
Bayesian Inference for the Community Seismic Network

Rishi Chandy

Arthur R. Adams SURF Fellow

Mentors: Andreas Krause and K. Mani Chandy

Abstract

The California Integrated Seismic Network uses a large array of expensive seismometers that measure ground motion. However, the network is unable to provide reliable real-time source estimation due to the sparseness of the sensors. The Community Seismic Network project at Caltech plans to create a dense seismic network by leveraging small, inexpensive accelerometers attached to personal computers and embedded in mobile phones. This network would be capable of reporting real-time acceleration data to central servers for analysis. With this data, the network can issue early warning alerts which would minimize widespread suffering and economic losses. New methods resulting from this research aim to estimate earthquake source and magnitude in real-time by leveraging the unique distributed aspects of the Community Seismic Network.

1 Introduction

Earthquakes pose a constant threat to society; a dependable early warning system could minimize widespread suffering and economic losses. Just a few seconds of advance warning could allow the activation of safeguards in critical operations, such as trains, industrial machinery, and “active response systems” in modern skyscrapers[1].

By creating a network of seismometers, sensors that record seismic waveforms, ground acceleration data can be used to quickly estimate the source and magnitude of an earthquake. Advances in digital communication and computer processing have allowed early warning systems to be implemented in several locations, starting with the Japanese train system in the 1960s, leading to installations in Italy, Taiwan and Mexico[2].

The California Integrated Seismic Network (CISN) uses a large array of seismometers that measure ground motion. These seismometers are very accurate but expensive, thus the cost of implementing such a system is prohibitive in developing countries. In addition, the network is unable to provide reliable real-time source estimation due to the sparseness of the sensors. The Community Seismic Network (CSN) project at Caltech aims to remedy these issues by leveraging small, inexpensive accelerometers attached to personal computers to create a dense seismic network. This network would be capable of reporting real-time acceleration data for server-side analysis on Google’s App Engine cloud computing platform.

Although the network will be dense, it will also be noisy due to the low resolution of the many different types of sensors, including mobile phone accelerometers and USB devices. Since the Community Seismic Network differs from a traditional network in so many ways, fundamentally different server-side algorithms, such as Bayesian learning[4], are necessary to analyze incoming sensor data.

2 System architecture

2.1 Client operation

After installing the desktop or mobile phone application, the user specifies the current sensor location or, in the case of the mobile application, the location is automatically determined using GPS. This information, along with other metadata such as sensor model, is sent to the servers as part of the registration message. An interface shows live tri-axial acceleration waveforms from the attached accelerometer. Even after the interface is closed, the client software continues to read acceleration data into a ring buffer in the background. During an earthquake, primary waves (P-waves) have a significant lead time over damaging secondary waves (S-waves). Once a P-wave is detected, the client immediately triggers a “pick” and sends the picked waveform data to the server. Using the standard short-term average over long-term average acceleration method (STA/LTA) method, the client sends a pick message when STA/LTA exceeds a certain tunable threshold, indicating the beginning of significant seismic activity. This message contains waveform data recorded from a window of time before the trigger.

Clients periodically send “heartbeat” messages containing their current location, picking algorithm parameter set version, and new waveform data so that the servers can maintain an accurate catalog of active sensors. The heartbeat response currently includes any new parameter updates, and the protocol also allows for requesting waveform data from a certain time window as part of log request. These log requests are useful for archiving and analyzing more comprehensive data some time after a verified earthquake. Waveform data is encoded in the Seismic Analysis Code (SAC) format[3].

2.2 Server operation

Clients and servers communicate through an open and extensible XML schema. The server application runs on Google App Engine platform for scalable cloud computing in Java. As pick and heartbeat messages arrive, the server updates its catalog of active sensors and returns appropriate parameter updates through the heartbeat response. CSN administrators can specify parameter updates for specific clients or categories of clients based on historical data and evaluate performance.

The web interface displays a near real-time heatmap which visualizes recent picks reported by sensors in the network and colors them according to relative acceleration magnitude. The map also shows location and limited metadata from active clients.

Preliminary approximations of possible earthquake source and magnitude are iteratively improved as more picks are received. Once the prediction reaches a high level of certainty, early warning alerts are sent to areas that will soon be affected by extreme ground motion.

3 Datastore API

The Datastore API was developed to allow researchers to access the datastore without worrying about Google App Engine implementation details. It provides access to all properties of client, pick, heartbeat, and parameter update objects, and is easily extensible to any new datatypes created in the future.

Users send query parameters as part of an HTTP request to the API URL, which returns results in XML. API queries can contain equality and inequality operators on almost all properties of supported objects. In addition, by specifying a rectangular bounding box, queries can request objects, such as clients and picks, located in region. The API also supports returning only a user-specified subset of the overall resultset through the use of continuation tokens, thus allowing users to retrieve manageable chunks of data.

4 Earthquake detection

The fault-tolerant algorithms for earthquake parameter estimation and early warning aim to leverage the unique aspects of the heterogeneous distributed CSN sensor network. Google’s App Engine cloud infrastructure was chosen to decentralize the data processing, allowing the network to be

more robust in the event of a powerful earthquake. However, this scalability and performance comes at the price of several restrictions, such as computation time limits, that must be taken into account.

4.1 Bayesian approach

In a straightforward implementation (Figure 1), the prior information consists of structural properties of the sensor network, historical earthquake trends for the region, and the Gutenberg-Richter Law. The likelihood function would be based on a normal distribution of pick arrival times.

$$P(I, L, T|D) = \frac{P(D|I, L, T)P(I, L, T)}{P(D)}$$

Figure 1: Bayesian inference, with I = earthquake intensity, L = location, T = earthquake start time, D = pick data

4.2 Heuristic approach

The heuristic methods (Figure 2) aim to leverage the properties of the dense CSN network to overcome the noisy data from individual sensors. By overlaying a grid on the earth, every location can be assigned to a set of grid cells of differing resolution (i.e. “zoom” level). As each pick arrives at the server, a counter for the corresponding cell is incremented. If the sensors in a cell become very active in a short period of time, then that cell is flagged for further inspection. A background process periodically examines the regions neighboring flagged cells at different resolutions for further evidence of a seismic event. The thresholds for flagging a cell and the configuration of neighboring cells are sensitive to tuning.

4.2.1 Evaluation framework

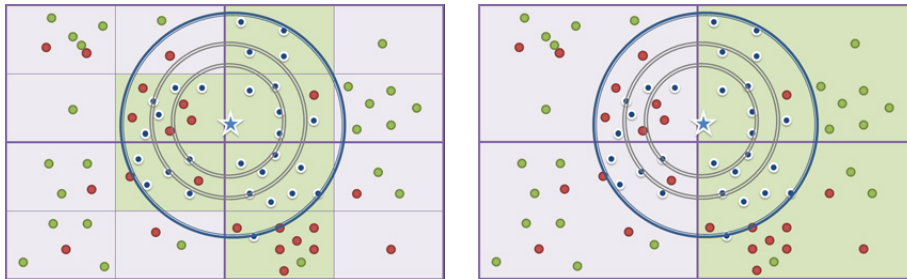


Figure 2: The heuristic approach uses ratios of picked (blue) to active (green) sensors to detect earthquakes. Inactive sensors are red. The left diagram shows zoom level 0 units cells, while the right diagram shows zoom level 1 cells.

A Java framework for simulating seismic events was developed to assess the feasibility of the heuristic approaches. First, the grid configuration application allows users to specify grid dimensions, composed of square units cells of area 25 km², and total number of sensors. The application then generates a list of uniformly distributed sensors compatible with the synthetic data generator. Next, the synthetic data generator creates SAC files for each sensor, and picks are extracted from these SAC files and input to the evaluation framework. The application simulates the pick messages from each sensor as the earthquake wavefront propagates, and flags cells when the majority of sensors in that cell have picked recently. A binary prediction is generated at each timestep.

5 Future directions

With the framework completed, future work involves evaluating the effect of varying sensor distributions on the true positive rate, while simultaneously minimizing false positives. Having accurate and

consistent timestamps across the network is important, so future work will allow clients to synchronize clocks using Network Time Protocol (NTP). Also, future work will implement XML message signing and verification to ensure the security of the network.

Acknowledgments

The author would like to thank Professors K. Mani Chandy and Andreas Krause for their valuable mentorship, Julian Bunn, Matthew Faulkner, and Michael Olson for their helpful suggestions and feedback, and Jonathan Krause and Daniel Rosenberg for their hard work and cooperation. He would also like to thank the Arthur R. Adams Foundation for its generous donation and sponsorship.

References

- [1] R.M. Allen and H. Kanamori, "The potential for earthquake early warning in southern California," *Science*, vol. 300, 2003, pp. 786-9.
- [2] A. Zollo et al., "Earthquake early warning system in southern Italy: Methodologies and performance evaluation," *Geophysical Research Letters*, vol. 36, 2009, pp. 1-6.
- [3] P. Goldstein et al., "SAC2000: Signal processing and analysis tools for seismologists and engineers," Invited contribution to The IASPEI International Handbook of Earthquake and Engineering Seismology, Edited by WHK Lee, H. Kanamori, P.C. Jennings, and C. Kisslinger, Academic Press, 2003.
- [4] Chang Kook Oh, James L. Beck, and Masumi Yamada, J, "Bayesian Learning Using Automatic Relevance Determination Prior with an Application to Earthquake Early Warning," *Engrg. Mech.* 134, 1013 (2008).