.

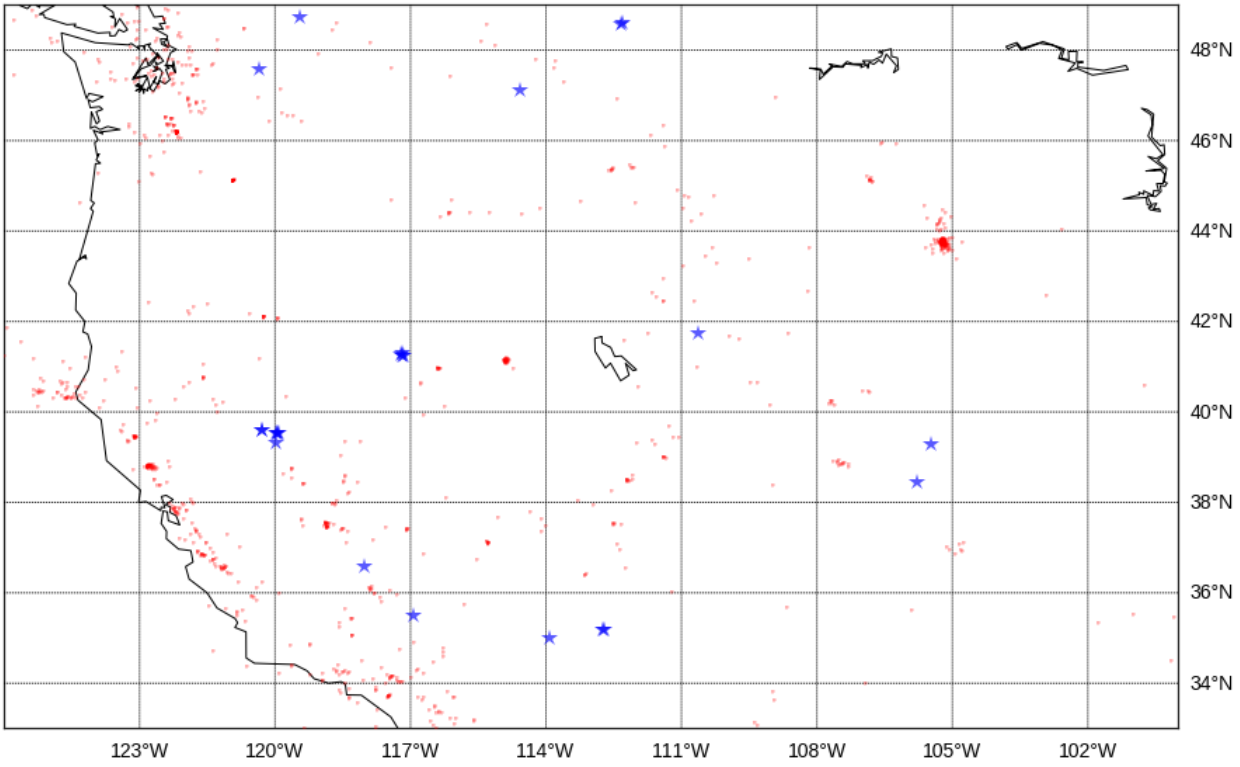
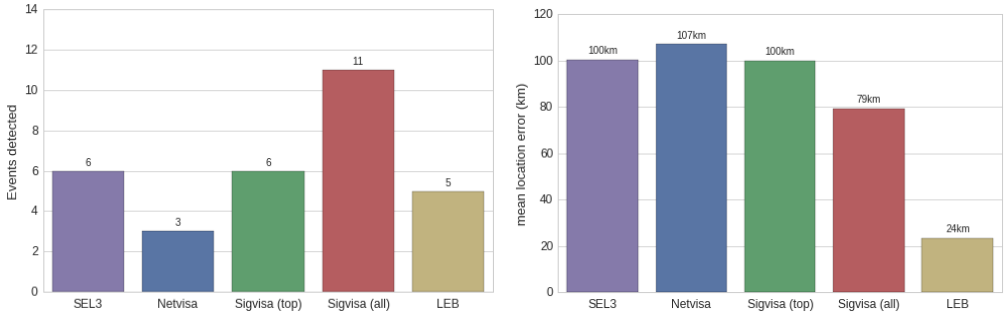


Figure 7.12: Locations of *de novo* test events (blue stars) relative to training events.



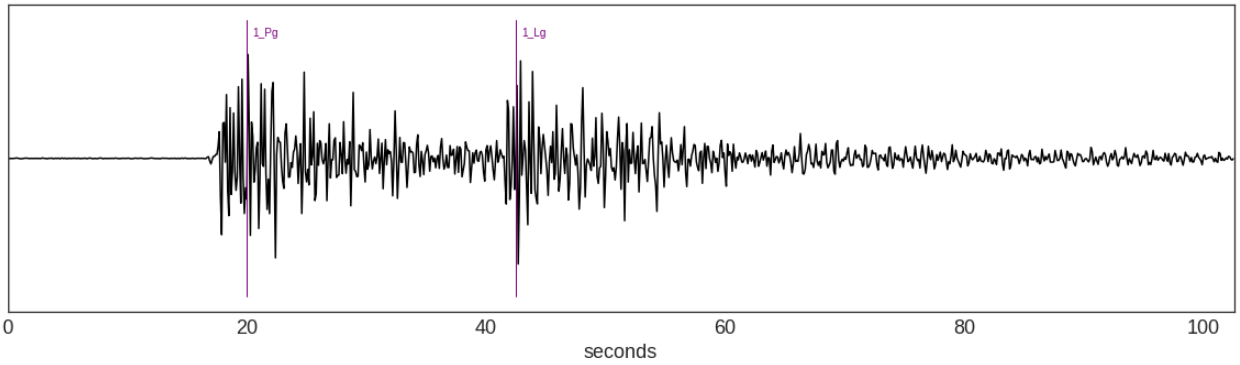
(a) Recall (b) Mean location error

Figure 7.13: Results for 24 *de novo* events between January and March 2008.

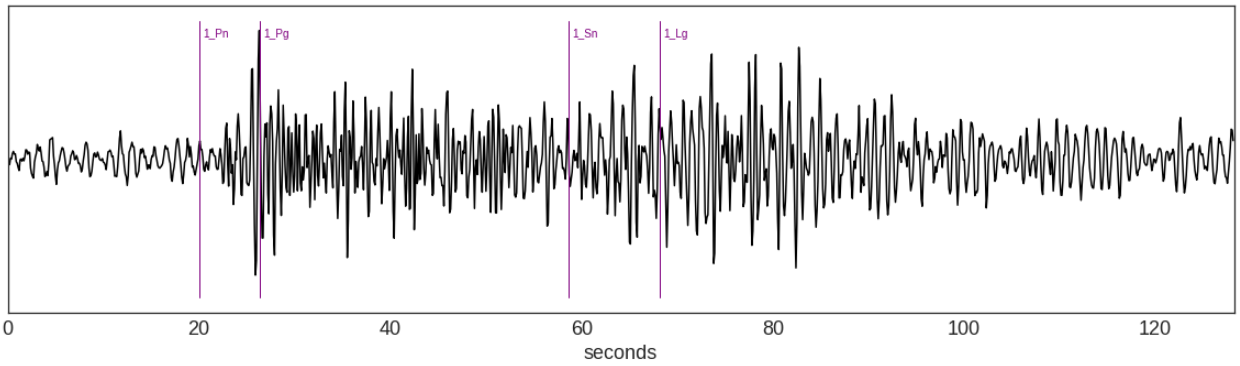
50km from the nearest event in the training set. As there are only three such events within the original two-week test period, we broaden the scope to the three-month period from January 1, 2008 through March 31, 2008, which includes twenty-four de novo events, shown in Figure 7.12.

To evaluate SIGVISA on these events, we ran inference on the hour-long period surrounding each of these events. Specifically, we construct a bulletin by merging four SIGVISA chains for each time period, each run for 48 hours, using broadband (0.8-4.5Hz) signals. For all other systems, we extract data from their catalogues for the corresponding periods. For each system we focus on recall specifically of de novo events: of the de novo events in the ISC reference bulletin, how many were detected?

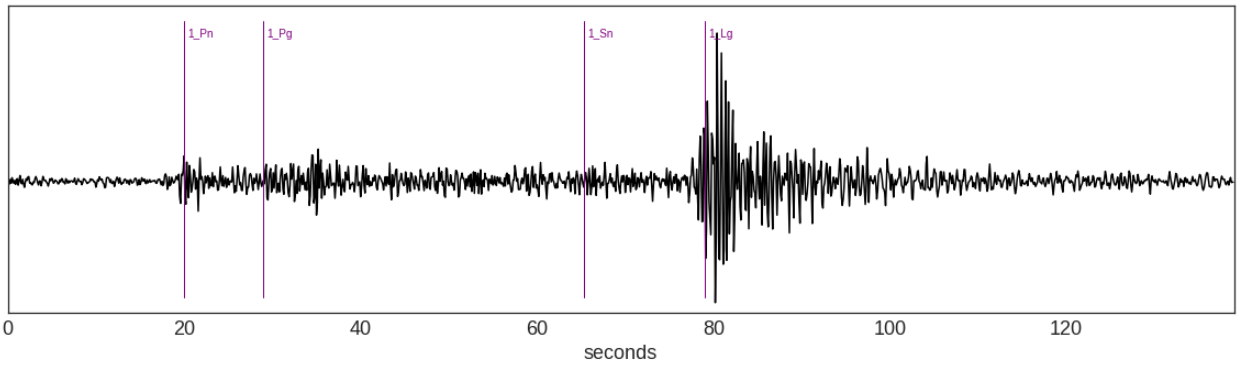
As shown in Figure 7.13, SIGVISA's de novo performance matches or exceeds the other systems. Operating at the same precision as SEL3, it detects the same number (6/24) of de novo events, while achieving comparable location accuracy. Although the sample size is small, this is encouraging evidence that SIGVISA's performance on repeated events – which, to be clear, includes by our definition almost all of the natural seismicity in the western US during our two-week test period – does not come at a cost for de novo events. Indeed, the full high-sensitivity SIGVISA bulletin includes six events registered by the ISC but not by any other IMS-based system. Figure 7.14 shows signals from one of these events. Although arrivals are visible at all three stations, station processing at ELK did not register any detections, preventing detection-based systems from building the event. It is not clear why this occurred; nonetheless, it does not prevent SIGVISA from building the event and locating it relatively accurately (approximately 20km from its location in the regional catalog).



(a) NVAR (distance 183km).



(b) YBH (distance 342km).



(c) ELK (distance 407km).

Figure 7.14: Vertical component broadband (0.8-4.5Hz) signals for the three IMS stations nearest to ISC evid 13484219, located by SIGVISA at 119.79°W, 39.60°N, with vertical bars indicating expected phase arrivals. The event was missed by SEL3, NETVISA, and LEB, presumably because there are no detections registered at ELK for this time period.

Chapter 8

Conclusions and future directions

The results in this thesis demonstrate the promise of signal-based Bayesian monitoring. We have described a generative probability model of repeatable seismic signals and developed an MCMC algorithm to perform effective inference, along with procedures for parallel training and inference to support large scale data sets. In an evaluation on IMS signals from the western United States, we showed that the system proposed in this thesis detects up to three times as many events as a detection-based baseline (SEL3) while operating at the same precision, reduces mean location errors by a factor of four, greatly increases sensitivity to low-magnitude events, and maintains effective performance even for events in regions with no historical seismicity in the training set.

Despite these achievements, there is still much work to be done. One obvious goal for future work is an evaluation of SIGVISA for global monitoring, using the full IMS seismic network rather than the 12 stations used in this thesis. From a computational perspective, we expect that our current system implementation should scale to this setting, given straightforward refinements including multicore parallelism and careful memory management. Some changes to the model may also become necessary as the focus shifts from regional to teleseismic (long-distance) arrivals, for example, considering additional phase types and improved travel time models.

One issue likely to emerge in global monitoring is the need to *relocate* events during the training process, since ground-truth regional data are not available throughout much of the world. Using waveform correlation evidence, Schaff and Richards (2011) find that global bulletins such as the CTBTO's Reviewed Event Bulletin contain significant location errors, with some events mislocated by up to hundreds of kilometers. Thus our training procedure cannot treat historical bulletins as ground truth, but should relocate the events under the joint model so that we learn coherent waveforms that vary smoothly between nearby events. This is a highly nonconvex optimization problem and will likely be quite computationally challenging. On the other hand, an efficient solution for joint relocation of historical events using our model would likely be of independent interest to seismology as a model-based Bayesian counterpart to existing methods such as double-differencing.

To obtain good locations from teleseismic arrivals, it may also be necessary to make more

explicit use of directional information arising from three-component stations and seismic arrays. In the case of three-component stations, we could model the components as recording noisy projections of a single underlying latent signal for each event, with the projection coefficients determined by the angle of the incoming ray. For array stations, we could treat each element as its own full-fledged station, and relax the independence assumptions of our model to treat signals at nearby array elements as correlated. However, directly modeling arrays in this way may be computationally prohibitive, so that intermediate solutions making use of existing array beamforming could be desirable.

Another avenue for future work involves extending our models of source mechanisms beyond simple point sources, to encompass more complex, potentially anisotropic sources having nonzero duration and spatial extent. Although this has not been the focus of our efforts, a Bayesian system should be able to directly answer queries discriminating between earthquake and explosion sources. Work in this direction would likely occur in concert with explicit modeling of signals across multiple, narrow, frequency bands, which form useful explosion discriminants as well as yielding more stable yield estimates from predictable coda decays (Mayeda et al., 2003).

There is still a great deal of work to be done in quantifying the performance of SIGVISA relative to conventional waveform correlation matching. In particular, the wavelet models of repeatable modulation described in this thesis have not been heavily tuned. It is possible they could be greatly improved by substituting the wavelet basis with a custom basis learned during the training process, to provide a data-driven representation that would be more compact and informative than our current representation via wavelet coefficients. This would in effect be a form of probabilistic principal components analysis (Bishop, 2006), with a Gaussian process prior on the coefficients to impose spatial coherence, and a compact support constraint on the basis itself to preserve the state space model formulation for efficient inference. There is some precedent for a similar approach in the work on correlation subspace detectors (Harris, 1997), which form correlation templates using principle components of aligned historical signals.

An advantage of the explicitly generative approach taken in this thesis is that it is easy to quantify the system's estimates of its own limits, by performing inference on signals generated by sampling from the model. Thus it should be possible to quantify the system's detection threshold as a function of event location, and to actively propose locations for new sensors to maximize network coverage.

We finally note that, although one of the goals of Bayesian monitoring is to separate modeling from inference algorithms, so that model improvements by domain experts lead directly to improved system performance, current implementations often fall short of this ideal. Much of the effort in developing SIGVISA went into implementing the custom inference algorithms described in Chapter 5, which are closely inspired by the model structure, and would likely need to be modified in the face of structural model changes including many of the improvements proposed in this chapter. Ongoing research into probabilistic programming systems, such as Bayesian Logic (BLOG) (Milch et al., 2005; Wu et al., 2016), Stan (Carpenter et al., 2016), and Venture (Mansinghka et al., 2014), promises to bring the real-

ity of Bayesian modeling closer to the ideal separation of model and inference. In the long run, expressing SIGVISA as a probabilistic program, with automatically derived inference algorithms, would be an excellent stress test for such systems. Furthermore, a working implementation would significantly speed up the process of iteratively developing and testing new model improvements. By allowing seismologists to build, evaluate, and share generative models in directly computable form, probabilistic programming would not only lead to more effective monitoring systems; it could represent a modern evolution of the scientific method itself.

Appendix A

Multivariate Gaussians

The section defines the multivariate Gaussian distribution, derives its density, and demonstrates that affine transformations, marginals, and conditionals of Gaussian random vectors are themselves Gaussian. We additionally show that products and quotients of Gaussian *densities* also maintain the form of a Gaussian density. These results are used to define Gaussian processes and state-space models (Chapter 3), and in particular to efficiently compute the marginal likelihood of the SIGVISA signal model by marginalizing out the wavelet coefficients describing each arrival (Section 4.9). All are standard and can be found in sources such as Rasmussen and Williams (2006), Koller and Friedman (2009), and Gelman et al. (2014).

We say that a random vector $\mathbf{x}$ is multivariate Gaussian with mean μ and covariance matrix $\mathbf{B}\mathbf{B}^T$ if it can be written as a transformation

$$\mathbf{x} = \mu + \mathbf{B}\mathbf{z},$$

where $\mathbf{z}$ is a vector of i.i.d. standard Gaussian random variables, $z_i \sim \mathcal{N}(0,1)$. This definition is justified by noting that the mean

$$E[\mathbf{x}] = \mu + \mathbf{B}E[\mathbf{z}] = \mu$$

and covariance

$$E\left[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T\right] = E[(\mathbf{B}\mathbf{z})(\mathbf{B}\mathbf{z})^T] = \mathbf{B}E[\mathbf{z}\mathbf{z}^T]\mathbf{B}^T = \mathbf{B}\mathbf{B}^T$$

are as desired.

Given a positive (semi)definite matrix $\mathbf{\Sigma}$, we construct a multivariate normal distribution with covariance $\mathbf{\Sigma}$ by finding $\mathbf{B}$ such that $\mathbf{\Sigma} = \mathbf{B}\mathbf{B}^T$. The Cholesky decomposition produces such a $\mathbf{B}$ directly; another approach is to consider the spectral decomposition $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where the columns of $\mathbf{U}$ are unit eigenvectors and $\mathbf{\Lambda}$ is the corresponding diagonal matrix of eigenvalues, and then take

$$\mathbf{B} = \mathbf{U}\mathbf{\Lambda}^{1/2},$$

where $\mathbf{\Lambda}^{1/2}$ is guaranteed to be real-valued since the eigenvalues of a positive semidefinite matrix are nonnegative.

Under this definition it is straightforward to derive the multivariate Gaussian density function. We begin with the i.i.d. Gaussian density on $\mathbf{z}$,

$$p(\mathbf{z}) = \prod_{i=1}^{n} p(z_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z_i^2} = \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}\mathbf{z}^T\mathbf{z}\right),$$

and perform the change of variables $\mathbf{z} = \mathbf{B}^{-1}(\mathbf{x} - \mu)$ to yield the multivariate Gaussian density,

$$\begin{aligned} p(\mathbf{x}) &= \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T(\mathbf{B}^{-1})^T\mathbf{B}^{-1}(\mathbf{x}-\mu)\right)\left|\frac{d\mathbf{z}}{d\mathbf{x}}\right| \\ &= \frac{1}{(2\pi)^{n/2}\left|\mathbf{\Sigma}\right|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right), \end{aligned} \tag{A.1}$$

in which we have used the Jacobian determinant

$$\left|\frac{d\mathbf{z}}{d\mathbf{x}}\right| = \left|\frac{d\left(\mathbf{B}^{-1}\mathbf{x} - \mu\right)}{d\mathbf{x}}\right| = |\mathbf{B}^{-1}| = |\mathbf{\Sigma}|^{-1/2}.$$

Note that this density is only defined when $\mathbf{\Sigma}$ is positive definite, i.e., when $\mathbf{B}$ is a square invertible matrix.

A.1 Affine transformations

It follows immediately that any affine transformation $\mathbf{w} = \mathbf{P}\mathbf{x} + \mathbf{b}$ of a Gaussian random vector $\mathbf{x}$ is itself multivariate Gaussian. Let $\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{\Sigma})$, so we can write

$$\mathbf{x} = \mathbf{B}\mathbf{z} + \mu$$

for some $\mathbf{B}$ s.t. $\mathbf{B}\mathbf{B}^T = \mathbf{\Sigma}$ and i.i.d. standard Gaussian vector $\mathbf{z}$. Therefore we have

$$\begin{aligned} \mathbf{w} &= \mathbf{P}(\mathbf{B}\mathbf{z} + \mu) + \mathbf{b} \\ &= \mathbf{P}\mathbf{B}\mathbf{z} + (\mathbf{P}\mu + \mathbf{b}), \end{aligned}$$

and from this form we can simply read off $\mathbf{w}$ as Gaussian with covariance $(\mathbf{P}\mathbf{B})(\mathbf{P}\mathbf{B})^T = \mathbf{P}\mathbf{\Sigma}\mathbf{P}^T$, verifying our conclusion

$$\mathbf{w} \sim \mathcal{N}(\mathbf{P}\mu + \mathbf{b}, \mathbf{P}\mathbf{\Sigma}\mathbf{P}^T). \tag{A.2}$$

As a special case, by choosing $\mathbf{P} = [\mathbf{I}, \mathbf{I}]$, so that $\mathbf{P}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \mathbf{x} + \mathbf{y}$, we see that the sum of independent Gaussian vectors $\mathbf{x} \sim N(\mathbf{a}, \mathbf{A})$ and $\mathbf{y} \sim N(\mathbf{b}, \mathbf{B})$ is itself Gaussian,

$$\mathbf{x} + \mathbf{y} \sim \mathcal{N}(\mathbf{a} + \mathbf{b}, \mathbf{A} + \mathbf{B}). \tag{A.3}$$

A.2 Marginalization and conditioning

Suppose $\mathbf{x}$ and $\mathbf{y}$ are vectors jointly Gaussian distributed:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{B} \end{bmatrix} \right).$$

Then it follows immediately by the linear transformation

$$\mathbf{x} = [\mathbf{I}, \mathbf{0}] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$

that the *marginal distribution* $p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y}$ is a Gaussian obtained simply by reading off the relevant submatrix of the joint distribution,

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{a}, \mathbf{A}). \tag{A.4}$$

The conditional distribution $p(\mathbf{x}|\mathbf{y})$ is also Gaussian:

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}\left(\mathbf{a} + \mathbf{C}^T\mathbf{B}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{C}^T\mathbf{B}^{-1}\mathbf{C}\right). \tag{A.5}$$

This can be shown by considering the Schur complement of the covariance matrix; see, e.g., Von Mises (1964, ch. VIII, section 9) for details. Note that $p(\mathbf{y})$ and $p(\mathbf{y}|\mathbf{x})$ follow immediately by symmetry.

A.3 Products and quotients

Given two Gaussian densities in the same variable $\mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A})$ and $\mathcal{N}(\mathbf{b}; \mathbf{x}, \mathbf{B})$, their product is an *unnormalized* Gaussian density

$$\begin{aligned} \mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A}) N(\mathbf{x}; \mathbf{b}, \mathbf{B}) &= z_c \cdot \mathcal{N}(\mathbf{x}; \mathbf{c}, \mathbf{C}) \\ \mathbf{c} &= \mathbf{C}\left(\mathbf{A}^{-1}\mathbf{a} + \mathbf{B}^{-1}\mathbf{b}\right) \\ \mathbf{C} &= \left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)^{-1}, \end{aligned} \tag{A.6}$$

with normalization constant also given by a Gaussian density,

$$z_c = \mathcal{N}\left(\mathbf{b}; \mathbf{a}, \mathbf{B} + \mathbf{A}\right).$$

This result can be easily extended by induction to the product of multiple Gaussian densities. The quotient of Gaussian densities can be derived similarly,

$$\begin{aligned} \frac{\mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A})}{\mathcal{N}(\mathbf{x}; \mathbf{b}, \mathbf{B})} &= z_d \cdot \mathcal{N}(\mathbf{x}; \mathbf{d}, \mathbf{D}) \\ \mathbf{d} &= \mathbf{D}\left(\mathbf{A}^{-1}\mathbf{a} - \mathbf{B}^{-1}\mathbf{b}\right) \\ \mathbf{D} &= \left(\mathbf{A}^{-1} - \mathbf{B}^{-1}\right)^{-1} \\ z_d &= \frac{|\mathbf{B}|}{|\mathbf{B} - \mathbf{A}|} \frac{1}{\mathcal{N}\left(\mathbf{b}; \mathbf{a}, \mathbf{B} + \mathbf{A}\right)}. \end{aligned} \tag{A.7}$$

We prove the result for the product of Gaussian densities; the quotient derivation is similar. We first introduce the following identity for *multivariate completion of squares*,

$$\frac{1}{2}\mathbf{z}^T\mathbf{P}\mathbf{z} + \mathbf{q}^T\mathbf{z} + \mathbf{r} = \frac{1}{2}(\mathbf{z} + \mathbf{P}^{-1}\mathbf{q})^T\mathbf{P}(\mathbf{z} + \mathbf{P}^{-1}\mathbf{q}) + \mathbf{r} - \frac{1}{2}\mathbf{q}^T\mathbf{P}^{-1}\mathbf{q}, \tag{A.8}$$

which is easy to verify by expanding the right side. We then write out the densities explicitly,

$$\begin{aligned}
\mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A})\mathcal{N}(\mathbf{x}; \mathbf{b}, \mathbf{B}) &\propto \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{a})^T\mathbf{A}^{-1}(\mathbf{x} - \mathbf{a})\right) \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{b})^T\mathbf{B}^{-1}(\mathbf{x} - \mathbf{b})\right) \\
&\propto \exp\left(-\frac{1}{2}\mathbf{x}^T\left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)\mathbf{x} - \left(\mathbf{a}^T\mathbf{A}^{-1} + \mathbf{b}^T\mathbf{B}^{-1}\right)\mathbf{x}\right) \\
&\propto \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x} - (\mathbf{C}^{-1}\mathbf{c}^T)\mathbf{x}\right),
\end{aligned}$$

and apply the identity (A.8), with $\mathbf{P} = \mathbf{C}^{-1}$, $\mathbf{q} = \mathbf{C}^{-1}\mathbf{c}^T$, and $\mathbf{r} = 0$, to yield

$$\propto \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c})^T\mathbf{C}^{-1}(\mathbf{x} - \mathbf{c})\right) \propto N(\mathbf{x}; \mathbf{c}, \mathbf{C}),$$

thus proving our result. Verifying the normalization constant z_c is left as an exercise for the reader.

Appendix B

Probabilistic interpretations of normalized correlation

The normalized correlation of two signal windows $\mathbf{a}$ and $\mathbf{b}$,

$$\kappa = \frac{\mathbf{a}^T\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}, \tag{B.1}$$

is commonly used as a measure of signal similarity, for example in aligning two signals by finding their cross-correlation peak. Although it is easy to compute and often useful in practice, normalized correlation is a heuristic with no probabilistic interpretation. For example, if one alignment produces a correlation of 0.6 and another 0.55, how probable is it that the first alignment is correct? Such a query can only be answered in the context of a model for the process generating the signals.

The SIGVISA model allows us to query the probability of a particular alignment between signals, or the presence of a particular event in a noisy signal. However, the full SIGVISA model is specific to seismic monitoring and requires complex inference calculations to answer simple queries. In this note we consider simpler statistics that attempt to recover the flavor of cross-correlation within a probabilistic framework, while preserving its closed-form simplicity.

Consider two signals $\mathbf{a}$ and $\mathbf{b}$ of the same length n, which we believe to be correlated. We observe $\mathbf{b}$, and want to predict $\mathbf{a}$. In the context of seismic waveform matching, we might suppose that $\mathbf{b}$ corresponds to a historical template, and $\mathbf{a}$ a newly observed signal. A simple class of models is

$$\mathbf{a} = \alpha\mathbf{b} + \epsilon,$$

where ϵ is a noise process of some sort, and α is an unknown scale parameter. In this chapter we will consider the model in which ϵ is i.i.d. Gaussian, and then the more general case where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is an arbitrary multivariate Gaussian, for example, an autoregressive noise process.

These models are deliberately simplistic; they do not account for uncertainty in the source template $\mathbf{b}$, which in reality will be noisily observed, nor for the possibility that

the new signal $\mathbf{a}$ may be generated from a source that resembles $\mathbf{b}$ but does not match it exactly (as with seismic signals from events with different source mechanisms, or in nearby but non-identical locations). Although the full SIGVISA model does include these effects, among others (e.g., it allows the repeatable signals to be reshaped by an envelope model), the simple models analyzed in this chapter allow for efficient closed-form evaluation and may be more broadly applicable.

B.1 IID noise

We first consider the case where $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is a Gaussian white noise process of variance σ^2, so that the conditional signal is distributed as

$$\begin{aligned}\log p(\mathbf{a}|\mathbf{b};\alpha) &= \log \mathcal{N}(\mathbf{a}; \alpha\mathbf{b}, \sigma^2\mathbf{I}) \\ &= -\frac{1}{2\sigma^2}\|\mathbf{a}-\alpha\mathbf{b}\|^2 - \frac{n}{2}\log(2\pi\sigma^2) \\ &= -\frac{1}{2\sigma^2}\left(\|\mathbf{a}\|^2 - 2\alpha\mathbf{a}^T\mathbf{b} + \alpha^2\|\mathbf{b}\|^2\right) - \frac{n}{2}\log(2\pi\sigma^2). \end{aligned} \tag{B.2}$$

To estimate the amplitude α by maximum likelihood, we set the derivative,

$$\frac{\partial \log p(\mathbf{a}|\mathbf{b};\alpha)}{\partial\alpha} = \frac{1}{\sigma^2}\left(\mathbf{a}^T\mathbf{b} - \alpha\|\mathbf{b}\|^2\right),$$

to zero, and solve for α to find

$$\hat{\alpha} = \frac{\mathbf{a}^T\mathbf{b}}{\|\mathbf{b}\|^2}.$$

Since the log probability is quadratic in α with a negative leading term, this estimate is the global maximum. Note that it is closely related to the normalized correlation. In particular, we have

$$\hat{\alpha} = \kappa\frac{\|\mathbf{a}\|}{\|\mathbf{b}\|},$$

so our estimated scaling factor is just the ratio of the signal norms, adjusted by their correlation.

In many applications it is sensible to restrict to $\alpha \geq 0$, yielding the constrained MLE $\hat{\alpha}^+$, which assumes the contributed amplitude to be zero if the estimated correlation is negative.

Plugging $\hat{\alpha}$ into (B.2) yields the optimized signal likelihood $\hat{p}$:

$$\begin{aligned}\log \hat{p}(\mathbf{a}|\mathbf{b}) &= -\frac{1}{2\sigma^2}\left(\|\mathbf{a}\|^2 - 2(\mathbf{a}^T\mathbf{b})^2/\|\mathbf{b}\|^2 + (\mathbf{a}^T\mathbf{b})^2/\|\mathbf{b}\|^2\right) - \frac{n}{2}\log(2\pi\sigma^2) \\ &= -\frac{1}{2\sigma^2}\left(\|\mathbf{a}\|^2 - (\mathbf{a}^T\mathbf{b})^2/\|\mathbf{b}\|^2\right) - \frac{n}{2}\log(2\pi\sigma^2) \\ &= -\frac{1}{2\sigma^2}\|\mathbf{a}\|^2\left(1-\kappa^2\right) - \frac{n}{2}\log(2\pi\sigma^2). \end{aligned} \tag{B.3}$$

Note that this likelihood is a function of the correlation between signals *and* of the signal magnitude $\|\mathbf{a}\|$: we are explaining some portion of $\mathbf{a}$ by the presence of $\alpha\mathbf{b}$, but the remainder must be explained as noise.

We can also consider the *log odds* of the signal under this model, versus the null model with $\alpha = 0$ in which $\mathbf{a}$ is simply generated as Gaussian noise with no contribution from $\mathbf{b}$. This is just the difference of likelihoods

$$\begin{aligned}\log \hat{p}(\mathbf{a}|\mathbf{b}) - \log p(\mathbf{a}) &= -\frac{1}{2\sigma^2}\|\mathbf{a}\|^2\left(1-\kappa^2\right) - \left(-\frac{1}{2\sigma^2}\|\mathbf{a}\|^2\right) \\ &= \frac{1}{2\sigma^2}\|\mathbf{a}\|^2\kappa^2, \end{aligned} \tag{B.4}$$

where the unconditional $p(\mathbf{a})$ evaluates $\mathbf{a}$ under the noise model, i.e., $\mathbf{a} = \varepsilon$.

The log odds (B.4) can be used as a substitute for the normalized correlation in computing signal alignments. Consider a signal $\mathbf{s}$ of length $T \gg n$, so that we want to align a historical signal $\mathbf{b}$ with a length-n subwindow of $\mathbf{s}$. Then the time step t that maximizes the log odds,

$$L(t) = \log \hat{p}(\mathbf{s}_{t:t+n}|\mathbf{b}) - \log p(\mathbf{s}_{t:t+n}),$$

is the step at which the hypothesis that $\mathbf{s}_{t:t+n}$ is generated using $\mathbf{b}$ has the greatest advantage over the hypothesis that $\mathbf{s}_{t:t+n}$ is pure noise. This is equivalent to modeling the entire signal $\mathbf{s}$ as generated by Gaussian noise, but with the addition of $\mathbf{b}$ under some unknown scaling at an unknown time t. The log-likelihood under such a model,

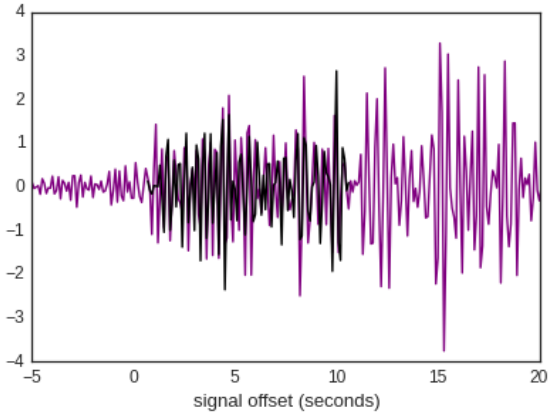
$$\log p(\mathbf{s}|t) = \log \mathcal{N}(\mathbf{s}_{0:t}; \mathbf{0}, \sigma^2\mathbf{I}) + \log \hat{p}(\mathbf{s}_{t:t+n}|\mathbf{b}) + \log \mathcal{N}(\mathbf{s}_{t+n:T}; \mathbf{0}, \sigma^2\mathbf{I}), \tag{B.5}$$

combined under Bayes' rule with a uniform prior $p(t) = \frac{1}{T}$, yields a posterior $p(t|\mathbf{s})$; it is easy to see that this posterior is proportional to $\exp(L(t))$.

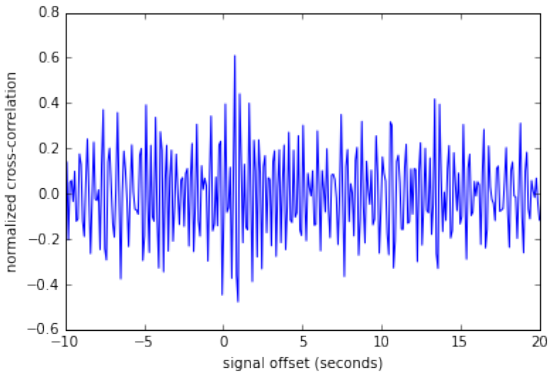
The Bayesian log odds (B.4) differ from the normalized correlation κ in that they also involve the amplitude $\|\mathbf{s}_{t:t+n}\|$ of each signal window. Under the assumption that the background noise is stationary, we would expect signal windows containing non-noise energy sources to have higher amplitudes than windows generated by pure noise; the log odds can be seen as a simple adjustment to the normalized correlation that incorporates this intuition. This is visible in Figure B.1, where the model-based methods assign relatively low scores to early alignments (<0s), which correspond to lower-amplitude signal windows.

B.2 General noise

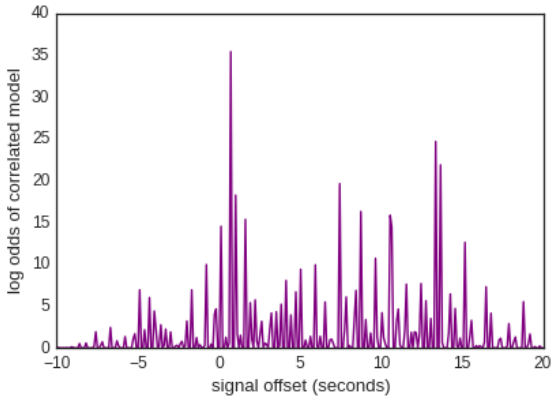
Extending the Bayesian formulation allows us to model explicit structure in the noise process. In this section we perform a similar analysis to the i.i.d. case above, but now assuming a general multivariate Gaussian noise process $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ with covariance matrix $\mathbf{R}$. Note that this includes autoregressive models, along with other state-space models, as special



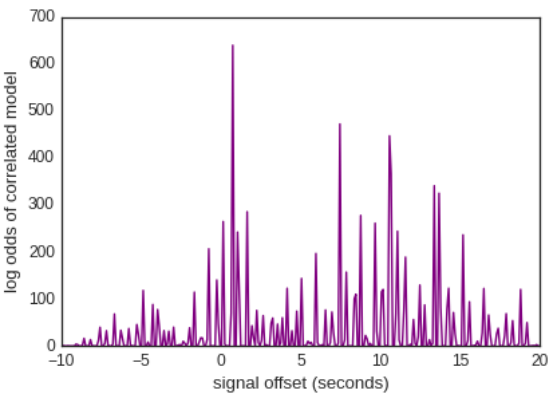
(a) Aligned waveforms **s** (purple) and **b** (black) showing strong correlation.



(b) Normalized cross-correlation trace from sliding **b** across **s**.



(c) Bayesian alignment log odds (B.4) under an i.i.d. Gaussian noise model.



(d) Bayesian alignment log odds (B.8) under an AR(1) noise model.

Figure B.1: Illustration comparing cross-correlation to a Bayesian alignment posterior. Signals are from doublet events, IMS evids 5334939 and 5335822, recorded at MKAR and filtered to 2.0-4.5Hz.

cases. Following a similar derivation as above,

$$\begin{aligned}\log p(\mathbf{a}|\mathbf{b};\alpha) &= \log \mathcal{N}(\mathbf{a};\alpha\mathbf{b},\mathbf{R}) \\ &= -\frac{1}{2}(\mathbf{a}-\alpha\mathbf{b})^T\mathbf{R}^{-1}(\mathbf{a}-\alpha\mathbf{b}) - \frac{1}{2}\log|\mathbf{R}| - \frac{n}{2}\log(2\pi) \\ &= -\frac{1}{2}\left(\mathbf{a}^T\mathbf{R}^{-1}\mathbf{a} - 2\alpha\mathbf{b}^T\mathbf{R}^{-1}\mathbf{a} + \alpha^2\mathbf{b}\mathbf{R}^{-1}\mathbf{b}\right) - \frac{1}{2}\log|\mathbf{R}| - \frac{n}{2}\log(2\pi).\end{aligned}$$

To estimate α by maximum likelihood, we similarly take the derivative

$$\frac{\partial \log p(\mathbf{a}|\mathbf{b};\alpha)}{\partial\alpha} = \mathbf{b}^T\mathbf{R}^{-1}\mathbf{a} - \alpha\mathbf{b}\mathbf{R}^{-1}\mathbf{b},$$

set to zero, and solve for α to find

$$\hat{\alpha} = \frac{\mathbf{b}^T\mathbf{R}^{-1}\mathbf{a}}{\mathbf{b}\mathbf{R}^{-1}\mathbf{b}}, \tag{B.6}$$

which as before is a global maximum. This suggests a generalization of the standard correlation,

$$\kappa_{\mathbf{R}} = \frac{\mathbf{b}^T\mathbf{R}^{-1}\mathbf{a}}{\sqrt{(\mathbf{b}\mathbf{R}^{-1}\mathbf{b})\,(\mathbf{a}\mathbf{R}^{-1}\mathbf{a})}}$$

defined with respect to an arbitrary noise process. Substituting as before, the log probability at the MLE is

$$\begin{aligned}\log \hat{p}_{\mathbf{R}}(\mathbf{a}|\mathbf{b}) &= -\frac{1}{2}\left(\mathbf{a}^T\mathbf{R}^{-1}\mathbf{a} - \left(\mathbf{b}^T\mathbf{R}^{-1}\mathbf{a}\right)^2/\mathbf{b}\mathbf{R}^{-1}\mathbf{b}\right) - \frac{1}{2}\log|\mathbf{R}| - \frac{n}{2}\log(2\pi) \\ &= -\frac{1}{2}\mathbf{a}^T\mathbf{R}^{-1}\mathbf{a}\left(1-\kappa_{\mathbf{R}}^2\right) - \frac{1}{2}\log|\mathbf{R}| - \frac{n}{2}\log(2\pi).\end{aligned} \tag{B.7}$$

This optimized log probability has the same relationship to the generalized correlation $\kappa_{\mathbf{R}}$ as the i.i.d. probability (B.3) had to the standard correlation κ. In particular, we can compute the log odds ratio,

$$\begin{aligned}\log \hat{p}_{\mathbf{R}}(\mathbf{a}|\mathbf{b}) - \log p_{\mathbf{R}}(\mathbf{a}) &= -\frac{1}{2}\mathbf{a}^T\mathbf{R}^{-1}\mathbf{a}\left(1-\kappa_{\mathbf{R}}^2\right) - \left(-\frac{1}{2}\mathbf{a}^T\mathbf{R}^{-1}\mathbf{a}\right) \\ &= \frac{1}{2}\mathbf{a}^T\mathbf{R}^{-1}\mathbf{a}\kappa_{\mathbf{R}}^2 \\ &= \frac{\left(\mathbf{b}^T\mathbf{R}^{-1}\mathbf{a}\right)^2}{2\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b}},\end{aligned} \tag{B.8}$$

analogously to the iid case. The general log odds ratio (B.8) plays a similar role to the i.i.d. log odds (B.4) as a model-based replacement for the normalized correlation κ, for example allowing us to define an alignment log-likelihood

$$\log p(\mathbf{s}|t) = \log \mathcal{N}(\mathbf{s}_{0:t};\mathbf{0},\mathbf{R}) + \log \hat{p}(\mathbf{s}_{t:t+n}|\mathbf{b}) + \log \mathcal{N}(\mathbf{s}_{t+n:T};\mathbf{0},\mathbf{R}), \tag{B.9}$$

analogous to (B.5), but now modeling structure in the noise process.

B.2.1 Computation

In many cases it is not convenient to work explicitly in terms of an explicit covariance matrix $\mathbf{R}$, but we may have access to the noise process via a function $f(\varepsilon) = \log p(\varepsilon)$ that efficiently computes the likelihood of ε under the noise distribution $\mathcal{N}(\mathbf{0}, \mathbf{R})$. This is the case, for example, with autoregressive processes, where the covariance and precision matrices are not straightforward to construct, but the likelihood calculation can be performed as an efficient, linear-time iteration over the signal. It turns out we can still compute the scale estimate $\hat{\alpha}$, generalized correlation $\kappa_{\mathbf{R}}$, and optimized likelihood (B.7) in this setting with only a little additional effort.

Given two signals $\mathbf{b}$, $\mathbf{a}$, write their log likelihoods under the noise process as

$$\begin{aligned} f(\mathbf{b}) &= -\frac{1}{2}\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b} - \frac{1}{2}\log|\mathbf{R}| - \frac{n}{2}\log 2\pi \\ &= -\frac{1}{2}\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b} + f(\mathbf{0}) \\ f(\mathbf{a}) &= -\frac{1}{2}\mathbf{a}^T\mathbf{R}^{-1}\mathbf{a} + f(\mathbf{0}), \end{aligned}$$

where $f(\mathbf{0})$ computes a normalizing constant. We can also write the likelihood of their difference,

$$\begin{aligned} f(\mathbf{b}-\mathbf{a}) &= -\frac{1}{2}(\mathbf{b}-\mathbf{a})^T\mathbf{R}^{-1}(\mathbf{b}-\mathbf{a}) + f(\mathbf{0}) \\ &= -\frac{1}{2}\left(\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b} - 2\mathbf{b}^T\mathbf{R}^{-1}\mathbf{a} + \mathbf{a}^T\mathbf{R}^{-1}\mathbf{a}\right) + f(\mathbf{0}) \\ &= f(\mathbf{b}) + f(\mathbf{a}) - f(\mathbf{0}) + \mathbf{b}^T\mathbf{R}^{-1}\mathbf{a}. \end{aligned}$$

By rearranging these quantities it is straightforward to compute the correlation statistics,

$$\hat{\alpha} = \frac{\mathbf{b}^T\mathbf{R}^{-1}\mathbf{a}}{\mathbf{b}\mathbf{R}^{-1}\mathbf{b}} = \frac{f(\mathbf{b}) + f(\mathbf{a}) - f(\mathbf{b}-\mathbf{a}) - f(\mathbf{0})}{2(f(\mathbf{b}) - f(\mathbf{0}))} \tag{B.10}$$

$$\kappa_{\mathbf{R}} = \frac{f(\mathbf{b}-\mathbf{a}) + f(\mathbf{0}) - f(\mathbf{b}) - f(\mathbf{a})}{2\sqrt{(f(\mathbf{b}) - f(\mathbf{0}))(f(\mathbf{a}) - f(\mathbf{0}))}} \tag{B.11}$$

$$\begin{aligned} \log\hat{p}_{\mathbf{R}}(\mathbf{a}|\mathbf{b}) &= (f(\mathbf{a}) - f(\mathbf{0}))\left(1 - \frac{f(\mathbf{b}-\mathbf{a}) + f(\mathbf{0}) - f(\mathbf{b}) - f(\mathbf{a})}{2\sqrt{(f(\mathbf{b}) - f(\mathbf{0}))(f(\mathbf{a}) - f(\mathbf{0}))}}\right) - f(\mathbf{0}) \\ &= f(\mathbf{a} - \hat{\alpha}\mathbf{b}), \end{aligned} \tag{B.12}$$

using only evaluations of the noise log likelihood f. This is the approach we use in practice to evaluate the autoregressive log odds (B.8).

B.2.2 Nonzero means

Here we consider a subtle point: when aligning signals under an AR noise model, we cannot simply compute the log odds (B.8) independently for each candidate alignment, as we did for

the i.i.d. (B.4), because the likelihood of the overall signal depends on timesteps preceding the current window $\mathbf{s}_{t:t+n}$, conditioned on which we can predict a nonzero noise mean for $\mathbf{s}_{t:t+n}$, as well as on timesteps following the current window, whose mean will be a function of the observed values for $\mathbf{s}_{t:t+n}$.

Rather than construct these means explicitly, we adjust the computations (B.10, B.11, B.12) for noise processes $\varepsilon \sim \mathcal{N}(\mathbf{c}, \mathbf{R})$ with nonzero mean $\mathbf{c}$, where we assume we have access to the log likelihood density

$$g(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\mathbf{c})^T\mathbf{R}^{-1}(\mathbf{x}-\mathbf{c}) - \frac{1}{2}\log|\mathbf{R}| - \frac{n}{2}\log 2\pi,$$

as well as its zero-mean counterpart

$$f(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T\mathbf{R}^{-1}\mathbf{x} - \frac{1}{2}\log|\mathbf{R}| - \frac{n}{2}\log 2\pi,$$

as above. This is straightforward in the AR case, where we can implicitly represent the nonzero mean by computing a likelihood g conditioned on past observations. Note the likelihood $g(\mathbf{a})$ under the nonzero-mean model is equal to $f(\mathbf{a}-\mathbf{c})$, so if we have explicit access to $\mathbf{c}$ we can simply subtract it from $\mathbf{a}$ and proceed as above. Otherwise, we replace f with g whenever it is applied to $\mathbf{a}$, giving the estimated amplitude

$$\hat{\alpha} = \frac{g(\mathbf{a}) + f(\mathbf{b}) - g(\mathbf{a}-\mathbf{b}) - f(\mathbf{0})}{2(f(\mathbf{b}) - f(\mathbf{0}))},$$

which can be verified by explicitly optimizing the likelihood $g(\mathbf{a}-\alpha\mathbf{b})$. The same approach also yields nonzero-mean counterparts to B.11 and B.12.

B.3 Discussion

Like the normalized correlation κ, the statistics $\hat{p}$ and $\hat{p}_{\mathbf{R}}$ derived in this section do not impose a preferred amplitude α; instead they optimize to find the best scale for each candidate alignment (as noted above, this has the property of preferring alignments for which large values of α can explain significant energy in the signal). This is mathemematically convenient but unsatisfying from a modeling perspective, since our statistics no longer correspond to probabilities under any particular generative model. Ideally we would integrate over α with respect to some prior on the likely amplitude of the signal being detected (as is done in the full SIGVISA model). This would eliminate certain pathological behavior; for example, under the models considered here we cannot query the probability that $\mathbf{b}$ is missing entirely from some signal $\mathbf{s}$, since we cannot distinguish this case from the case where $\mathbf{b}$ is present with $\alpha = 0$. Integrating over α would cause a proper Bayesian analysis to impose a complexity penalty for the latter case, preferring the simpler pure-noise model when that model fits the data. It is not obvious that this can be done in closed form, though this is an interesting question for future work. If possible, such a statistic would represent a further departure from the scale-agnostic properties of normalized correlation; whether this is desirable would depend on the application being considered.

Bibliography

Abramovich, Felix, Theofanis Sapatinas, and Bernard W Silverman (1998). "Wavelet Thresholding via a Bayesian Approach". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 60.4, pp. 725–749.

Adler, RJ (1981). *The Geometry of Random Fields*. Chichester: Wiley.

Aki, Keiiti (1969). "Analysis of the Seismic Coda of Local Earthquakes as Scattered Waves". In: *Journal of geophysical research* 74.2, pp. 615–631.

Aki, Keiiti and Paul G Richards (1980). *Quantitative Seismology*. San Francisco: W.H. Freeman and Co.

Andrieu, Christophe et al. (2003). "An Introduction to MCMC for Machine Learning". In: *Machine learning* 50.1-2, pp. 5–43.

Anstey, Nigel Allister (1964). "Correlation Techniques—A Review". In: *Geophysical Prospecting* 12.4, pp. 355–382.

Arora, Nimar, Stuart Russell, and Erik Sudderth (2013). "NET-VISA: Network Processing Vertically Integrated Seismic Analysis". In: *Bulletin of the Seismological Society of America* 103.2A, pp. 709–729.

Arora, Nimar, Dmitry Storchak, and Stuart Russell (2015). "Using ISC Data to Build a Prior of Seismicity for NET-VISA". CTBT: Science and Technology 2015 Conference (SnT2015) presentation. URL: https://www.ctbto.org/fileadmin/user_upload/SnT2015/SnT2015_Orals/T3.3-O9.pdf.

Ballard, Sandy et al. (2015). "Improved Bulletin Generation using an Iterative Processing Framework". CTBT: Science and Technology 2015 Conference (SnT2015) presentation. URL: https://www.ctbto.org/fileadmin/user_upload/SnT2015/SnT2015_Posters/T3.3-P16.pdf.

Billingsley, Patrick (2008). *Probability and Measure*. John Wiley & Sons.

Bishop, Christopher M (2006). *Pattern Recognition and Machine Learning*.

Borges, Jorge Luis (1998). "Collected Fictions". In: trans. by Andrew Hurley. Viking New York. Chap. On Exactitude in Science.

Box, George EP (1976). "Science and Statistics". In: *Journal of the American Statistical Association* 71.356, pp. 791–799.

Bras, Ronan Le et al. (1994). *Global Association*. Tech. rep. ADA304805. San Diego, CA: Science Applications International Corp. URL: http://handle.dtic.mil/100.2/ADA304805.

Brooks, Steve et al. (2011). *Handbook of Markov Chain Monte Carlo*. CRC press.

Brune, James N (1970). "Tectonic Stress and the Spectra of Seismic Shear Waves from Earthquakes". In: *Journal of Geophysical Research* 75.26, pp. 4997–5009.

Carpenter, Bob et al. (2016). "Stan: A Probabilistic Programming Language". In: *Journal of Statistical Software*.

CTBTO (2008). *Monitoring technologies: Seismic Monitoring*. Tech. rep. Accessed: 2016-11-21. URL: https://www.ctbto.org/verification-regime/monitoring-technologies-how-they-work/seismic-monitoring/.

— (2015). *The CTBT Verification Regime: Monitoring the Earth for nuclear explosions*. Tech. rep. Accessed: 2016-11-21. URL: https://www.ctbto.org/fileadmin/user_upload/public_information/2015/Verification_Regime_final_2015_final.pdf.

Cua, Georgia B (2005). "Creating the Virtual Seismologist: Developments in Ground Motion Characterization and Seismic Early Warning". PhD thesis. California Institute of Technology.

Daubechies, Ingrid (1992). "Ten Lectures on Wavelets". In: *Regional conference Series in Applied Mathematics (SIAM)*. Philadelphia.

Dempster, Arthur P, Nan M Laird, and Donald B Rubin (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38.

Doob, Joseph L. (1953). *Stochastic processes*. John Wiley & Sons.

Duda, Richard O and Peter E Hart (1972). "Use of the Hough Transformation to Detect Lines and Curves in Pictures". In: *Communications of the ACM* 15.1, pp. 11–15.

Duvenaud, David (2014). "Automatic Model Construction with Gaussian Processes". PhD thesis. University of Cambridge.

Geiger, Ludwig (1912). "Probability method for the determination of earthquake epicenters from the arrival time only". In: *Bull. St. Louis Univ* 8.1, pp. 56–71.

Gelman, Andrew et al. (2014). *Bayesian Data Analysis*. Vol. 2. Chapman & Hall/CRC Boca Raton, FL, USA.

Gibbons, Steven J and Frode Ringdal (2006). "The Detection of Low Magnitude Seismic Events using Array-Based Waveform Correlation". In: *Geophysical Journal International* 165.1, pp. 149–166.

— (2012). "Seismic Monitoring of the North Korea Nuclear Test Site using a Multichannel Correlation Detector". In: *IEEE Transactions on Geoscience and Remote Sensing* 50.5, pp. 1897–1909.

Gorbachev, Mikhail (2010). "The Senate's Next Task: Ratifying the Nuclear Test Ban Treaty". In: *The New York Times*.

Grewal, Mohinder S and Angus P Andrews (2014). *Kalman Filtering: Theory and Practice with MATLAB*. John Wiley & Sons.

Gutenberg, Beno and Charles Richter (1954). *Seismicity of the Earth and Associated Phenomena*. New Jersey: Princeton University Press.

Harris, David B (1997). *Waveform Correlation Methods for Identifying Populations of Calibration Events*. Tech. rep. Lawrence Livermore National Lab., CA (United States).

Hastie, David I and Peter J Green (2012). "Model Choice Using Reversible Jump Markov chain Monte Carlo". In: *Statistica Neerlandica* 66.3, pp. 309–338.

Helmberger, DV (1983). "Theory and application of synthetic seismograms". In: *Earthquakes: observation, theory and interpretation* 37, pp. 173–222.

Horn, Roger A and Charles R Johnson (2012). *Matrix Analysis*. Cambridge University Press.

Jurkevics, Andy (1988). "Polarization Analysis of Three-Component Array Data". In: *Bulletin of the Seismological Society of America* 78.5, pp. 1725–1743.

Kanamori, Hiroo (2001). "Energy budget of earthquakes and seismic efficiency". In: *International Geophysics* 76, pp. 293–305.

Kanamori, Hiroo and TC Hanks (1979). "A moment magnitude scale". In: *J. Geophys. Res* 84, pp. 2348–2349.

Kanasewich, Ernest R (1981). *Time Sequence Analysis in Geophysics*. University of Alberta.

Kelly, KR et al. (1976). "Synthetic Seismograms: A Finite-Difference Approach". In: *Geophysics* 41.1, pp. 2–27.

Kennett, BLN and ER Engdahl (1991). "Traveltimes for Global Earthquake Location and Phase Identification". In: *Geophysical Journal International* 105.2, pp. 429–465.

Koller, Daphne and Nir Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Komatitsch, Dimitri and Jean-Pierre Vilotte (1998). "The Spectral Element Method: An Efficient Tool to Simulate the Seismic Response of 2D and 3D Geological Structures". In: *Bulletin of the seismological society of America* 88.2, pp. 368–392.

Krige, D (1951). "A Statistical Approach to some Basic Mine Valuation Problems on the Witwatersrand". In: *Journal of Chemical, Metallurgical, and Mining Society of South Africa*.

Le Bras, Ronan, Jan Wuster, and Science Applications International Corporation (2002). *IDC Processing of Seismic, Hydroacoustic, and Infrasonic Data*. Tech. rep. IDC5.2.1Rev1. IDC Documentation Series.

Lindley, Dennis Victor (1972). *Bayesian Statistics: A Review*. SIAM.

MacKay, David JC (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge university press.

Magotra, N, N Ahmed, and E Chael (1987). "Seismic event detection and source location using single-station (three-component) data". In: *Bulletin of the Seismological Society of America* 77.3, pp. 958–971.

Mahalanobis, Prasanta Chandra (1936). "On the Generalized Distance in Statistics". In: *Proceedings of the National Institute of Sciences (Calcutta)* 2, pp. 49–55.

Mallat, Stephane (1999). *A Wavelet Tour of Signal Processing*. Academic Press.

Mansinghka, Vikash, Daniel Selsam, and Yura Perov (2014). "Venture: A Higher-Order Probabilistic Programming Platform with Programmable Inference". In: *arXiv preprint arXiv:1404.0099*.

Marshall, PD and PW Basham (1972). "Discrimination between earthquakes and underground explosions employing an improved Ms scale". In: *Geophysical Journal International* 28.5, pp. 431–458.

Maybeck, Peter S (1982). *Stochastic models, estimation, and control*. Academic Press. Chap. 12.7.

Mayeda, Kevin (1993). "mb (LgCoda): A Stable Single Station Estimator of Magnitude". In: *Bulletin of the Seismological Society of America* 83.3, pp. 851–861.

Mayeda, Kevin et al. (2003). "Stable and Transportable Regional Magnitudes Based on Coda-derived Moment-rate Spectra". In: *Bulletin of the Seismological Society of America* 93.1, pp. 224–239.

Milch, Brian and Stuart Russell (2010). "Extending Bayesian Networks to the Open-Universe Case". In: *Heuristics, Probability and Causality: A Tribute to Judea Pearl*.

Milch, Brian et al. (2005). "BLOG: Probabilistic Models with Unknown Objects". In: *Proceedings of the 19th international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 1352–1359.

Minka, Thomas P (2001). "Expectation Propagation for Approximate Bayesian Inference". In: *Proceedings of Uncertainty in Artificial Intelligence (UAI)*. Morgan Kaufmann Publishers Inc., pp. 362–369.

Moore, David and Stuart Russell (2015). "Gaussian Process Random Fields". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 3357–3365.

Mueller, Richard A and John R Murphy (1971). "Seismic Characteristics of Underground Nuclear Detonations Part I. Seismic Spectrum Scaling". In: *Bulletin of the Seismological Society of America* 61.6, pp. 1675–1692.

Myers, Stephen C, Gardar Johannesson, and William Hanley (2007). "A Bayesian Hierarchical Method for Multiple-event Seismic Location". In: *Geophysical Journal International* 171.3, pp. 1049–1063.

Pasyanos, Michael E, William R Walter, and Kevin M Mayeda (2012). "Exploiting Regional Amplitude Envelopes: A Case Study for Earthquakes and Explosions in the Korean Peninsula". In: *Bulletin of the Seismological Society of America* 102.5, pp. 1938–1948.

Rasmussen, Carl and Chris Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Roberts, RG, A Christoffersson, and F Cassidy (1989). "Real-time event detection, phase identification and source location estimation using single station three-component seismic data". In: *Geophysical Journal International* 97.3, pp. 471–480.

Ruggeri, Fabrizio and Brani Vidakovic (2005). "Bayesian Modeling in the Wavelet Domain". In: *Handbook of Statistics* 25, pp. 315–338.

Sato, Haruo, Michael C Fehler, and Takuto Maeda (2012). *Seismic Wave Propagation and Scattering in the Heterogeneous Earth*. Vol. 496. Springer.

Schaff, David P, Won-Young Kim, and Paul G Richards (2012). "Seismological Constraints on Proposed Low-Yield Nuclear Testing in Particular Regions and Time Periods in the Past". In: *Science and Global Security* 20.2-3, pp. 155–171.

Schaff, David P and Paul G Richards (2004). "Repeating Seismic Events in China". In: *Science* 303.5661, pp. 1176–1178.

— (2011). "On Finding and Using Repeating Seismic Events in and near China". In: *Journal of Geophysical Research: Solid Earth* 116.B3.

Schaff, David P and Felix Waldhauser (2010). "One Magnitude Unit Reduction in Detection Threshold by Cross Correlation Applied to Parkfield (California) and China Seismicity". In: *Bulletin of the Seismological Society of America* 100.6, pp. 3224–3238.

Schaff, David P et al. (2004). "Optimizing Correlation Techniques for Improved Earthquake Location". In: *Bulletin of the Seismological Society of America* 94.2, pp. 705–721.

Shearer, Peter M (2009). *Introduction to Seismology*. Cambridge University Press.

Shumway, Robert H and David S Stoffer (2010). *Time Series Analysis and its Applications: with R examples*. Springer Texts in Statistics.

Simmons, Nathan A et al. (2012). "LLNL-G3Dv3: Global P Wave Tomography Model for Improved Regional and Teleseismic Travel Time Prediction". In: *Journal of Geophysical Research: Solid Earth* 117.B10.

Slinkard, Megan E, Dorthe B Carr, and Christopher J Young (2013). "Applying Waveform Correlation to Three Aftershock Sequences". In: *Bulletin of the Seismological Society of America* 103.2A, pp. 675–693.

Smith, Kenneth et al. (2011). "Preliminary Analysis of the Mw 6.0 Wells, Nevada, Earthquake Sequence". In: *Nevada Bureau of Mines and Geology Special Publication* 36, pp. 127–145.

Stein, Seth and Michael Wysession (2009). *An Introduction to Seismology, Earthquakes, and Earth Structure*. John Wiley & Sons.

Storchak, Dmitry A, Johannes Schweitzer, and Peter Bormann (2003). "The IASPEI Standard Seismic Phase List". In: *Seismological Research Letters* 74.6, pp. 761–772.

Storvik, Geir (2011). "On the Flexibility of Metropolis–Hastings Acceptance Probabilities in Auxiliary Variable Proposal Generation". In: *Scandinavian Journal of Statistics* 38.2, pp. 342–358.

US Energy Information Administration (2016). *Annual Coal Report*. Tech. rep.

Utsu, T (1961). "A Statistical Study on the Occurrence of Aftershocks." In: *Geophysical Magazine* 30.4.

Von Mises, Richard (1964). *Mathematical Theory of Probability and Statistics*. Academic Press.

Waldhauser, Felix and William L Ellsworth (2000). "A Double-Difference Earthquake Location Algorithm: Method and Application to the Northern Hayward Fault, California". In: *Bulletin of the Seismological Society of America* 90.6, pp. 1353–1368.

Waldhauser, Felix and David P Schaff (2008). "Large-scale Relocation of Two Decades of Northern California Seismicity using Cross-Correlation and Double-Difference Methods". In: *Journal of Geophysical Research: Solid Earth* 113.B8.

Wang, Wei and Stuart Russell (2015). "A Smart-Dumb/Dumb-Smart Algorithm for Efficient Split-Merge MCMC". In: *Proceedings of Uncertainty in Artificial Intelligence (UAI)*.

Wiener, Norbert (1949). *Extrapolation, interpolation, and smoothing of stationary time series*. Cambridge: MIT Press.

Withers, Mitchell et al. (1998). "A Comparison of Select Trigger Algorithms for Automated Global Seismic Phase and Event Detection". In: *Bulletin of the Seismological Society of America* 88.1, pp. 95–106.

Wu, Yi et al. (2016). "Swift: Compiled Inference for Probabilistic Programming Languages". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.