Landslide Susceptibility Mapping in Nepal Using Spatial Feature Vectors

Markus Zechner, Muhammad Almajid, Kuy Hun Koh Yoo

Section 1: Introduction

According to the USGS (U.S. Geological Survey), the earthquake of April 2014 in Nepal and its aftershocks claimed the life of almost 9,000 people. Many of the deaths were caused by landslides which also blocked important routes for emergency response and evacuations. These kinds of disasters are not uncommon in other parts of the world and vastly underestimated as recent research shows. Landslide prediction can be a valuable tool for planning first response (Where to send first response helpers if there is no communication after an earthquake?) and emergency evacuations.

Unlike the US, countries like Nepal do not have extensive infrastructure to measure and monitor seismic occurrences that could help predict and locate disasters. For this project we try to identify high risk zones prone to landslides based on features that have been studied by research institutions (vanWesten et al., 2008).

We use expert labeled GPS locations, which occurred in Nepal as a consequence of the earthquake, as our training set. **Figure 1** shows a zoomed-in image of some of Nepal's landslides taken from Google Earth.

Figure 1: A close-up image of 7 landslide locations (unnamed yellow pins) and one village (named yellow pin)

Section 2: Related Work

Landslide hazard assessment is an important step towards landslide hazard and risk management. Several methods of Landslide Hazard Zonation (LHZ) like heuristic, semi quantitative, quantitative, probabilistic and multi-criteria decision making process are applied to predict landslides.

In last few years a change from a heuristic (knowledge based) approach to a data driven approach (statistical approach) was observed with the goal of minimizing subjectivity and increasing reproducibility. The statistical methods for LHZ are grouped into two: 1) bi-variate statistical analysis and 2) multi-variate statistical analysis. The bi-variate statistical analysis for LHZ compares each data layer of features to the existing landslide distribution (Kanungo et al. 2009). Multi-variate statistical analysis for LHZ considers relative contribution of each thematic data layer to the total landslide susceptibility (Kanungo et al. 2009). These methods calculate the percentage of landslide area for each pixel and the landslide absence. Logistic regression model, Discriminant analysis, Multiple Regression Models, Conditional Analysis, Artificial Neural Networks are commonly used methods for LHZ mapping (Wang and Sassa 2005, Ayalew and Yamagishi 2005, Guzzetti et al. 1999, Ercanglu 2005, Catani et al. 2005, Pradhan and Lee 2009, Pradhan and Lee 2010, Bui et al. 2012)

Although current research is promising in assessing critical features for the prediction of landslides, we believe there is room for improvement. Since in most situations the data is highly unbalanced, it is important to study machine learning model selection and to evaluate the prediction performance with relevant metrics.

Section 3: Dataset and Features

The locations of the landslides triggered by the earthquake that occurred in April of 2015 with a magnitude of 7.8 after Richter were obtained from the USGS. It is important to note that the reported locations are the only

locations where people reported landslides. It is more than possible that there were other landslides that have not been reported. It is therefore not surprising that most observations were made near populated areas.

Elevation, aspect ratio, slope, wetness index, NDVI (Vegetation Index), NDWI (Water Index) and curvature were used as features and were obtained from Google Earth Engine. The selection of these features was based on previous studies on landslide susceptibility based on spatial data (vanWesten et al., 2008). Nevertheless, other critical features that were considered relevant were not readily available for the area of interest such as drainage networks and distance to main faults. **Figure 2** shows individual color maps for some features. The location of the landslides are denoted by blue dots on each figure.

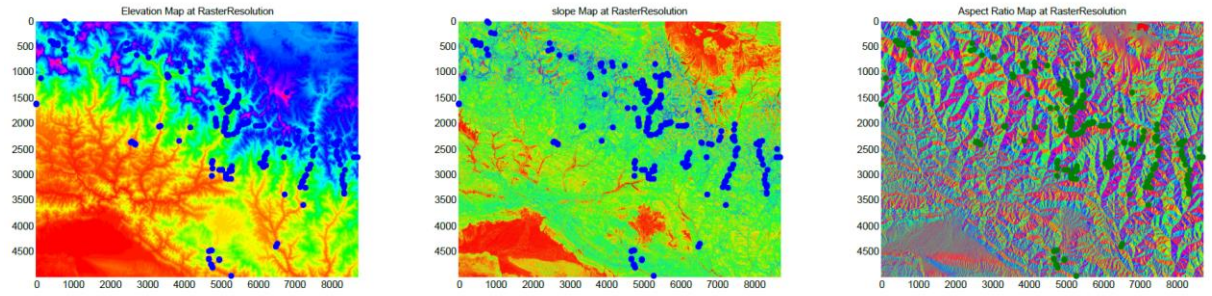

Figure 2: Post-processed feature maps obtained from Google Earth Engine

Our main challenge was cleaning and preparing the data for our chosen algorithm. In a first step, we mapped our landslide coordinates onto a 1 km grid we established (landslide grid). A grid cell that contains a landslide was assigned a value of 1. Conversely, a grid cell with value of 0 is assigned when no landslides are reported. Since all the features have different grid resolution, our main task in the second step was to cut the feature map obtained from Google Earth Engine so that it matched the area covered by our landslide grid. Given that the grid resolutions between the feature map and landslide grid are different, the last and final step was to interpolate feature properties onto our landslide grid. Nevertheless, such up/down scaling of features did involve some kind of spatial averaging which could eventually affect the descriptive strength of the feature itself (i.e. high peaks with steep slopes would appear to be average after feature post-processing).

It is important to note that for our given dataset and landslide grid, the percentage of "YES" grid cells is only 0.4%. This will become important in the selection and implementation of our learning algorithm.

For all our algorithms we used cross validation to evaluate the performance of prediction. The training data set was 70% from our data, the 30% left were used for testing.

Section 4: Methods

In this section we present a short description of the learning algorithms used in this project.

Naïve Bayes (NB)

In the NB model, the features are assumed to be conditionally independent given the responses. This is a strong assumption. Given our training set $\{(x^{(i)}, y^{(i)}); i = 1, \dots, m\}$, we can fit our parameters: $\phi_{(i|y=1)}$, $\phi_{(i|y=0)}$, and ϕ_y by maximizing the joint likelihood of the data. Using the fitted parameters, we make a prediction on a new example by picking the class which gives us the higher posterior probability.

Logistic Regression

Logistic Regression is a method used to predict probabilities for binary classification. The algorithm is defined as:

Repeat until convergence { $\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}$ (for every j) }

Where $h_\theta(x^{(i)})$ is the sigmoid function. We predict one class if the probability is above a certain threshold and the other class otherwise. The threshold can be changed based on how stringent we want to be on the data.

Support Vector Machine (SVM)

We used the usual SVM algorithm with regularization to take into account non-separable cases. In SVM, we are essentially trying to find the weights that should be multiplied by our example. The primal problem that the optimization algorithm solves is,

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

$$subject\ to\ y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, i = 1, \dots, n$$

Tree-Based Methods (CART, Random Forest, Boosting)

The Classification and Regression Trees algorithm (CART) is based on splitting the feature space recursively into binary partitions. In other words, they are piecewise constant models. In a node m, representing a region R_m with N_m observations (ICME ML Workshop, 2015),

$$class\ k(m) = \arg\max_{k} \hat{p}_{mk}$$

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in \Re_m} I(y_i = k)$$

Random forest is a large collection of uncorrelated trees that are built and then averaged. The idea is to reduce variance because the bias of averaged bagged trees is the same as that of an individual tree (Hastie et al., 2009). They're similar to CART in that they capture high-order interactions between variables and handle mixed predictors. Random forests are better than CART in that they're less prone to overfitting and because of using out-of-bag samples, cross validation is already built-in.

Finally, boosting fits additional predictors to residuals from initial predictions (ICME ML Workshop, 2015):

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda\, \hat{f}^b(x), \qquad \lambda: Learning\ Rate$$

Section 5: Experiments/Results/Discussion

Defining Error Metrics for Imbalanced Data Sets

Many performance metrics have been proposed for imbalanced data sets but all of them are based on the confusion matrix. When data sets have classes with significantly different sample sizes (only 0.4% of our data indicate landslides) the standard definition of accuracy is not applicable any more – for example a classifier which predicts all grid cells with 'no landslide' would have a very good accuracy but obviously a very bad prediction performance.

A much better performance indicator are 'sensitivity' and 'specificity' which are often called true positive (TP) rate and true negative (TN) rate, respectively.

Furthermore, the **Area under the Receiver Operating Characteristic Curve (AUC)** is also a very indicative measure for the prediction capabilities in an imbalanced data set. The Receiver Operating Characteristic (ROC) plots the 'true positive rate' versus the 'false positive rate' with a varying classification threshold (by default it is 0.5) – the higher AUC is the better the performance of the algorithm. A high slope in the beginning indicates a very good prediction performance – in other words – by changing the threshold we can gain many more 'true positives' without scarifying 'false positives', when the curve bends over we still can increase the 'true positives' but also need to except more 'false positives'.

We use the AUC and the ROC as the error metrics for our project as we believe they are the most intuitive ones.

Learning Algorithms Results: Base Case

We began our project testing the simplest algorithms (Naïve Bayes and Logistic Regression) with two features (elevation and slope) to establish a baseline. Given our imbalanced data set, we thought that Naïve Bayes would be a good match as it is known to converge quicker than discriminative models and it requires less training data to converge. Furthermore, the Naïve Bayes conditional independence assumptions apply very well for our problem statement since a landslide is unlikely to have an influence on its surroundings since the grid-block size is one square kilometer.

Both, Logistic Regression and Naïve Bayes have the advantage of a probabilistic interpretation, unlike decision trees or SVMs. This probabilistic framework makes it easy to adjust the classification thresholds which are crucial in imbalanced data sets. In the first trial with two features, Naïve Bayes showed a good performance whereas Logistic Regression performed rather poorly (**Figure 3**).

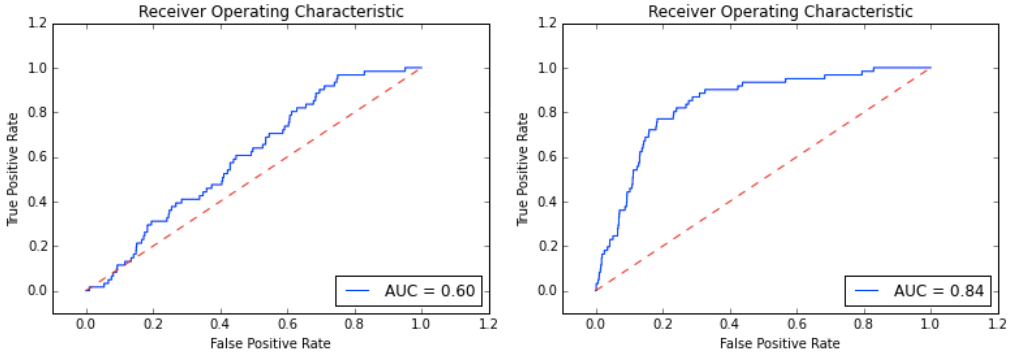

Figure 3: ROC curves from implementation of Logistic Regression (LEFT) and Naïve Bayes (RIGHT)

Surprisingly the SVM algorithm did not give a good result. Since SVM, like Logistic Regression, is based on a linear separator we expected the linear SVM to have a similar performance as Naïve Bayes, but the results did not support our initial hypothesis. After testing with the linear separator we decided to apply the kernel trick to allow for non-linear separators – we tested the polynomial, Gaussian and the exponential kernel on our training test but without success – the Area under the Curve (AUC) stayed constant at 0.29.

Furthermore, we tested for the regularization parameter C (i.e. a penalty parameter of the error term) in the range of 0.01 to 100 – the upper bound showed a very small (0.01) improvement with respect to AUC. Since our dataset is highly imbalanced, we assigned weights inversely proportional to class frequencies in the input data. Nevertheless, this tuning did not improve our results with respect to AUC. We would have liked to perform further testing on the hyper parameters of the SVM but due to enormous runtimes we could only test for lower and upper bounds without considering any possible interactions.

Finally, we explored tree-based methods for our data set, mainly for two reasons. Decision trees are easy to interpret and it is possible to identify critical features for the problem. They can often provide good insights into the problem and enhance the understanding of the problem itself. Decision trees also often perform well on imbalanced datasets. The splitting rules that look at the class variable used in the creation of the trees, can force both classes to be addressed. A major disadvantage is that they easily over fit, but that's where ensemble methods like random forests or boosted trees come in.

In this baseline case we used elevation and slope as features. Furthermore, gradient boosting performed very well (AUC=0.85) whereas the forest tree algorithm performed poorly with an AUC of 0.5.

Learning Algorithms Results: Full Feature Set

When the full set of features were considered, the best results were obtained using Logistic Regression and Naïve Bayes. In both cases, we used thresholds to be more stringent on deciding whether there is landslide or not. **Figure 4** shows the ROC curve, the confusion matrix and the AUC vs. Training set size for both algorithms. In both AUC vs. Training set size plots, we observe that the performance increases with increasing training set size, equivalent to a decrease in error in the standard learning curve. For large training set sizes a decrease in performance is noticed – we believe that this is due to the fact that if we have a large portion of training example

only a few positive example are left for prediction – only a few false predictions could lead to a large error – this also explains the large variance on the AUC.

The confusion matrix shows that for both algorithms, we were able to predict the presence of a landslide with an accuracy of about 85%.

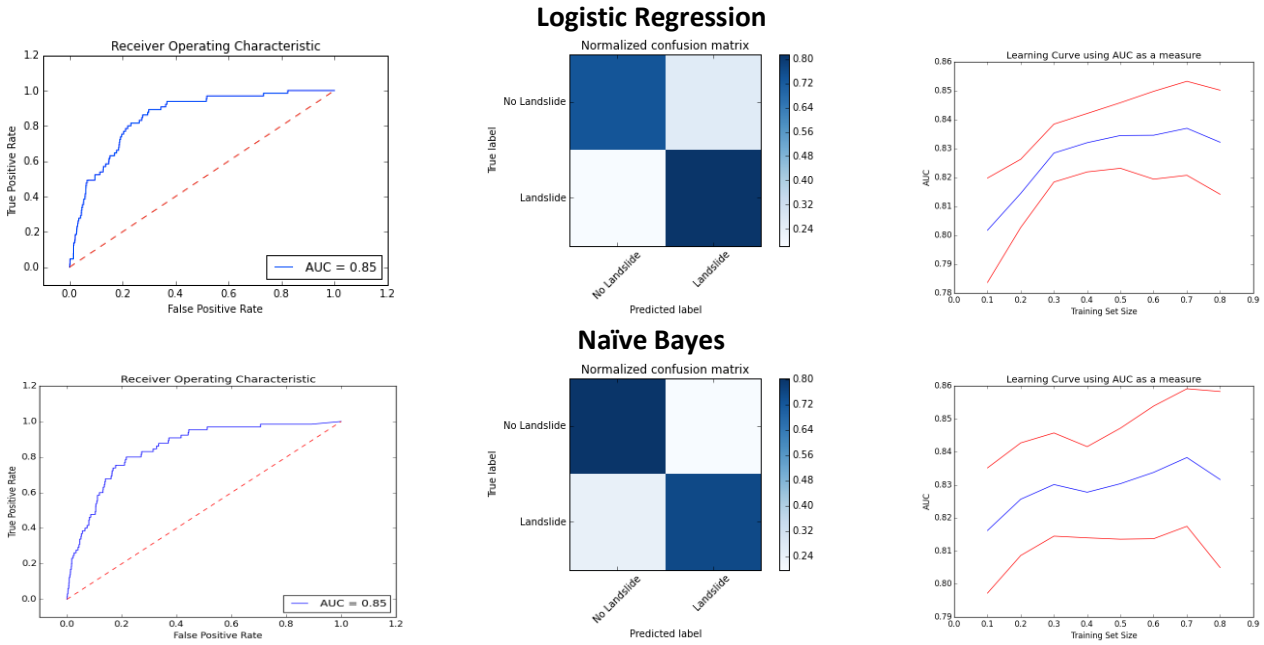

Figure 4: Results of Logistic Regression and Naïve Bayes using 7 features. From left to right, ROC Curve, Confusion Matrix and Area under Curve vs. Training Set Size (Standard Deviation shown in Red)

Section 6: Conclusion & Future Work

In conclusion we can say that it is very hard to 'debug' a problem with a highly imbalanced data set and strategically improve it with respect to variance and bias. Testing different algorithms showed that Naïve Bayes and Logistic regression perform the best when having all available features included and adjusting the threshold for probability in the case of Naïve Bayes. We were able to predict ~85% of the landslides.

Finally, it is very hard to predict the exact location at which a future landslide will occur given the unbalanced data set. Nevertheless, the probabilistic interpretation provides a useful tool to generate spatial probabilistic hazard maps (e.g. what areas are landslides most likely to occur?).

In terms of future work, we believe that including more critical features such as the distance to faults, distance to rivers, and formation deformation from InSAR satellite data is worth exploring. Other unsupervised learning algorithms can also be tested such as anomaly detection because our data set is highly imbalanced.

Section 7: References

[1] Ayalew, Lulseged, and Hiromitsu Yamagishi. "The application of GIS-based logistic regression for landslide susceptibility mapping in the Kakuda-Yahiko Mountains, Central Japan." Geomorphology 65.1 (2005): 15-31.

[2] Bui, Dieu Tien, et al. "Landslide susceptibility assessment in the Hoa Binh province of Vietnam: a comparison of the Levenberg–Marquardt and Bayesian regularized neural networks." Geomorphology 171 (2012): 12-29.

[3] Catani, F., et al. "Landslide hazard and risk mapping at catchment scale in the Arno River basin." Landslides 2.4 (2005): 329-342.

[4] Ercanoglu, M. "Landslide susceptibility assessment of SE Bartin (West Black Sea region, Turkey) by artificial neural networks." Natural Hazards and Earth System Science 5.6 (2005): 979-992.

[5] Guzzetti, Fausto, et al. "Landslide hazard evaluation: a review of current techniques and their application in a multi-scale study, Central Italy." Geomorphology 31.1 (1999): 181-216.

[6] Hastie, Trevor, Robert Tibshirani, and J. H. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer, 2009. Print.

[7] Kanungo, D. P., et al. "Landslide susceptibility zonation (LSZ) mapping—a review." J South Asia Disaster Stud 2.1 (2009): 81-105.

[8] Pradhan, Biswajeet, and Saro Lee. "Landslide risk analysis using artificial neural network model focusing on different training sites." Int J Phys Sci 3.11 (2009): 1-15.

[9] Pradhan, Biswajeet, and Saro Lee. "Regional landslide susceptibility analysis using back-propagation neural network model at Cameron Highland, Malaysia." Landslides 7.1 (2010): 13-30.

[10] "Precision and Recall." Wikipedia. Wikimedia Foundation, n.d. Web. 11 Dec. 2015.

[11] vanWesten, C. J., Castellanos, E., and Kuriakose, S. L. (2008). Spatial data for landslide susceptibility, hazard, and vulnerability assessment: An overview. Engineering geology, 102(3):112–131.

[12] Wang, H. B., and K. Sassa. "Comparative evaluation of landslide susceptibility in Minamata area, Japan." Environmental Geology 47.7 (2005): 956-966.

[13] ICME Machine Learning Workshop, August 24-28 2015