Locality Sensitive Hashing for Scalable Earthquake Detection

Kexin Rong, Clara Yoon, Karianne Bergen, Hashem Elezabi, Peter Bailis, Philip Levis, Gregory Beroza
Stanford University
This talk

Background and Problem Setup:
Detect low-magnitude earthquakes from continuous seismic data

Scalability Challenges and Optimizations:
For fingerprint extraction, similarity search, post-processing

Qualitative Results:
Scientific discoveries near the Diablo Canyon nuclear power plant

Takeaways and applications beyond seismology:
IoT monitoring, server analytics, data exploration
What is an earthquake?
Earthquake Detection: the foundation of observational seismology

seismograph

1 day of seismic data
Seismic station in action
Challenge: Seismology has a lot of data

- Large and growing data volumes:
  
  Big Networks (Large-N)
  1000’s of sensors

Seismic Stations in Southern California
Challenge: Seismology has a lot of data

- Large and growing data volumes:
  - Big Networks (Large-N)
    - 1000’s of sensors
  - Long Duration (Large-T)
    - >10 years continuous data

1 day of seimsic data
Challenge: Many Undetected Small Earthquakes

- Earthquake catalog: ~100K events per year
- Over 1.5 million (est.) earthquakes are not detected by conventional means per year
Why Small Earthquakes?

- Uncover unknown seismic sources
- Understand earthquake mechanisms
- Predict future events

75 earthquakes
$1.2 < M_L < 2.9$

14,604 earthquakes
$-1.5 < M_L < 2.9$

Yoon et al. (2017)
Key observation: waveform similarity

Reoccurring earthquakes show near identical waveforms

Yoon et al. (2015)
Problem Setup

Detect *low magnitude* earthquakes from continuous seismic data

*Unsupervised*, purely based on physical properties of data

*Single-machine*, main-memory execution on commodity servers
This talk

Background and Problem Setup:
Detect low-magnitude earthquakes from continuous seismic data

Scalability Challenges and Optimizations:
For fingerprint extraction, similarity search, post-processing

Qualitative Results:
Scientific discoveries near the Diablo Canyon nuclear power plant

Takeaways and applications beyond seismology:
IoT monitoring, server analytics, data exploration
Context of the Collaboration

3 months of continuous data
5 days per channel on a single processor

Yoon et al. (2015)
Bergen et al. (2016)
Yoon et al. (2017)
**Fingerprint Overview**

**Input** Continuous ground motion measurement (e.g. 1 month of 100Hz time series data)

**Output** Binary vectors for each time series segment

---

**Diagram:**
- **Fingerprint Extraction**
- **Similarity Search (LSH)**
- **Spatiotemporal Alignment**
Fingerprints preserve waveform similarity

\[
\text{Correlation}(a, b) = \frac{a \cdot b}{\|a\| \|b\|}
\]

\[
\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}
\]
Fingerprinting Algorithm

Advantages of fingerprints:
- Incorporate frequency domain features
- More robust to noise

Bergen et al. (2016)
Fingerprinting Algorithm Implementation

Runtime grows mostly linearly with input size
20min to process 1 month of 100Hz data

Parallelization bottleneck: MAD
Solution: Sampling

<table>
<thead>
<tr>
<th>Sampling Rate (%)</th>
<th>Accuracy (%)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>99.5</td>
<td>10.5x</td>
</tr>
<tr>
<td>1</td>
<td>98.7</td>
<td>99.8x</td>
</tr>
<tr>
<td>0.1</td>
<td>94.9</td>
<td>350x</td>
</tr>
</tbody>
</table>
Similarity Search Overview

Input: Binary fingerprints

Output: Pairs of similar binary fingerprints

Fingerprint 5 is similar to Fingerprint 18
Locality Sensitive Hashing Overview

Hashes items into low-dimensional space such that similar items have a higher collision probability in the hash tables\(^1\)

Used in text and image search, near duplicate detection etc.

Sublinear linear query time (asymptotic improvement from naïve \(O(n)\) query time)

\(^1\)A. Gionis, P. Indyk, et al. (1999). Similarity search in high dimensions via hashing.
Scalability remains a challenge

- Runtime grows near quadratically with input size

1 month of data: 10 minutes
5 month of data: 5 hours

Can not detect earthquakes that occur only once in the data; scaling to longer datasets are important for scientific discoveries.
LSH Performance and Average Bucket size

Worst case: $O(n^2)$

Best case: $O(n)$

Large hash buckets slow down query performance
Challenge: Correlation in Input

\[ \mathbb{P}(a_i = 1, a_j = 1) \gg \mathbb{P}(a_i = 1) \mathbb{P}(a_j = 1) \]

Fingerprints reflect physical correlations in the data.

Increase the average hash bucket sizes.
Challenge: Repeating noise patterns

Not all ground motion is due to earthquakes

- Creates large LSH hash buckets that significantly increases the number of lookups per query
- Matches of noise dominates matches of earthquakes
Solution: apply domain priors

Frequency bands of earthquakes

Results

Up to 16x improvement in runtime
Also improved detection recall
Solution: apply domain priors

Frequency of occurrences

- Earthquakes are rare events
- Probably not going to reoccur 1000 times per day
- During similarity search, filter out fingerprints that generate too many near neighbors

Results

<table>
<thead>
<tr>
<th>Threshold</th>
<th>FP Rate (%)</th>
<th>Filtered Input (%)</th>
<th>Runtime (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>0</td>
<td>0.1</td>
<td>149.3</td>
</tr>
<tr>
<td>1%</td>
<td>0</td>
<td>30.1</td>
<td>31.0</td>
</tr>
</tbody>
</table>

Also decreased output size by over 50x
LSH Performance Sensitive to Parameters

Near identical detection probability

10x difference in runtime
Spatiotemporal Alignment Overview

**Input** Sparse similarity matrix

**Output** Groups in the matrix corresponding to potential earthquake detections
Challenge: Large LSH output

- LSH Output: 12.4 TB
- Channel Level: 926 GB
- Station Level: 119 GB
- Network Level: 5048 detections

Partition and Parallelization
Earthquake Events ↔ Thin Diagonals

Earthquake waveforms

Noise waveforms

Bergen and Beroza (2018)
**Spatiotemporal Alignment Overview**

**Channel Level**

<table>
<thead>
<tr>
<th>Channel 1</th>
<th>Channel 2</th>
<th>Channel 3</th>
</tr>
</thead>
</table>

Combine similarity matrix from channels of the same station

**Station Level**

- Station A
- Station B
- Station N

Cluster entries of the similarity matrix into thin diagonals

**Network Level**

Find clusters with the same interevent time (on the same diagonal) across stations

*Bergen and Beroza (2018)*
Solution: Out-of-core Implementation

Similarity matrix format:

\[(fingerprint1, fingerprint2, similarity)\]

\[(dt, fingerprint1, similarity)\]

Sort non-zero entries by diagonal \((dt)\)

- Combine similarity matrix reduces to an external merge sort + reduction
- Easy partition along the diagonals
- Bounding box summarizations requires a sequential pass
Effect of Optimizations

Enabled over 100x speedups in the end-to-end detection
Observations

Parallelization gets us pretty far
Near linear scalability across the board

Tradeoff between runtime and accuracy
Fingerprints: 99.8x speedup with 98.7% accuracy
LSH: 60-200x speedup with 6% false negative rate

Pushing domain priors
Improves both performance and quality of results
Area of interest: Diablo Canyon nuclear power plant

The power plant is surrounded by fault lines unknown when the facility was first proposed.

Overall low local earthquake activities: **535** catalog earthquake events from 2007 to 2017.
Overview of detected earthquakes

10 years of continuous time series data (11 stations, 27 channels)

Detected

3957 catalog earthquakes
597 new local earthquakes
Result: 355 new earthquakes

Magnitude -0.2 to 2.4

66.3% of existing catalog

Confirms that faults near the power plant are active; further seismic analysis will help resolve fault structures
Takeaways from the collaboration

Optimization under constraints
   How to make existing algorithms faster, instead of inventing new algorithms

Domain knowledge to the rescue
   Especially useful in cases when labels/ground truth is limited

Scalability is key
   Improved scalability can lead to new scientific discoveries
Broader Impact

• Source code available at https://github.com/stanford-futuredata/quake

• Complement existing earthquake detection method

• Contribute to label generation for potential supervised earthquake detection method
This talk and beyond

Unsupervised earthquake detection pipeline through scalable time series similarity search

Pushing domain constraints into the pipeline can improve both the efficiency and quality of the detection

What’s the right interface for specifying the priors for time series?

Applications beyond seismology
IoT monitoring, server analytics, data exploration

Has this event happened before?