Automating the Detection of Dynamically Triggered Earthquakes via a Deep Metric Learning Algorithm

Vivian Tang*[1], Prem Seetharaman[2], Kevin Chao[1], Bryan A. Pardo[2], and Suzan van der Lee[1]

Abstract

Detecting subtle signals from small earthquakes triggered by transient stresses from the surface waves of large magnitude earthquakes can contribute to a more general understanding of how earthquakes nucleate and interact with each other. However, searching for signals from such small earthquakes in thousands of seismograms is overwhelming, and discriminating them from a miscellany of noise is challenging. Here, we explore how we can automate the detection of such dynamically triggered earthquakes using a simple, diagnostic signal-to-noise ratio (SNR) threshold as well as a convolutional deep metric learning network. Our analysis shows that the deep learning network was more reliable at detecting small earthquakes than the SNR method.

Cite this article as Tang, V., P. Seetharaman, K. Chao, B. A. Pardo, and S. van der Lee (2020). Automating the Detection of Dynamically Triggered Earthquakes via a Deep Metric Learning Algorithm, *Seismol. Res. Lett.* **91**, 901–912, doi: 10.1785/0220190165.

Supplemental Material

Introduction

Knowledge of dynamically triggered seismic events provides an additional and unique perspective to our larger goal of understanding geologic faulting processes and seismic activity (Freed, 2005; Brodsky and van der Elst, 2014; Hill and Prejean, 2015). During their propagation, surface waves from strong or large magnitude earthquakes generate transient stresses and strains that can trigger slip on faults thousands of kilometers from their epicenters. In some cases, a slip is accompanied by tectonic tremors (Chao *et al.*, 2013; Chao and Obara, 2016), and in other cases one or more earthquakes are triggered (Fig. 1) (Velasco *et al.*, 2008; Gonzalez-Huizar *et al.*, 2012; Aiken and Peng, 2014), sometimes up to days later (Van der Elst *et al.*, 2013).

Although the size of these dynamic stresses has been implicated in the occurrence of dynamically triggered tremor (Peng and Gomberg, 2010), no simple relation appears to exist between the size of dynamic stress and the triggering of an earthquake (Wang *et al.*, 2018). Some triggered or induced earthquakes, however, have been found to correlate not with the size of dynamic stresses but with the state or rate of relatively static stresses such as increasing pore pressure (e.g., Van der Elst and Brodsky, 2010; Weingarten *et al.*, 2015; Quinones *et al.*, 2018). Therefore, finding instances of dynamically triggered earthquakes with methods that rely only on the peak ground velocity of a surface wave, which is a proxy for the peak size of an important component of the dynamic stress tensor, might be far from optimal.

Small triggered earthquakes release seismic energy instantly, resulting in characteristic, sharp signal peaks that clearly show up in low-noise seismograms recorded at nearby stations at frequencies above 2 Hz. In theory, a high signal-to-noise ratio (SNR) that takes advantage of this peak signal could serve as a detection tool (Chao and Obara, 2016). In practice, however, we find a significant number of spurious signals that also yield a high SNR. These signals exhibit a high amplitude, as they do in earthquakes, but are otherwise missing typical characteristics of earthquake signals (e.g., no clear *P* and *S* arrivals), and they may originate from environmental or instrumental activity. Similarly, if earthquake signals are present and noise is high, SNR values are low, leading to missed signals. Therefore, whether the SNR is an effective way of automating the detection of dynamically triggered earthquakes is still in question.

Currently, the process of detecting potentially triggered seismic events in well-studied, highly seismically active regions is facilitated by matched filtering with template signals or faster equivalents from previously detected, repeating earthquakes in the region (Van der Elst *et al.*, 2013; Yoon *et al.*, 2015). However, for other regions and in more exploratory studies such detections are performed by human experts. Researchers find themselves combing through thousands of seismograms per earthquake, often adjusting the data subsets, amplitude scales, the time windows, and frequency band multiple times per seismogram to identify actual earthquake signals among old and new types of glitches that appear to be earthquake signals but are not. Automating this

1. Department of Earth and Planetary Sciences, Northwestern University, Evanston, Illinois, U.S.A.; 2. Department of Computer Science, Northwestern University, Evanston, Illinois, U.S.A.

*Corresponding author: vivian@earth.northwestern.edu

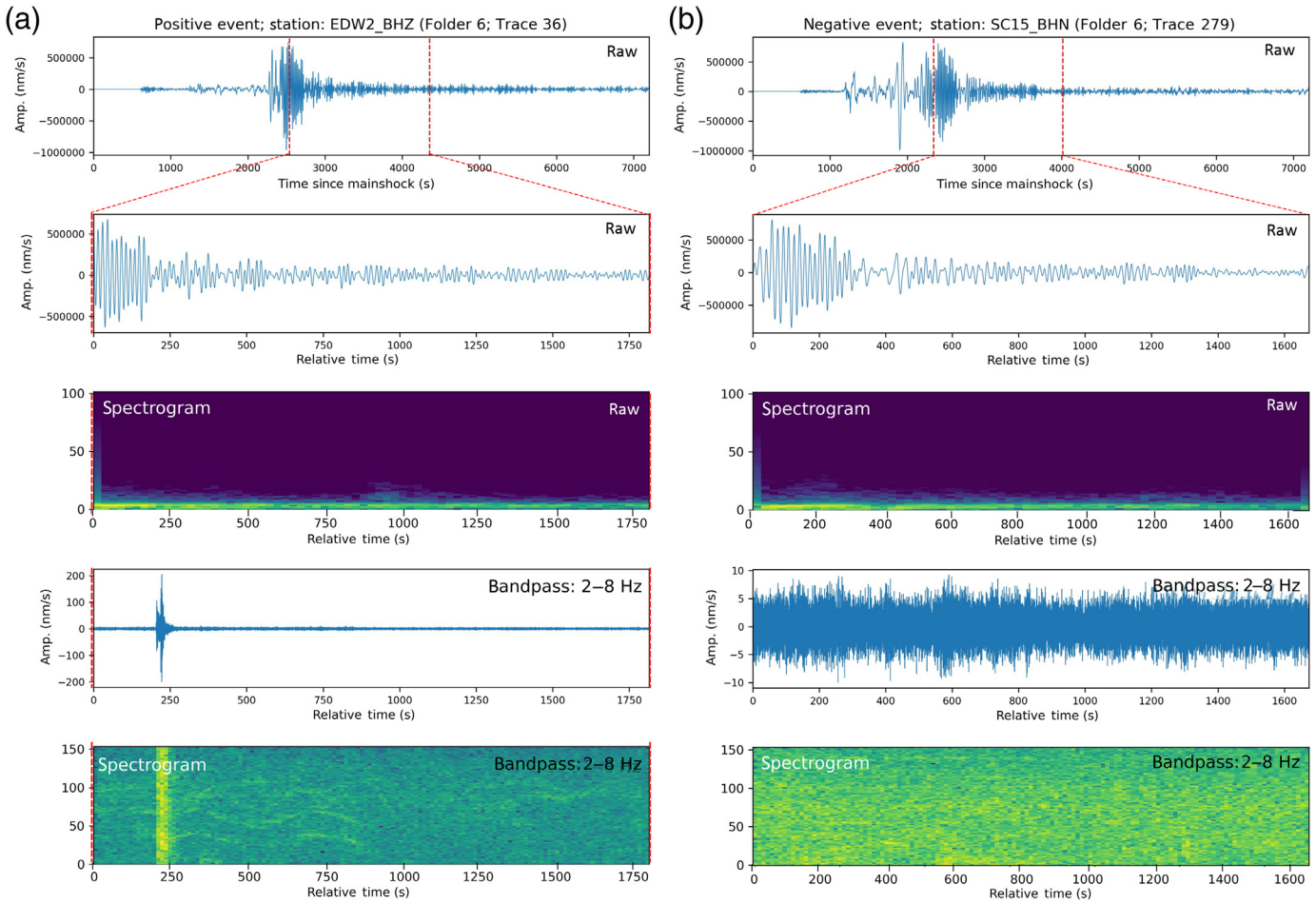

Figure 1. Examples of the potentially dynamic triggering of (a) the triggered earthquake and (b) an untriggered example following the surface waves of the 27 February 2010 M_w 8.8 Chile earthquake. The color version of this figure is available only in the electronic edition.

process, even partially, could save thousands of hours of expert time (Velasco *et al.*, 2016; Chao *et al.*, 2017). Here, we investigate methods that can examine all seismograms rather than a subset. We explore whether machine learning, in particular deep learning (Xing *et al.*, 2002; Perol *et al.*, 2018), can facilitate and/or automate the detection of dynamically triggered earthquakes.

The primary goal of this study is to train a deep learning model, specifically a deep metric learning approach, to identify signals from potentially triggered earthquakes in these seismograms. In metric learning, one learns a distance metric for examples (such as a group of seismograms with earthquake signals) in which examples of the same class should be near each other and those of different classes should be far apart. Given such a metric space, and a data set of seismograms with known class labels (e.g., earthquake vs. other), one can label new seismograms automatically.

One simple way to do this is using a K-nearest neighbor (KNN) approach. One measures the distance in the metric space between a new seismogram and the seismograms with known labels. A label is applied to the new seismogram by selecting the most frequent label for the K-nearest seismograms. Another obvious approach is to apply a Gaussian mixture model (GMM) by fitting a Gaussian distribution to the set of earthquakes and another Gaussian to the other events. A new seismogram can then be classified by selecting the Gaussian that assigns the highest probability to the new seismogram.

Learning the metric space so that one can effectively apply a GMM or KNN classifier is often done by mapping examples (e.g., seismograms) to points in a K-dimensional embedding space. We learn an appropriate embedding space with a deep convolutional neural network (CNN) that uses a deep clustering loss function to guide learning. This loss function (Hershey *et al.*, 2016) pushes embedding points belonging to the seismograms of different classes apart while pulling those of the same class together.

We train the algorithm on human-labeled seismograms spanning a certain time window after large teleseismic earthquakes and apply it to detect local earthquake signals in a similar seismogram time window after a separate large teleseismic earthquake, the so-called testing earthquake.

TABLE 1
Teleseismic Earthquake Information Used in This Study

Number	Date and Time (yyyy/mm/dd hh:mm:ss.s)	Longitude (°)	Latitude (°)	Depth (km)	M_w	Positive Example	Negative Example
1	2002/11/02 01:26:25.9	96.0	2.7	23.0	7.2	57	35
2	2003/09/25 21:08:19.5	143.6	41.8	47.3	7.3	4	118
3	2003/09/27 11:33:36.2	87.9	50.0	15.0	7.2	0	81
4	2004/12/26 01:01:09.0	94.3	3.1	28.6	9.0	0	5
5	2004/12/26 04:21:36.5	92.8	6.6	13.6	7.2	129	14
6	2005/03/28 16:10:31.5	97.1	1.7	25.8	8.6	89	1
7	2005/07/24 15:42:16.2	91.9	7.9	12.0	7.2	227	14
8	2010/02/27 06:35:14.5	−73.2	−36.0	23.2	8.8	98	301

In the following sections, we first describe how to interactively identify potentially dynamically triggered earthquake signals in a data set of over 1000 seismograms from several large-magnitude teleseismic earthquakes. The seismograms were recorded by seismic station arrays in the United States and the Himalaya–Tibet collision zone. We then describe a deep neural net to make a useful metric space and measure its performance by testing it on a subset of our earthquake data sets. Finally, we compare the performance of the deep learning model with that of a simple SNR approach. The supplemental material provides details on some of the data used.

Data

Data selection

We selected eight teleseismic earthquakes with magnitudes greater than 7 between 2002 and 2010. Among these eight teleseismic earthquakes (Table 1), surface waves from the 2010 M_w 8.8 Maule, Chile, earthquake likely triggered a local earthquake in California (Peng *et al.*, 2010). Two of the earthquakes, in Siberia and Japan, triggered none to few far-field earthquakes. The other five earthquakes occurred near Sumatra and Andaman Islands; their surface waves appear to have triggered several local earthquakes in the Himalaya–Tibet region. These earthquakes were selected based on previously published detections of possibly triggered earthquakes (Yao *et al.*, 2015; Mendoza *et al.*, 2016; Chao and Yu, 2018) and from additional analysis of data in this region (Fig. 2, Figs. S1, S2). Seismograms of these earthquakes were recorded by 882 stations from 10 networks (see Data and Resources) and obtained via the Incorporated Research Institutions for Seismology Data Management Center (IRIS-DMC), with the majority of the data from Nabelek *et al.* (2009).

We labeled waveforms that contained clear signals from the triggered earthquakes as positive examples and waveforms without clear earthquake signals as negative examples. Among all waveforms, 604 of the seismograms contain triggered earthquake signals (e.g., Fig. 1a) whereas the remaining 569 of the waveforms do not (e.g., Fig. 1b).

We identified earthquake signals via visual inspection of the seismograms at multiple time scales and in multiple frequency bands. To simplify the identification of dynamically triggered earthquakes, we define the triggering time window to contain the first arrival of Love waves and the last arrival of direct Rayleigh waves (Peng *et al.*, 2011; Chao *et al.*, 2013). Unlike the earlier arrival of body waves that contain high-frequency energy from the teleseismic hypocenter, the time window of surface waves typically contains little high-frequency energy, so it is easier to assign observed high-frequency energy to local, triggered earthquakes. To determine the surface-wave time window, we choose relatively loose bounding group velocities of 5 and 2 km/s, respectively, to cut the beginning and end of the window.

Data representation

We designed a data representation focused on the characteristics of signals from small, local earthquakes and that was minimally different from the raw time series. For each earthquake, we read in the seismograms in the surface-wave time window for both the 604 positive and 569 negative examples. We low-pass filtered all 1173 seismograms below 8 Hz while transferring the recorded digital units to ground-velocity seismograms and then high-pass filtered them above 2 Hz. Before feeding the filtered waveforms into the neural network, each waveform was standardized to have zero mean and unit variance. This was achieved for each waveform by subtracting the mean and dividing by the standard deviation of the waveform values (Aksoy and Haralick, 2001). This is a standard practice in machine learning that allows neural networks to ignore absolute amplitude as a distinguishing feature. The normalized

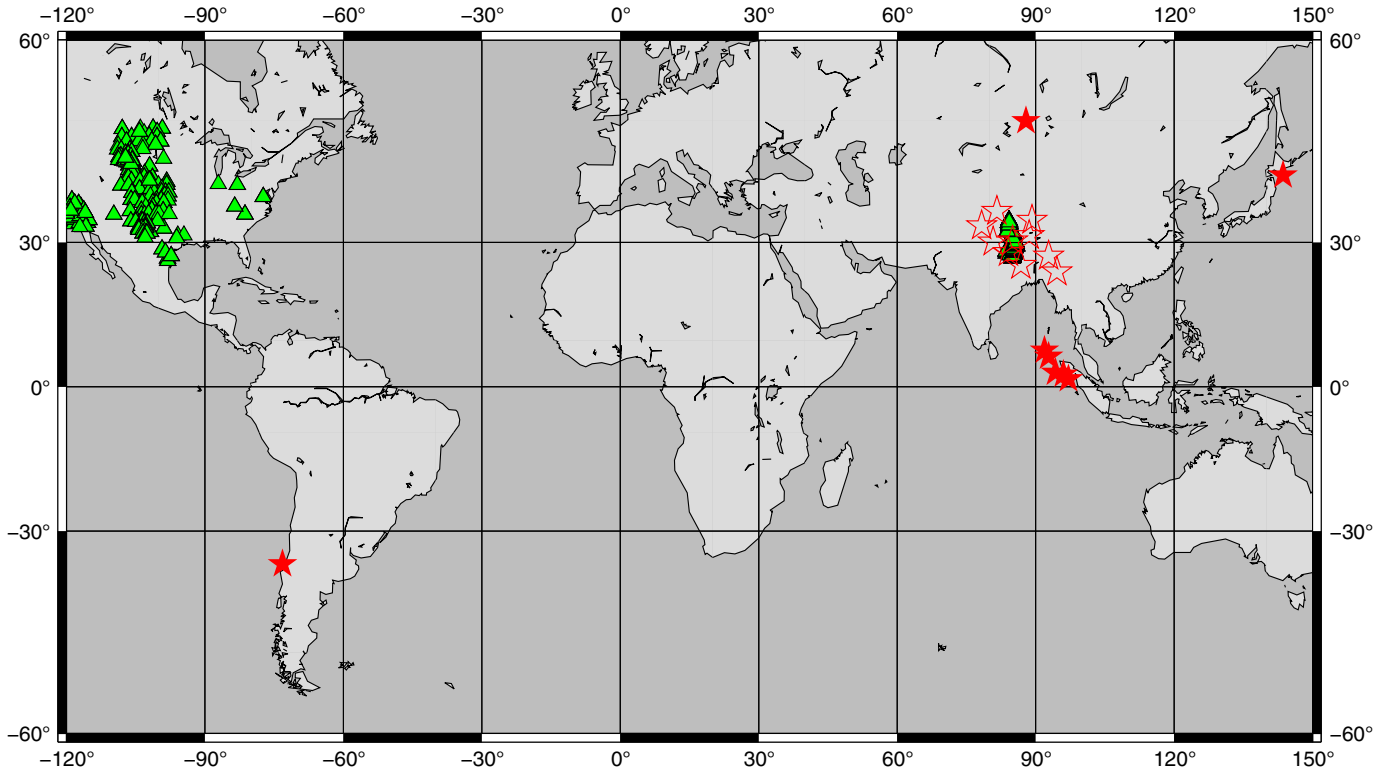

Figure 2. Map of the stations (triangles) and events (stars). Filled stars represent the large earthquakes listed in Table 1, and open stars represent the small earthquakes listed in Table 2. The color version of this figure is available only in the electronic edition.

data were then fed into a deep metric learning architecture for training, which we describe in the following section.

Methods

Machine learning

Network architecture. We used a CNN to label a new waveform as to whether it is likely to contain a signal from a small, local earthquake during a time window that marks the passage of surface waves from large distant earthquakes. Deep neural networks typically require many thousands of examples for effective training and generalization. Because of the rarity of dynamically triggered earthquakes and the amount of effort and expertise required to find and label these events reliably, we have access to relatively little training data. Even though we expand the data set below with waveforms from regular small, local earthquakes, the total amount of data remain insufficient for directly learning to classify individual events as earthquake or no earthquake.

We use a deep metric learning approach because it takes advantage of examples in relation to each other rather than only to a particular class, which has the effect of increasing the information that is fed into the network. If there are N examples, this results in $N^2/2$ pairwise relations. In metric learning, examples are mapped to points in a K-dimensional embedding space. If two examples are of the same class, they should be near each other in the embedding space. A pair of examples of different classes should be far apart.

Our neural network architecture takes filtered and standardized single channel recordings of a seismogram as input. This representation (a time series) is processed by a stack of neural network layers of neurons, each consisting of a convolution, a batch norm, a rectified linear unit, and a max pool. Each convolution has 16 channels, and there are 10 such layers in the network. The resulting network is very small, consisting of only 7440 parameters. We trained the network using an Adam optimizer, a batch size of 128, and a learning rate of 2×10^{-5}, with L2 regularization of 0.1. We trained for 100 epochs, in which one epoch consists of an entire run across the training set. Each batch was constructed such that there were an equal number of positive and negative examples from the training set. We have data from eight earthquakes for training. Rather than randomly selecting one earthquake as providing testing data and the seven others as providing training data, we did cross validation by giving each earthquake a turn at representing testing data while the network is trained on data from the remaining seven. In addition, we experimented with an independent data set of 10 small local earthquakes (Table 2, see the Results section).

The loss function. We used a deep metric learning approach to overcome the paucity of available training data. In metric learning, examples are mapped to points in a K-dimensional embedding space. Examples of the same class should be near

TABLE 2
10 Earthquakes with a Magnitude of <3.6 for Expanding the Training Data Set

Number	Date and Time (yyyy/mm/dd hh:mm:ss)	Longitude (°)	Latitude (°)	Depth (km)	M_w	Positive Example	Negative Example
1	2005/01/21 13:25:53	92.8	27.4	80.3	3.5	72	45
2	2005/03/05 12:19:24	84.1	28.1	33.0	3.5	137	72
3	2005/03/25 15:59:18	88.6	31.3	91.5	3.4	121	66
4	2005/04/02 01:55:39	94.6	24.3	95.8	3.5	87	37
5	2005/05/09 15:14:47	85.0	30.3	33.0	3.5	179	87
6	2005/05/22 20:30:08	89.2	34.1	10.0	3.3	117	30
7	2005/05/27 22:12:16	86.8	25.6	33.0	3.5	129	80
8	2005/06/06 08:36:13	81.6	35.7	10.0	3.6	85	39
9	2005/08/18 01:17:50	78.3	33.1	3.0	3.4	15	27
10	2005/08/21 19:53:33	80.8	30.1	33.0	3.5	60	41

TABLE 3
Metric Network (ML) and Signal-to-Noise Ratio (SNR) Accuracy Rate (Percentage) of the Test Data Set

Test Earthquake (yyyy/mm/dd hh:mm:ss.s)	ML Accuracy	ML Accuracy (Includes Small Earthquakes)	SNR Accuracy (Threshold of 13.8)	SNR Accuracy (Learning Threshold)
2002/11/02 01:26:25.9	92.4%, (*P*: 87.7%, *N*: 100%)	100%, (*P*: 100%, *N*: 100%)	83.7%, (*P*: 73.7%, *N*: 100%)	83.7%, (*P*: 73.7%, *N*: 100%)
2003/09/25 21:08:19.5	95.9%, (*P*: 75%, *N*: 96.6%)	99.2%, (*P*: 100%, *N*: 99.2%)	92.6%, (*P*: 100%, *N*: 92.4%)	92.6%, (*P*: 100%, *N*: 92.4%)
2003/09/27 11:33:36.2	91.4%, (*P*: n/a, *N*: 91.4%)	88.9%, (*P*: N/A%, *N*: 88.8%)	100%, (*P*: n/a, *N*: 100%)	100%, (*P*: n/a, *N*: 100%)
2004/12/26 01:01:09.0	100%, (*P*: n/a; *N*: 100%)	100%, (*P*: n/a, *N*: 100%)	100%, (*P*: n/a, *N*: 100%)	100%, (*P*: n/a, *N*: 100%)
2004/12/26 04:21:36.5	88.8%, (*P*: 87.6%, *N*: 100%)	90.2%, (*P*: 89.1%, *N*: 100%)	25.9%, (*P*: 17.8%, *N*: 100%)	25.9%, (*P*: 17.8%, *N*: 100%)
2005/03/28 16:10:31.5	100%, (*P*: 100.0%, *N*: 100%)	100%, (*P*: 100%, *N*: 100%)	98.9%, (*P*: 98.9%, *N*: 100%)	98.9%, (*P*: 98.9%, *N*: 100%)
2005/07/24 15:42:16.2	97.1%, (*P*: 99.6%, *N*: 57.1%)	97.5%, (*P*: 100%, *N*: 57.1%)	99.2%, (*P*: 99.1%, *N*: 100%)	99.2%, (*P*: 99.1%, *N*: 100%)
2010/02/27 06:35:14.5	88.7%, (*P*: 99%, *N*: 85.4%)	95.2%, (*P*: 93.9%, *N*: 95.7%)	98.5%, (*P*: 100%, *N*: 98.0%)	75.6%, (*P*: 100%, *N*: 67.7%)

P is the percentage of true positive examples ($\frac{P_{true}}{P_{total}}$), *N* is the percentage of true negative examples ($\frac{N_{true}}{N_{total}}$), and n/a represents no data.

each other in the embedding space, and examples of different classes should be far apart. Learning how different points compare to each other is a way of increasing the amount of available data as we now compare data points to each other, rather than fit each data point in isolation. Therefore, N examples result in $N^2/2$ pairwise relations. Doing this has been shown to effectively leverage data in machine learning paradigms for which there would be insufficient labeled training data (Koch *et al.*, 2015; Sung *et al.*, 2018) if examples were considered in isolation.

The loss function we used to train our network was the deep clustering loss function (Hershey *et al.*, 2016). Deep clustering works as follows: consider a set of seismograms X, with size N by T, in which N is the number of seismograms in the set and T is the length of each seismogram in samples. Our network mapped each seismogram to an embedding of size K. The output of the network across the entire set of seismograms is V, with size N by K. Each seismogram in the set has an expert label (see the Data Selection section) that indicates whether it contains a potentially triggered earthquake or not. This label is either a 1 or a 0, respectively, for each earthquake. For each seismogram, we constructed a one-hot encoding corresponding to its label. A one-hot label is a 1D vector with a 1 in the spot that corresponds to the index of the label and 0 s everywhere else. In this application, there are two classes (earthquake and no earthquake), which correspond to [1, 0]

or [0, 1] as the possible one-hot encodings for each seismogram. The set of labels is Y, which has size N by C, in which C is the number of classes ($C = 2$, in this case). Now, consider the ground-truth affinity matrix, YY^T, which has size N by N. Element (i, j) of YY^T contains 1, if the corresponding seismograms (i, j) have the same class. Otherwise, it contains 0. Now, consider the estimated affinity matrix VV^T, which also has size N by N. Element (i, j) of VV^T is the dot product of the K-dimensional embeddings output by the network on the corresponding seismograms in the set. The loss function of deep clustering enforces that the estimated affinity matrix is close to the ground-truth affinity matrix as follows:

$$L_{DC} = ||VV^T - YY^T||_2^F.$$

By making the estimated affinity matrices match the true affinity matrices, we learn an embedding space in which seismograms of the same class are near each other and those of different classes are far away from each other. We trained the network using batches from the training set. A batch contained 128 seismograms, resulting in affinity matrices of size 128 by 128. A final note: deep clustering was originally designed for audio source separation and not for general metric learning problems. This work is the first we know of that applies the deep clustering loss function for metric learning in another domain.

Clustering in the embedding space. We used the network as a feature extraction step from the seismogram. First, we visualized the embeddings of the training data and ran each training example through the network, producing embeddings for each seismogram. Then, we projected the embeddings onto a 2D space using principal component analysis (PCA). PCA is a dimension-reduction technique that finds a lower dimensional space in which each axis maximizes the variance in the data. Because we reduce the embedding space to two dimensions, we can easily visualize it (Figs. 3, 4). Points that are far from one another in the 2D space are also far from one another in the full embedding space.

The embedding outputs by the network for the training data are linearly separable into earthquakes and no earthquakes (Fig. 3). The contours in Figures 3 and 4 show the distribution of the training data in the embedding space. Positive examples (e.g., those that contain an earthquake) are contoured with dashed lines and negative examples with dotted lines. The dimensions of the plots in Figures 3 and 4 show the two dimensions along which the embeddings have maximum variance; overlapping points can still be separated in a third or higher dimension. Codes are available (see Data and Resources).

We plot the test data as open white (positive) and black (negative) circles. Strong overlap between the training and testing distributions in the embedding space suggests excellent detection performance. However, to determine whether our machine learning approach is superior to other automated methods that are computationally cheaper and easy to implement, we compare these results with those from using an SNR threshold.

SNR

Straightforward attempts to automate the detection of dynamically triggered earthquakes include computing an SNR between a signal in the window of interest and pre-*P*-wave background noise. The SNR is calculated for seismograms that have already been bandpassed between 2 and 8 Hz by estimating the signal (S) as the maximum value of the seismogram envelope within the surface-wave time window and the noise (N) as the median of the seismogram envelope within a time window immediately preceding the arrival of seismic waves from a distant earthquake.

Results

Machine learning

To test the robustness of the performance of our algorithm, we apply it for each of the eight training versus testing splits (e.g., events 2–8 as the training data and event 1 as the test data, listed in Table 1). Each training data set contains about 1000 seismograms from the remaining seven of the eight earthquakes (Table 1). We then test the trained network on seismograms from the remaining earthquake in Table 1, which is not used for training (Fig. 3).

To assess the performance of the machine learning approach, we define accuracy A as the fraction of true label examples (P_{true}) in the test set divided by all examples (P_{total}) in the test set:

$$A = \frac{P_{\text{true}}}{P_{\text{total}}}.$$

For all of the training–testing splits, accuracy values are above 88%, shown in Table 3. Data projections onto the 2D subspace of the embedding space (Fig. 3) show that both training and testing data are well, though not entirely, separated. Detection accuracy for the training data sets is consistently over 95%.

Clearly, using a deep metric learning approach leads to acceptable levels of accuracy (>88%) even when few data are available for training. To further improve the training of our algorithm, we added data from 10 regular local earthquakes (Fig. 2) with magnitudes of less than 3.6. We processed these data in the same way as we processed the data from the large teleseismic earthquakes, except that we defined time windows starting at random time shifts from the earthquake signals. In addition to these 1002 new positive examples, we also selected negative examples using the same window length as the positive example but shifting it to fall entirely before the *P*-wave arrival. Because about half of the noise windows contain spurious signals, we reduced the number of negative examples to 524 seismograms. Including these additional examples (Table 2), we retrained our algorithm and show

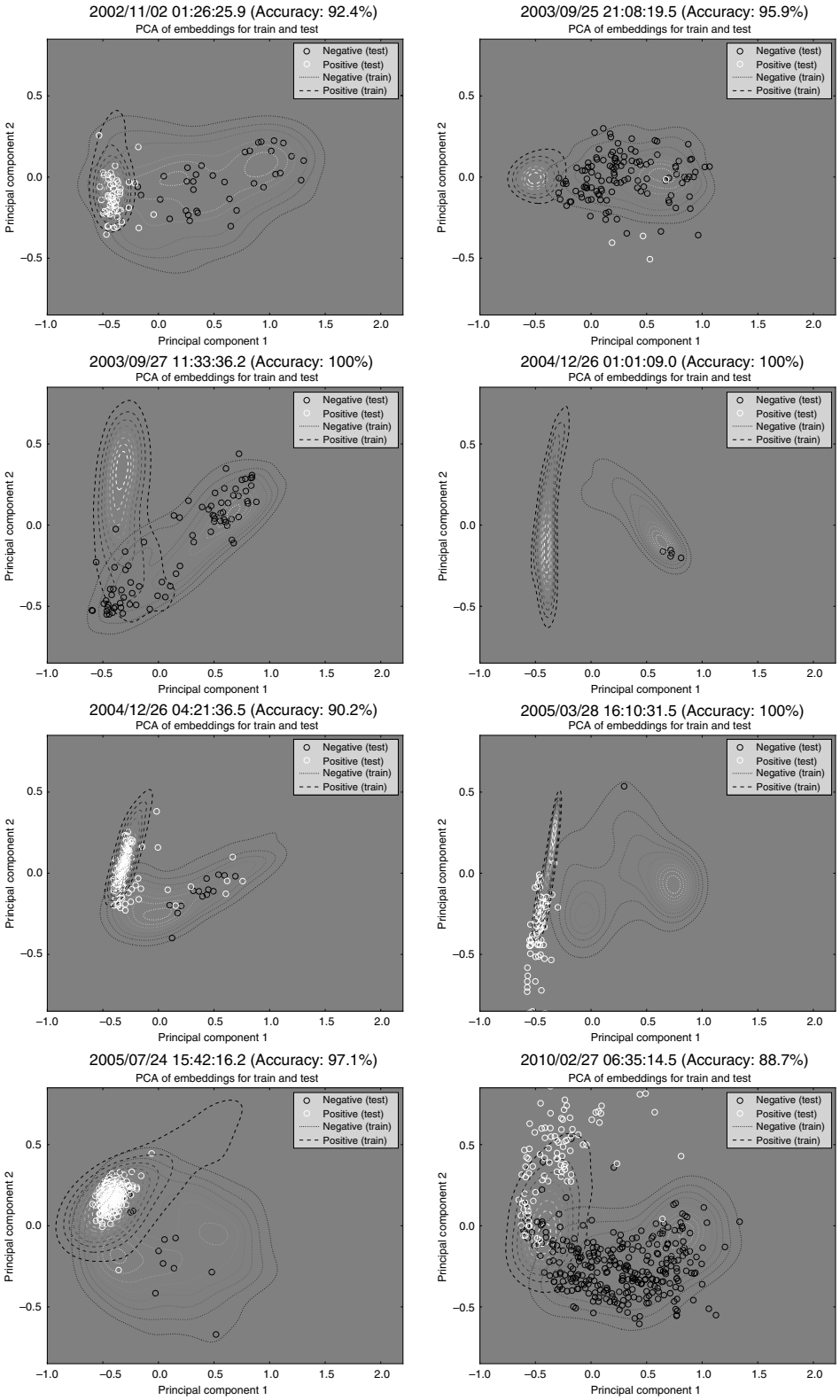

Figure 3. Names and times of the test earthquakes listed in each figure. Projections of 2D spaces via the principal component analysis (PCA) from the 10D embeddings. The convolutional neural network (CNN) for classifying the training data set (contour maps) and the test data set (open circles) from the teleseimic earthquakes. Each circle represents a seismogram, white circles represent a positive event (e.g., an earthquake), and black circles represent a negative event. The accuracy of the test earthquake is listed on the title of each figure.

results in Figure 4. This expansion of the training data set increased the earthquake detection accuracy for waveforms from earthquake 8 (Chile) by 6.5%. For several other testing earthquakes, the training expansion had the effect of tightening the positive and negative classes and enhancing their separation, while causing testing labels to fall more outside of the learned domains. Nevertheless, the accuracy of the results remains above 88%. Thus, we find that our algorithm is relatively stable despite the data set being small for a machine learning application and continue to use the original data set of 1173 seismograms.

Separation by SNR value

Figure 5 shows the distribution of SNR values for the seismograms from the earthquakes listed in Table 1. From 604 positive examples and 569 negative examples, we choose an optimal SNR threshold of 13.8 (Table 1) to separate the positive examples (SNR > threshold) from the negative examples (SNR < threshold). We labeled testing seismograms with an SNR over the threshold as positive and the others as negative. Figure 5 shows that this approach provides variable detection accuracy of 25.9%–100% (Table 3).

We also explored the accuracy of detection with a learned rather than fixed threshold. To this end, we computed the root mean square of the SNR for only the training data set and call this value the learned SNR threshold. Figure S3 shows the accuracy and SNR histograms for each training–testing split we also used in the machine learning part of this article.

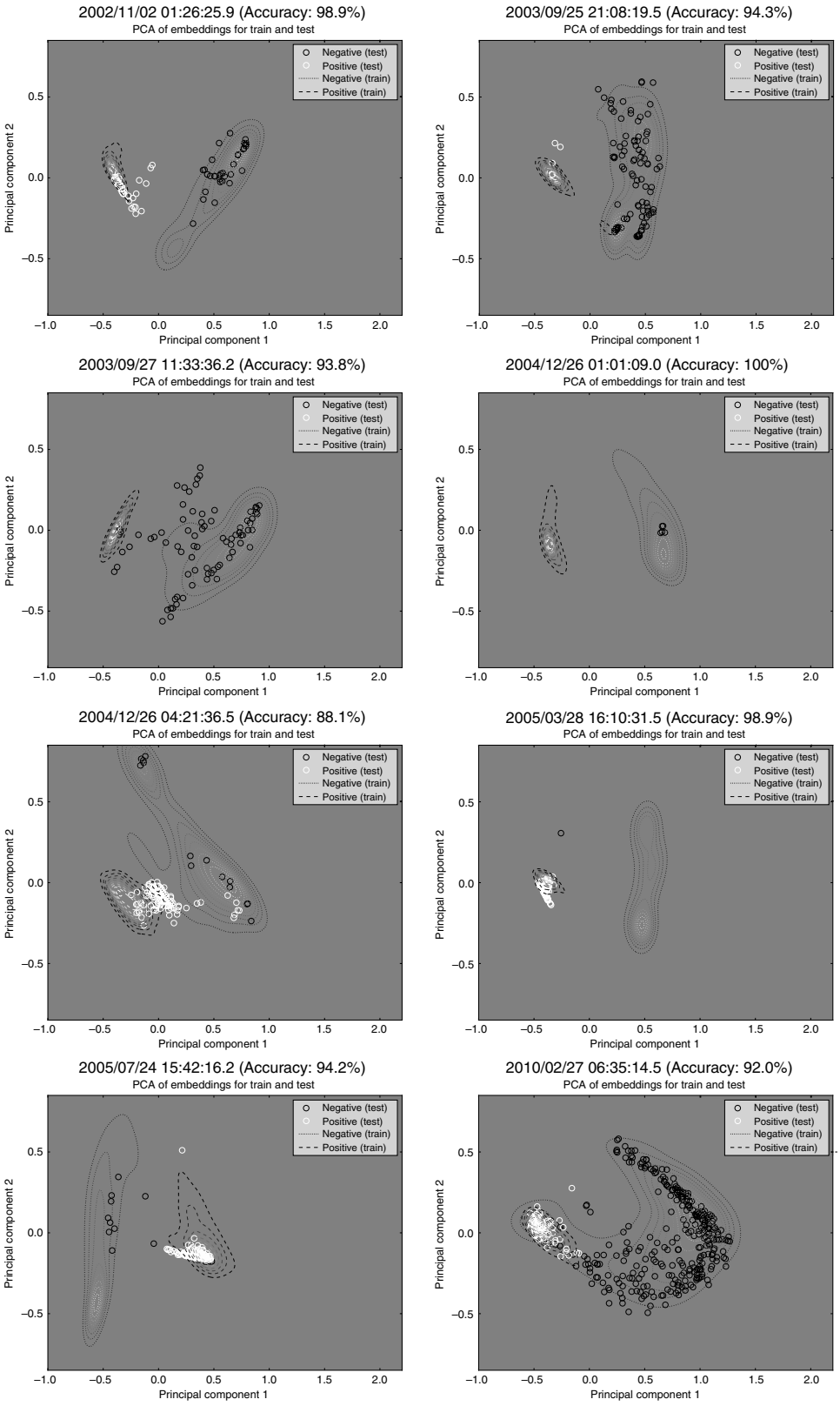

Figure 4. Names and times of the test earthquakes listed in each figure. We add more data to the training set from the 10 small earthquakes listed in Table 2. Projections of 2D space via PCA from the 10D embeddings. A CNN for classifying the training data set (contour maps) and the test data set (open circles) from the teleseismic earthquakes. Each circle represents a seismogram; white circles represent a positive event (e.g., an earthquake), and black circles represent a negative event. Accuracy of the test earthquake is listed on the title of each figure.

Perhaps surprisingly, Figure S3 shows slightly worse accuracy values than Figure 5 and a particularly low accuracy value for the Sumatra earthquake 5 and the Chile earthquake 8 (Table 3), and Figure S2 shows seismograms from the training data set with the lowest and highest SNR. Both examples are from earthquake 8 (27 February 2010 M_w 8.8 Maule, Chile, earthquake).

Discussion

In this article, we explored and tested the potential effectiveness of using a supervised machine learning for identifying seismograms that contain signals from potentially dynamically triggered earthquakes. The motivation for this research stemmed from a need to quickly and optimally identify instant triggered earthquakes. As we had access to relatively few data to address our problem, we chose a deep metric learning network approach over other machine learning algorithms (e.g., ImageNet Krizhevsky *et al.*, 2012). By exploiting the relation between the examples in our data set as a training signal (e.g., both seismograms are examples of potentially triggered earthquakes) rather than the relation between an example and its class (e.g., this seismogram does not contain an earthquake), we effectively squared the amount of data we used to train the model. To the best of our knowledge, this work is the first application of deep clustering to this problem of dynamically triggered earthquake detection. We compared the success of this approach to that of a more straightforward approach

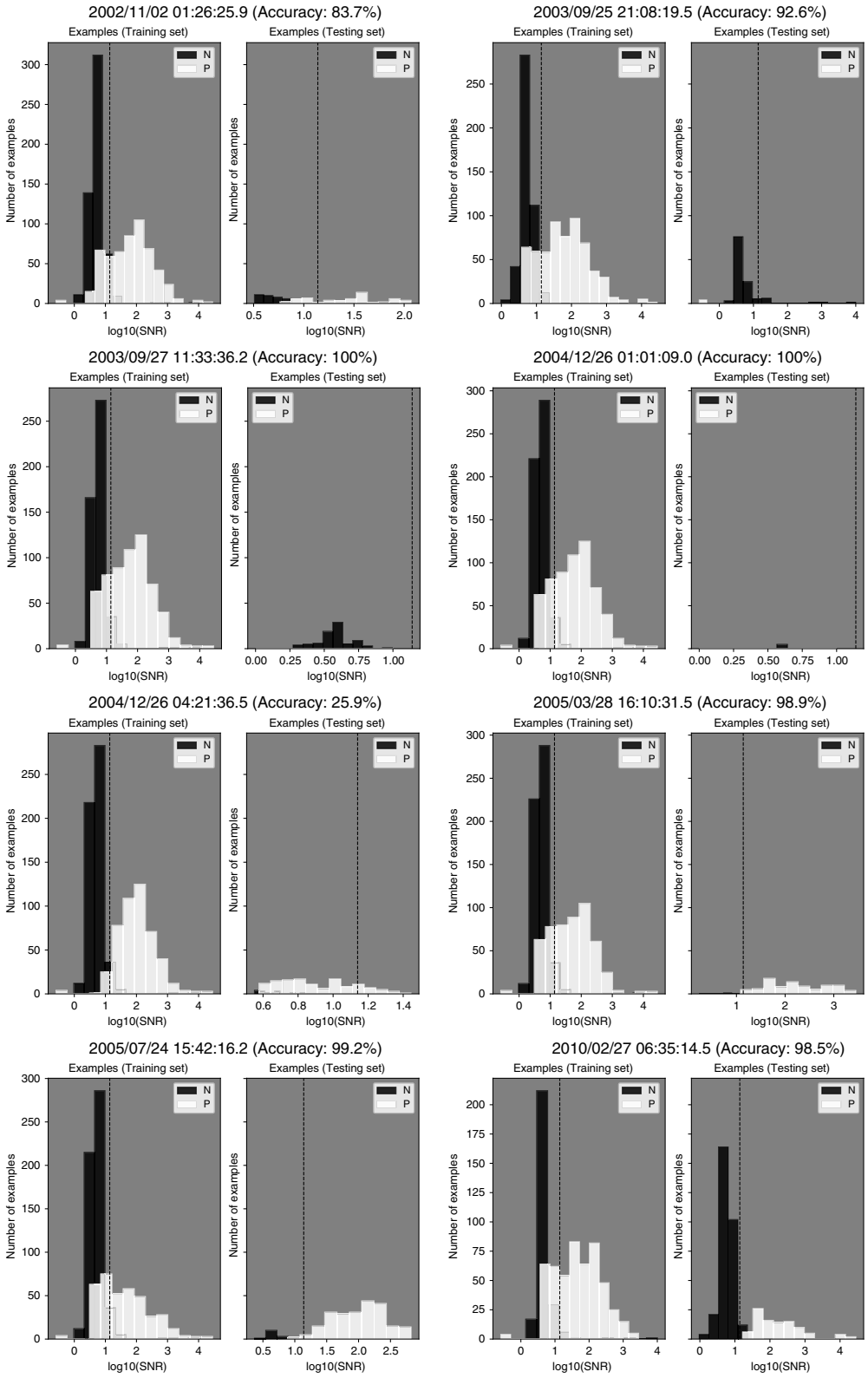

Figure 5. Histogram of signal-to-noise ratio (SNR) measurements from the teleseismic earthquakes listed in Table 1. For each earthquake, the left histogram shows the base-10 logarithm of the SNR of positive and negative examples in the training set, and the right histogram is the base-10 logarithm of the SNR results for the test set. The vertical dashed line, with an SNR of 13.8, is the threshold of the SNR. The white bars represent positive examples, and the black bars represent negative examples. The accuracy of the test set is listed on the title of each figure.

based on an SNR threshold and measured success in terms of accuracy.

Table 3 lists eight training–testing configurations with accuracy values of over 88% for the testing data. We also classified positive and negative examples by simply comparing the SNR to a threshold SNR of 13.8. We measured the SNR for each test earthquake (Table 3) and observed that the 26 December 2004 M_w 7.2 Sumatra earthquake had the lowest classification accuracy 25.9%. When used for testing, the seven other earthquakes each yielded higher classification accuracies in the range of 83.7% to 100.0% (Fig. 5). We observed a clear series of potentially triggered earthquake signals during the surface-wave windows of the 26 December 2004 M_w 7.2 Sumatra earthquake but also noted the presence of high-seismic energy in the noise windows, yielding low-SNR values for the positive examples.

To discuss the performance of the SNR and the metric network, we selected data from earthquake 1 (Sumatra) as a representative testing data set because it contains a relatively even distribution of positive and negative examples and also leaves a relatively even distribution of positive and negative examples for training. The classification accuracy of the SNR method was 83.7% and that of the metric network 92.4%.

The 27 February 2010 M_w 8.8 Maule, Chile, earthquake was a test case that provided several interesting results: for one, the classification accuracy of the network for detecting true positive examples (Table 3) was high in the

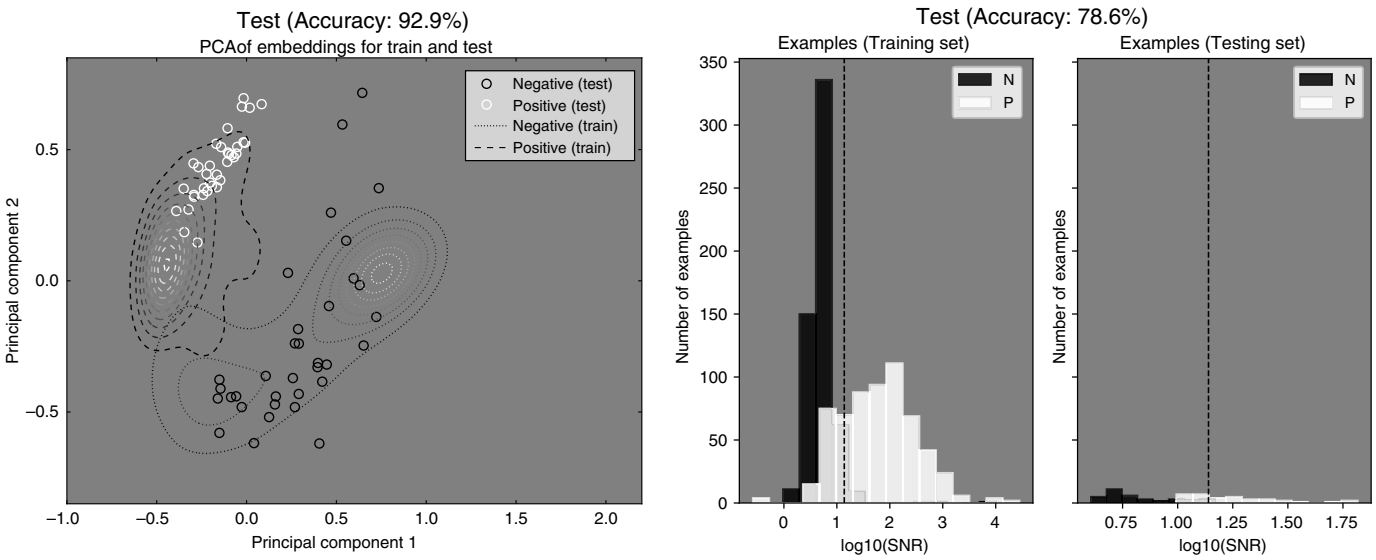

Figure 6. Test for the stability of the machine-learning algorithm with a less winnowed data set in Table 1. We selected 35 positive examples (white) of earthquakes that include earthquake signals as well as relatively strong noise signals in the surface-wave window and 35 negative examples (black) that do not include earthquake signals but have high-SNR due to strong noise signals in the surface-wave window. The accuracy of the machine-learning algorithm is 92.9% and the SNR is 78.6%, and seismograms are shown in Figure S4.

training data set and higher than that for true negative examples. In addition, this earthquake was the only earthquake recorded by North American stations and its seismograms recorded different path and site effects than those recorded in the Himalaya–Tibet region. Therefore, for this particular case, the training data set consisted entirely of data from the Himalaya–Tibet region.

In the process of applying both the simple SNR threshold and the machine learning method to achieve the functional detection of potentially triggered earthquakes, we found ourselves increasingly winnowing our data set, which primarily consisted of the removal of ambiguous data. The data set that we eventually used consisted of only the clearest examples of earthquake signals and clear absences of high-frequency signals in the surface-wave windows.

To test the algorithm on a less winnowed data set, we selected 35 positive examples and 35 negative examples as the test set, which was slightly noisier (Fig. S4) than the clean data set in Table 1. These data might have had a relatively high-noise level or contained more than one type of signal. We used the clean data set in Table 1 as the training set. For the testing data, we found that the accuracy value of the machine learning algorithm was 92.9% and that of the SNR was 78.6% shown in Figure 6. From the projection of the 2D space figure, these 70 examples were well separated. This finding indicates that our algorithm was able to detect small earthquakes with various data sets and was more than 10% more accurate than SNR.

To explore the necessity and limitations of neural networks, we tested the convolution with 2, 4, 8, and 16 channels and the network with 1, 2, 5, and 10 layers. Figure S5 indicates that if the network has less than eight channels and/or the network has only one layer, the accuracy of the testing set begins to deteriorate. However, with eight channels and two layers, the algorithm's detection accuracy improves to nearly the levels reported above for the full 10 layers and 16 channels network (Figs. S6, S7 and Table S1). In the future, we plan to include more data from different regions into the training set and explore the possibility of detecting tremor, so we need to have a stable and efficient algorithm with room for growth. Therefore, we decided to use the convolution with 16 channels and 10 layers network in this study.

Conclusion

Our analysis showed that our deep metric learning algorithm performed better and more robustly at detecting small, local, possibly triggered earthquakes than the SNR approach, and allowed the use of a more diverse dataset. The trained deep metric networks displayed a classification accuracy of higher than 88% for all test datasets. In this study, we focused on the identification of local earthquakes that occur during the passage of teleseismic surface waves, but it is not restricted to either this time window or to earthquakes. In principle, for example, our machine-learning algorithm may also be applied to detecting earthquakes at any time or to detecting triggered tremor.

Data and Resources

All of the seismic data were downloaded through the Incorporated Research Institutions for Seismology Data Management Center (IRIS) Wilber 3 system (http://ds.iris.edu/wilber3/find_event) and the IRIS Data Management System (DMC) (http://ds.iris.edu/ds/nodes/dmc/data/).

They included the seismic networks of the USArray: (1) the AZ (ANZA Regional Network, doi: 10.7914/SN/AZ); (2) the BK (Berkeley Digital Seismic Network [BDSN] doi: 10.7932/BDSN); (3) the CI (Southern California Seismic Network, doi: 10.7914/SN/CI); (4) the II (Global Seismograph Network [GSN]-IRIS/International Deployment of Accelerometers [IDA], doi: 10.7914/SN/II); (5) the IU (Global Seismograph Network, GSN-IRIS/U.S. Geological Survey [USGS], doi: 10.7914/SN/IU); (6) the TA (USArray Transportable Array, National Science Foundation [NSF] EarthScope Project); (7) the US (United States National Seismic Network, doi: 10.7914/SN/US); (8) the XF (Nepal–Himalaya–Tibet Seismic Transect, doi: 10.7914/SN/XF_2002, Nabelek, 2002); (9) XR (Seismic Investigation of Edge Driven Convection Associated with the Rio Grande Rift-Flex Array, doi: 10.7914/SN/XR_2008); and (10) the XV (Geometry and Kinematic of Basement-Involved Foreland Arches, doi: 10.7914/SN/XV_2009). GitHub codes are available at https://github.com/interactiveaudiolab/earthquakes. All websites were last accessed August 2019. The supplemental material for this article includes examples of data, variants of our implementation, and a table of results.

Acknowledgments

This research is funded by two seeds Grants from Northwestern University's Data Science Initiative (DSI) and the Integrated Data-Driven Discovery in Geophysical and Astrophysical Sciences (ID^{3}EAS) program under National Science Foundation (NSF) Grant Number NSF-NRT 1450006. The authors are grateful to Sue Griesbach and Bob Langan for having donated the current graphics card and graphic processing unit (GPU) machine on which our calculations were performed. The authors are also grateful to John Nabelek for having generated a treasure trove of data for our machine learning algorithm with the broadband Nepal–Himalaya–Tibet Seismic Transect (Himalayan-Tibetan Continental Lithosphere during Mountain Building [Hi-CLIMB]) (Nabelek, 2002) and for a discussion of seismicity in the Hi-CLIMB study region. The authors especially thank Chastity Aiken, whose thoughtful reviews helped to improve this article.

References

Aiken, C., and Z. Peng (2014). Dynamic triggering of microearthquakes in three geothermal/volcanic regions of California, *J. Geophys. Res.* **119,** 6992–7009, doi: 10.1002/2014JB011218.

Aksoy, S., and R. M. Haralick (2001). Feature normalization and likelihood-based similarity measures for image retrieval, *Pattern Recogn. Lett.* **22,** no. 5, 563–582, doi: 10.1016/S0167-8655(00)00112-4.

Brodsky, E. E., and N. J. van der Elst (2014). The uses of dynamic earthquake triggering, *Annu. Rev. Earth Planet. Sci.* **42,** no. 1, 317–339, doi: 10.1146/annurev-earth-060313-054648.

Chao, K., and K. Obara (2016). Triggered tectonic tremor in various types of fault systems of Japan following the 2012 Mw 8.6 Sumatra earthquake, *J. Geophys. Res.* **121,** no. 1, 170–187, doi: 10.1002/2015JB012566.

Chao, K., and C. Yu (2018). A MATLAB GUI for examining triggered tremor: A case study in New Zealand, *Seismol. Res. Lett.* **89,** no. 6, 2362–2373, doi: 10.1785/0220180057.

Chao, K., Z. Peng, H. Gonzalez-Huizar, C. Aiken, B. Enescu, H. Kao, A. A. Velasco, K. Obara, and T. Matsuzawa (2013). A global search for triggered tremor following the 2011 Mw 9.0 Tohoku earthquake, *Bull. Seismol. Soc. Am.* **103,** no. 2B, 1551–1571, doi: 10.1785/0120120171.

Chao, K., Z. Peng, Y.-J. Hsu, K. Obara, C. Wu, K.-E. Ching, S. van der Lee, H.-C. Pu, P.-L. Leu, and A. Wech (2017). Temporal variation of tectonic tremor activity in southern Taiwan around the 2010 ML6.4 Jiashian earthquake, *J. Geophys. Res.* **122,** no. 7, 5417–5434, doi: 10.1002/2016JB013925.

Freed, A. M. (2005). Earthquake triggering by static, dynamic, and postseismic stress transfer, *Annu. Rev. Earth Planet. Sci.* **33,** no. 1, 335–367, doi: 10.1146/annurev.earth.33.092203.122505.

Gonzalez-Huizar, H., A. A. Velasco, Z. Peng, and R. Castro (2012). Remote triggered seismicity caused by the 2011, M9.0 Tohoku, Japan earthquake, *Geophys. Res. Lett.* L10302, doi: 10.1029/2012GL051015.

Hershey, J. R., Z. Chen, J. Le Roux, and S. Watanabe (2016). Deep clustering: Discriminative embeddings for segmentation and separation, *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference*, 31–35.

Hill, D. P., and S. G. Prejean (2015). Dynamic triggering, in *Treatise on Geophysics*, Second Ed., G. Schubert (Editor), Vol. 4, Elsevier, Oxford, United Kingdom, 273–304.

Koch, G., R. Zemel, and R. Salakhutdinov (2015). Siamese neural networks for one-shot image recognition, *ICML Deep Learning Workshop*, Lille, France, 6–11 July 2015, Vol. 2, available at https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf (last accessed December 2018).

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Editors), Neural Information Processing Systems, Nevada, 1097–1105.

Mendoza, M. M., A. Ghosh, and S. S. Rai (2016). Dynamic triggering of small local earthquakes in the central Himalaya, *Geophys. Res. Lett.* **43,** no. 18, 9581–9587, doi: 10.1002/2016GL069969.

Nabelek, J. (2002). Collaborative research: Lithospheric scale dynamics of active mountain building along the Himalayan-Tibetan collision zone, *International Federation of Digital Seismograph Networks. Dataset/Seismic Network*, doi: 10.7914/SN/XF_2002.

Nabelek, J., G. Hetenyi, J. Vergne, S. Sapkota, B. Kafle, M. Jiang, H. Su, J. Chen, B. S. Huang, and the Hi-CLIMB Team (2009). Underplating in the Himalaya-Tibet collision zone revealed by the Hi-CLIMB experiment, *Science* **325,** no. 5946, 1371–1374.

Peng, Z., and J. Gomberg (2010). An integrated perspective of the continuum between earthquakes and slow-slip phenomena, *Nature Geosci.* **3,** no. 9, 599.

Peng, Z., D. P. Hill, D. R. Shelly, and C. Aiken (2010). Remotely triggered microearthquakes and tremor in central California following the 2010 Mw8.8 Chile Earthquake, *Geophys. Res. Lett.* **37,** L24312, doi: 10.1029/2010GL045462.

Peng, Z., C. Wu, and C. Aiken (2011). Delayed triggering of microearthquakes by multiple surface waves circling the Earth, *Geophys. Res. Lett.* **38,** no. 4, doi: 10.1029/2010GL046373.

Perol, T., M. Gharbi, and M. Denolle (2018). Convolutional neural network for earthquake detection and location, *Sci. Adv.* **4,** no. 2, doi: 10.1126/sciadv.1700578.

Quinones, L. A., H. R. DeShon, M. B. Magnani, and C. Frohlich (2018). Stress orientations in the Fort Worth Basin, Texas,

determined from earthquake focal mechanisms, *Bull. Seismol. Soc. Am.* **108,** no. 3A, 1124–1132, doi: 10.1785/0120170337.

Sung, F., Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales (2018). Learning to compare: Relation network for few-shot learning, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 1199–1208.

Van der Elst, N. J., and E. E. Brodsky (2010). Connecting near-field and far-field earthquake triggering to dynamic strain, *J. Geophys. Res.* **115,** no. B07311, doi: 10.1029/2009JB006681.

Van der Elst, N. J., H. M. Savage, K. M. Keranen, and G. A. Abers (2013). Enhanced remote earthquake triggering at fluid-injection sites in the midwestern United States, *Science* **341,** no. 6142, 164–167, doi: 10.1126/science.1238948.

Velasco, A. A., R. Alfaro-Diaz, D. Kilb, and K. L. Pankow (2016). A time-domain detection approach to identify small earthquakes within the continental United States recorded by the USArray and regional networks, *Bull. Seismol. Soc. Am.* **106,** no. 2, 512–525, doi: 10.1785/0120150156.

Velasco, A., S. Hernandez, and T. Parsons (2008). Global ubiquity of dynamic earthquake triggering, *Nature Geosci.* **1,** 375–379, doi: 10.1038/ngeo204.

Wang, B., R. M. Harrington, Y. Liu, H. Kao, and H. Yu (2018). Remote dynamic triggering of earthquakes in three unconventional Canadian hydrocarbon regions based on a multiple-station matched-filter approach, *Bull. Seismol. Soc. Am.* **109,** no. 1, 372–386, doi: 10.1785/0120180164.

Weingarten, M., S. Ge, J. W. Godt, B. A. Bekins, and J. L. Rubinstein (2015). High-rate injection is associated with the increase in U.S. mid-continent seismicity, *Science* **348,** no. 6241, 1336–1340, doi: 10.1126/science.aab1345.

Xing, E. P., A. Y. Ng, M. I. Jordan, and S. Russell (2002). Distance metric learning, with application to clustering with side-information, *Proc. 15th International Conference on Neural Information Processesing Systems*, 521–528.

Yao, D., Z. Peng, and X. Meng (2015). Remotely triggered earthquakes in South-Central Tibet following the 2004 Mw 9.1 Sumatra and 2005 Mw 8.6 Nias earthquakes, *Geophys. J. Int.* **201,** no. 2, 543–551, doi: 10.1093/gji/ggv037.

Yoon, C. E., O. O'Reilly, K. J. Bergen, and G. C. Beroza (2015). Earthquake detection through computationally efficient similarity search, *Sci. Adv.* **1,** no. 11, doi: 10.1126/sciadv.1501057.

Manuscript received 8 July 2019
Published online 2 January 2020