Identifying Asperity Patterns Via Machine Learning Algorithms

Kostantinos Arvanitakis(✉) and Markos Avlonitis

Department of Informatics, Ionian University, 49100 Corfu, Greece
{c14arva,avlon}@ionio.gr

Abstract. An asperity's location is very crucial in the spatiotemporal analysis of an area's seismicity. In literature, b-value and seismic density have been proven as useful indicators for asperity location. In this paper, machine learning techniques are used to locate areas with high probability of asperity existence using as feature vector information extracted solely by earthquake catalogs. Many machine learning algorithms are tested to identify those with the best results. This method is tested for data from the wider region of Hokkaido, Japan where in an earlier study asperities have been detected.

Keywords: Asperity · Density · b-value · Seismicity · Machine learning

1 Introduction

Asperities are considered to be large and strong patches on a seismic fault. They have dimensions ranging from less than a kilometer to tens of kilometers. These are locked inside the faults under high pressure and release most of their energy during the eventual earthquake [1]. Asperities can accumulate large portions of tectonic stress and by their rupture an earthquake of great magnitude is generated.

In literature it has been shown that the b-value, i.e. the slope of the Frequency - Magnitude distribution, is significantly lower in asperities, in comparison with other fault zones which have higher b-value [1].

In a recent study made in the region of Hokkaido (Japan), [2] the authors proposed a method for locating asperities by means of the earthquakes' density. Therein, asperities were selected as sections of a region with small number of events, at least one event with high magnitude and surrounded by sections with large number of events.

There are two popular methods of how to locate an asperity. As by definition, an asperity is a high stress area surrounded by low stress areas, the former method calculates the stress levels of a region and points the areas with higher levels [3–5].

The later method uses the surface's slip. By using GPS data of the slip distribution on the earth's surface, the asperities are located in regions where big deformations are detected [6–12].

The main problem with these methods is that they are based on reverse engineering and as a result produce non-unique probabilistic conclusions. The creation of an accurate method to locate asperities is of highly importance. As previously stated asperities are

L. Iliadis and I. Maglogiannis (Eds.): AIAI 2016, IFIP AICT 475, pp. 87–93, 2016.
DOI: 10.1007/978-3-319-44944-9_8

responsible for generating large earthquakes. Thus, an asperity's location is information of high value. Due to the high probability of generating large earthquakes, this information can help the state apparatus in decision making and strategic planning of the seismic regulations according to the building construction and also the expansive policy of cities and civil engineering projects, in order to increase the safety of human life.

Many examples of machine learning, data mining, and feature extraction methods can be found in literature, that have been used in seismicity analysis as tools for the earthquake hazards prediction and prevention. A recent study [13] used co-occurrence cluster mining to identify earthquake swarms and seismic patterns in different regions but with similar properties that probably will be correlated. Also data mining methods have been used [14] for forecasting the month or the year an earthquake will occur. Neural networks also have been used for earthquake prediction. In detail, Panakkat et al. [15] proposed a recurrent neural network, with training and testing data from the Southern California and the San Francisco Bay, to predict the time and location of seismic events. In another study [16], relating with neural networks, the authors proposed and tested in the wider region of Chile, a neural network that could predict the probability that an earthquake of magnitude larger than a threshold value will be generated and also the probability of an earthquake's magnitude with a limited magnitude interval.

The purpose of this study is to introduce a machine learning approach in locating asperities in space. Our hypothesis of whether machine learning can identify asperities will be tested on data from the region of Hokkaido based on the result of Takahashi and Kasahara [2].

2 Materials and Methods

2.1 Seismic Data

The hypocenter data used in the experiments were determined by Hokkaido University, Sapporo, Japan. The catalog contains data from July 1st, 1976 until December 31st, 2002. Every earthquake in the catalog is a record with information about the time the earthquake occurred (year, month, day, hour, minute), the earthquakes epicenter (latitude, longitude, depth), and the earthquake's magnitude. The tested area (Hokkaido region) is located between 41^{o} – 43^{o} Latitude and 142.5^{o} – 145.5^{o} Longitude.

2.2 Data Representation

Many experiments were conducted using WEKA [17], a platform that allows experimenting with state-of-the-art techniques in machine learning. Due to the complexity of the earthquake phenomenon, there are not many features that can describe an area's seismicity thorough. For the presented task, the b-value and the seismic density features were selected, which are acceptable characteristics, among seismology researchers, of an area's seismicity. Also the longitude and latitude attributes were selected to describe the data's location.

– Latitude of the corresponding area (Numeric)
– Longitude of the corresponding area (Numeric)
– Density of earthquake instances in the corresponding area (Numeric)
– b-value of the Gutenberg-Richter frequency-magnitude distribution (Numeric)
– Asperity indicator (Binary)

The attribute "Asperity indicator", is a binary variable (Yes or No) indicating if an area consist an asperity or not, and it was used as the classification class of the vector in the experiments conducted.

2.3 Feature Vector Extraction

For the purposes of this paper the wider area of Hokkaido region was separated in a grid by 0.1 latitude and longitude degrees. In order to ensure the robustness of the estimated b-values, the radius of every cell that had at least 30 events was increased in order to contain 50 events, using the data of the surrounding cells. The process of creating the grid was automated. Software was created in C language that composes a separate catalog for each cell of the grid and also measures the corresponding density. Every cell was labeled with the latitude and longitude of its centroid.

For all the sections where the number of events (density) was greater than 50, the b-value was calculated. In the formula, that describes the Gutenberg-Richter frequency-magnitude distribution (G-R FMD) (1), N is the accumulated number of events, M is the events magnitude, a-value indicates the total seismicity rate of the region, and the b-value constitutes the slope of the distribution and describes the ratio of small and big earthquakes in an earthquake catalog [18]. The most often used procedures to calculate the b-value of a G-R FMD is the Maximum Likelihood Estimate of b-value [19] method created by Utsu and the least square technique [20]. For the b-value estimation the Maximum likelihood method was chosen.

$$\mathrm{Log}(N) = a - b * M \tag{1}$$

The calculations were made by the software ZMAP [21]. The purpose of this application is to determine the quality of seismic data, which are included in earthquake catalogs, and also to calculate and extract useful features. The application combines many basic and useful tools for seismological research.

In our feature vector, every section with density lower than 50 was marked with b-value '?' corresponding to the WEKA's missing value symbol.

3 Machine Learning Algorithms

In total 39 classification algorithms were used to test our hypothesis. The five most effective in means of precision and recall are described below.

Random Forest is a tree classification algorithm developed by Leo Breiman [22]. This algorithm creates a forest of random trees. Random vectors are created from the training set and based on them the growth of each tree is made. The algorithm ensures

that every random vector will be unique. Finally each tree vote for the class that every testing instance will be registered and the most popular class is selected from the forest.

Ridor is an implementation of the Ripple-Down Rule learner [23] classification algorithm in WEKA. The Ripple Down Rule creates a binary decision tree different from the ordinary tree classifiers, were a decision can be reached in an interior node in contrast with standard trees were a decision can be made only in the root of the tree. All the rules-nodes are connected with a two way relation. If the premises for a node are all true then the testing subject will be asserted by this node. All these nodes are connected with if-true and if-false sub-nodes. The parental node can make a decision only when the correspondingly sub-nodes are fully in line.

Simple CART (Classification and Regression Tree) is an algorithm also developed by Leo Breiman [24]. The Cart algorithm consist a greedy algorithm where in each stage of the tree building process chooses the most discriminatory feature. To do so each time the attribute to be split is chosen be means of entropy. Finally the algorithm creates a binary decision tree.

BFTree (Best First Tree) [25] is a decision tree algorithm similar to the depth first tree algorithm. In a best-first tree an attribute is placed in the root of the tree and branches are created based on criteria. The training instances are split in subsets, one for every branch. The process is repeated for the branch with the "best" subset using only the attributes that reaches the branch. The construction process stops when all nodes are pure.

4 Experimental Process

Due to high imbalance of examples between the two classification classes (539 No and 61 Yes) the SMOTE [26] preprocess algorithm was used. This algorithm creates synthetic examples of the minority class. To do so, it uses the k nearest neighbors of every example of the minority class. The minority class was thus oversampled by 783 % in order to even the examples in both classes resulting with 539 "No" and 538 "Yes" examples.

With the two classes evenly matched experiments were conducted using all the available classifying algorithms of WEKA.

Every time the 10-folds cross validation technique was used. The available examples are randomly portioned in 90 % for training and 10 % for testing. This process is repeated 10 times. In every tested algorithm the results are derived from combining the output of both 10 experiments conduct with randomly created training and test samples.

The following parameters where used:

The RandomForest algorithm was set to generate 100 trees and every time to use all the vector's features.
In the Simple cart algorithm, the internal cross-validation was made with 5 folds and the minimal number of objects at the terminal nodes was 2.
For the rule based classification algorithm Ridor 1 fold of the data was used for pruning and 2 folds for growing the rules.

In the BFTree algorithm the internal cross-validation was made with 5 folds and all the available data were used for training set.

In Table 1 the building model time is presented for every algorithm.

Table 1. Building time for the algorithm's model in seconds.

Classifier	Time
RandomForest	0.37
SimpleCart	0.14
Ridor	0.9
BFTree	0.13
NBTree	0.17

5 Evaluation

The evaluation of the algorithms results is made by means of precision and recall. The top five algorithms are Random Forest, Simple CART, Ridor, BFtree, and NBTree. All five algorithms exhibit precision and recall higher than 0.9, Fig. 1 below displays the classification results.

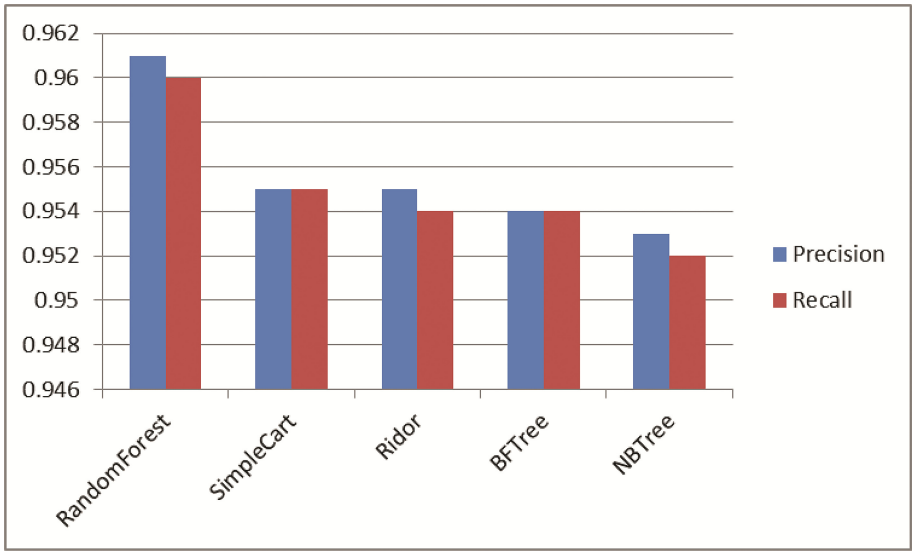

Fig. 1. Precision and recall comparison for five classification algorithms

The Random Forest algorithm really stands among the other four with precision 0.961 and recall 0.96. Simple CART, Ridor, and BFTree are really close with precision and recall ranking above 0.954. Table 2 contains precision, recall and the confusion matrix of the top five algorithms.

Table 2. Results of the top five algorithms.

Classifier	Precision	Recall	a	b	
RandomForest	0.961	0.96	509	30	**a = No**
			13	525	**b = Yes**
SimpleCart	0.955	0.955	504	35	**a = No**
			14	524	**b = Yes**
Ridor	0.955	0.954	497	42	**a = No**
			8	530	**b = Yes**
BFTree	0.954	0.954	506	33	**a = No**
			17	521	**b = Yes**
NBTree	0.953	0.952	500	39	**a = No**
			13	525	**b = Yes**

6 Conclusion

In the wider region of Hokkaido in north Japan, supervised machine learning algorithms were used to identify areas with asperity properties. The proposed feature vector consisted of geographic location information, the seismic density and b-value of the examined region. Impressive results were produced from the conducted experiments in the WEKA platform. Many classification algorithms appeared to be effective in terms of precision and recall, with the Random forest algorithm to be the most efficient with 0.961 precision and 0.96 recall, indicating that machine learning algorithms can be a useful tool for mapping asperities in space.

Future research could take into account more earthquake characteristics such as the earthquake interval time and the magnitude range. Also the described method's efficiency should be tested in other regions with well-known asperities before it is used in the field.

Acknowledgments. Fruitful discussions with Assistant Professor Katia Lida Kermanidis and Doctor Ioannis Karydis of the dept. of Informatics, Ionian University, Greece are gratefully acknowledged.

References

1. Wiemer, S., Wyss, M.: Mapping the frequency-magnitude distribution in asperities: an improved technique to calculate recurrence times? J. Geophys. Res. **102**, 115–128 (1997)
2. Takahashi, H., Kasahara, M.: Spatial relationship between interseismic seismicity, coseismic asperities and aftershock activity in the southwestern Kuril islands. In: Volcanism and Subduction: The Kamchatka Region (2013)
3. Hatzfeld, D., et al.: The Galaxidi earth-quake of 18 November 1992 a possible asperity within the normal fault system of the Gulf of Corinth (Greece). Bull. Seismol. Soc. Am. **86**, 1987–1991 (1996)

4. Park, S.C., Mori, J.: Are asperity patterns persistent Implication from large earthquakes in Papua New Guinea. J. Geophys. Res. Solid Earth. **112**, 3 (2007)
5. Dalguer, L.A., Irikura, K., Riera, J.D.: Simulation of tensile crack generation by three-dimensional dynamic shear rupture propagation during an earthquake. J. Geophys. Res. Solid Earth, **108** (2003)
6. Pulido, N.: Broadband frequency asperity parameters of crustal earthquakes from inversion of near-fault ground motion. In: 13th World Conference on Earthquake Engineering (2004)
7. Irikura, K., Miyake, H., Iwata, T., Kamae, K., Kawabe, H., Dalguer, A.: Recipe for predicting strong ground motions from future large earthquakes. In: 13th World Conference on Earthquake Engineering (2004)
8. Ozacar, A., Beck, S.L.: The 2002 Denali fault and 2001 Kunlun fault earthquakes: complex rupture processes of two large strike-slip events. Bull. Seismol. Soc. Am. **94**, 278–292 (2005)
9. Kagawa, T., Irikura, K., Somerville, P.G.: Differences in ground motion and fault rupture process between the surface and buried rupture earthquakes. Earth Planets Space **56**, 3–14 (2004)
10. Murotani, S., Satake, K., Fujii, Y.: Scaling relations of seismic moment, rupture area, average slip, and asperity size for M ~9 subduction-zone earthquakes. Geophys. Res. Lett. **40** (2013)
11. Spence, W., Mendoza, C., Engdahl, E.R., Choy, G.L., Norabuena, E.: Seismic subduction of the Nazca ridge as shown by the 1996–97 Peru earthquakes. Pure. appl. Geophys. **154**, 753–776 (1999)
12. Pulido, N., Aoi, S., Fujiwara, H.: Rupture process of the 2007 Notohanto earthquake by using an isochrones back-projection method and K-NET/KiK-net data. Earth Planets Space **60**, 1035–1040 (2008)
13. Ken-ichi, F., Daiki, I., Masayuki, N.: Discovering seismic interactions after the 2011 Tohoku earthquake by co-occurring cluster mining. Trans. Jpn. Soc. Artif. Intell. **29**(6), 493–502 (2014)
14. Otari, G.V., Kulkarni, R.V.: A review of application of data mining in earthquake prediction. Int. J. Comput. Sci. Inf. Technol. **3**, 3570–3574 (2012)
15. Panakkat, A., Adeli, H.: Neural network models for earthquake magnitude prediction using multiple seismicity indicators. Int. J. Neural Syst. **17**, 13–33 (2007)
16. Reyesa, J., Morales-Estebanb, A., Martínez-Álvarezc, F.: Neural networks to predict earthquakes in Chile. Appl. Soft Comput. **13**, 1314–1328 (2013)
17. Hall, M., Frank, E., Holmes, G., Pfahringer, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explor. Newslett. **11**, 10–18 (2008)
18. Kulhanek, O.: Seminar on b-value. In: Department of Geophysics, Charles University, Prague (2005)
19. Aki, K.: Maximum likelihood estimate of b in the formula log N = a − bM and its confidence limits. Bull. Earthq. Res. Inst. **43**, 237–239 (1965)
20. Gutenberg, B., Richter, C.F.: Frequency of earthquakes in California. Bull. Seismol. Soc. Am. **34**, 185–188 (1944)
21. Wiemer, S.: A software package to analyze seismicity: ZMAP. Seismol. Res. Lett. **72**, 373–382 (2001)
22. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
23. Gaines, B.R., Compton, P.: Induction of ripple-down rules applied to modeling large databases. J. Intell. Inf. Syst. **5**, 211–228 (1995)
24. Kalmegh, S.: Analysis of WEKA data mining algorithm REPTree, simple cart and randomtree for classification of indian news. Int. J. Innov. Sci. Eng. Technol. **2**, 438–446 (2015)
25. Shi, H.: Best-first decision tree learning. Thesis in Department of Computer Science, University of Waikato, Hamilton, New Zealand (2006)
26. Chawla, N.V., Bowyer, K.W., Hall, L.H., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)