Predicting Coral Reef Regimes from Human and Natural Influences

Tiffany Cheng
Statistics Department
Stanford University
tiffc@stanford.edu

Austin Murphy
Statistics Department
Stanford University
amurphy5@stanford.edu

Sallie Walecka
ICME
Stanford University
swalecka@stanford.edu

Abstract

We perform a multi-class prediction of coral reef regime types based on anthropogenic and environmental features. We use several models - logistic regression, SVM, decision trees - with different architectures and features and compare performance. We achieve a best cross-validation micro-average F1 score of 0.70, up from 0.60 in our baseline model. We complete error analysis and PCA to understand the limits of our models.

1. Introduction

Researchers have established that climate change and other human-driven factors have been putting pressure on coral reefs around the world. As coral reef conditions change, the reefs transition into alternative regimes, including some regimes that are considered as "degraded" [1]. Recently, Jouffray J-B *et al.,* 2019 took a novel approach of using Boosting Regression Trees to distill the impact of interactions between both human, biotic, and abiotic conditions on the coral reef regimes in different locations in Hawaii [3]. This research used spatially-collected data, which included anthropogenic and environmental predictors believed to be more directly related to coral regimes than proxies used in previous studies [4].

While this paper aimed to understand the most influential predictors for each of the four distinct reef regimes, our study focuses on analyzing how prediction accuracy of the coral regimes changes depending on the type of classification model used and its complexity. We take both continuous and factor inputs measuring human activity (e.g. new development) and environmental influences (e.g. wave power) and use them to predict the corresponding coral reef regime using a variety of classification methods.

We are interested in predicting the regimes because an accurate model will allow researchers to get a better sense of the distribution of regimes in locations beyond Hawaii. More importantly, it will also help researchers identify regimes that need immediate attention and restoration. While regimes 2, 3, and 5 describe various types of healthy habitats, regime 1 has "characteristics of a degraded reef, i.e. low fish biomass, low hard coral cover and high algae cover."[1] Therefore, accurately identifying reefs that are in this damaged regime, including those that have newly transitioned into this regime, will help coral reef management groups prioritize their interventions.

2. Previous Methodology

A group of scientists obtained funding to create an in-depth data set of Hawaiian coral reefs [4]. Five coral regimes were derived from a Gaussian Mixture Model that used features measuring fish, coral, algae and other coral reef organisms [1]. These regimes can be thought of as different types of reefs that one might come across while snorkeling (e.g., more fish/coral, or less fish/high algae cover). The authors investigated the frequency of transitions between the five regimes to understand the dynamics. Apart from the columns used for the GMM, additional metrics of the reef such as wave power, sedimentation, and commercial fishing were captured. Previous research had shown that the effect of natural drivers on coral reef structures were non-linear in nature [2]. Therefore, to get a high performing yet interpretable result, boosted regression trees were used. A boosted regression tree was fitted for each regime instead of one multinomial model. After the initial fitting, more research was done to fit over different resolution of the location data, as some measurements were taken over different scales (100m vs 4000m) [3].

3. Data set

The data set was collected to understand spatial human and natural drivers for coral reefs [4]. Using the GMM from [1], regimes were identified at different locations across the Hawaiian islands. Regime 4 was highly variable, so in order to use high-confidence data, all data from regime 4 was excluded. Additional data points were dropped if regimes could not be predicted for them with high probability [3].

Our data set has 620 observations and 20 predictors. The

predictors are mainly of two types: anthropogenic (effluent, new development, commercial fishing, etc.) and biophysical (wave power, depth). The complete data dictionary can be found in Table 1 of [3]. We noticed there were many 'NA' in the column for 'complexity'. Therefore, we imputed the missing data points in different ways and settled on the best one (see Section 3.3).

We used all 20 predictors from the data set. The ecological features from the GMM were kept for reference, but were not used as features and are not included in this count. As described later on in the Feature Engineering section, we also added 5 additional predictors using domain knowledge.

3.1. Train/Test Split

We split our data 80/20 into train (n=496) and test (n=124) sets. The test set was held out during training and only used once with our final model. We used 5-fold cross-validation on our training set to get our estimated validation error and to tune our hyper-parameters. After finding the best model via cross-validation, we train our model on the entire training set, evaluate it once with the test set, and report the micro-average F1 score.

3.2. Data Analysis

Initial exploratory data analysis regarding correlation and collinearity for this data set was documented in [3] and the author's (Jouffray J-B) github repo. We've forked the repo and included our sanity check of the data and additional visualizations at https://github.com/salliewalecka/Hawaii_RegimesPredictors/blob/master/notebooks/EDA.ipynb. Figure 1 shows that the areas randomly split into the test and train set have relatively similar distributions across the islands. We also ensured that the distribution of regimes in the train-test split were comparable. Finally, we ensured that the Variance Inflation Factors (a measure of collinearity) were not too big, and Figure 2 replicates the correlation analysis done in [3].

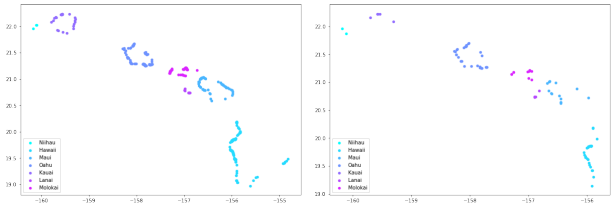

Figure 1. Train (right) vs Test (left) split of data

3.3. Complexity Column Imputation

The data set had NA values in two columns, 'Complexity' and 'Depth', which are both readings of the seafloor. The NAs accounted for 18% of the observations, so rather

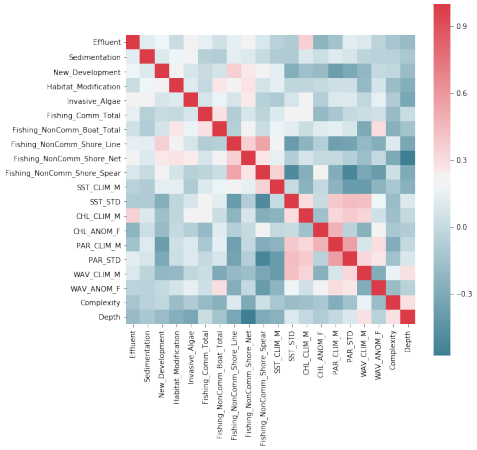

Figure 2. Correlation matrix for predictors

than discarding the observations, we decided to estimate them. We fit three estimation models: a lasso regression model, a k-means clustering estimate, and a mean imputation of the 'Complexity' values. Both the lasso and k-means estimates had lower RMSE than simply using the mean. Because the k-means estimation gave the lowest RMSE, we used it as our imputation.

4. Method

4.1. Metric: Micro-averaging

We choose micro-averaging of F1 score as our performance metric for our multiclass classification problem as an alternative to accuracy. Micro-averaging the F1 scores is used in order to account for the class imbalance of coral regimes. The micro-average F-score is calculated by taking the harmonic mean of the micro-average of precision:

$$\frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + FP_i}$$

and of recall:

$$\frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + FN_i}$$

where TP, FP, and FN stand for True Positives, False Positives, and False Negatives, respectively, and C is the number of classes.

4.2. Feature Engineering

Since the initial fitting of logistic models did not achieve the desired performance, we iterated and tried constructing features that are non-linear in the predictors. In a previous study [2], authors studied how similar predictors on coral reef regimes at Palmyra Atoll exhibited non-linear responses to the predictors and one of the takeaways was the

non-linearity of wave forces on the reef. Therefore, we create interaction terms between wave power, seafloor complexity, and depth. We added another interaction between invasive algae and high irradiance as well another for measuring the amount of anomalous events in the two anomaly predictors. In the end, we added 5 features to our existing 20. Further explanation is available in the supplemental materials feature engineering notebook.

4.3. Model Training

We train our models using 5-fold cross-validation on the training set. For validation F1 scores, we report the mean validation F1 scores across all folds. For models with tunable hyper-parameters, we again use 5-fold cross-validation across a grid of values to fine-tune those hyper-parameters.

We train and iterate several variants of the following models:

4.3.1 Logistic Regression

The multinomial logistic regression loss function with L2 regularization is given as follows:

$$L(\theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K} 1\{y^{(i)} = k\} \log P(y^{(i)} = k|x^{(i)};\theta) + \lambda||\theta||^2$$

where θ corresponds to the predictor weights, k is the class indicator, and $y^{(i)}$ is the estimate of the class.

4.3.2 SVM

The optimization problem for a non-separable SVM problem is as follows:

$$\begin{aligned} \min_{\gamma,\omega,b} \ & \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \ & y^{(i)}(\omega^T x^{(i)} + b) \geq 1 - \xi_i, i = 1,\ldots,n \\ & \xi_i \geq 0, i = 1,\ldots,n. \end{aligned}$$

where ξ_i corresponds to the size of the margin, $y^{(i)}(\omega^T x^{(i)}+b)$ is the functional margin, and C controls the relative weighting between the optimization goals of minimizing $||\omega||^2$ and ensuring that most examples have a functional margin of at least 1.

4.3.3 Decision Trees

Decision trees look to find the optimal split across all variables that reduce the classification error in the model. The structural model for a classification decision tree is as follows:

$$\hat{c}(x) = \sum_{m=1}^{M} \hat{c}_m 1\{x \in R_m\}$$

where $\hat{c}(x)$ is the predicted class, R_m are the mutually-exclusive regions in the predictor space, and $1\{x \in R_m\}$ is an indicator of whether the input feature x is in the respective region.

A decision tree model searches for the best split across the predictor variables. The best split is determined by the decrease in classification error due to the new split. Because the regions that carve up the input space are mutually-exclusive, this difference in misclassification error can be computed within the region that the split is made in.

5. Results

5.1. Baseline Model

Our baseline model is a vanilla multinomial logistic regression model which excludes all observations with NAs. This model attained a 0.60 micro-averaged F1 validation score using cross-validation. We then iteratively improve this model, and others, and compare their performance.

5.2. Logistic Regression

We train several variants of logistic regression. We choose to iterate the multinomial model instead of the 1-vs-rest in order to get predictions for all regime types with one classifier. We decided to test several versions of the model: 1) dropping all rows that contained NA values, 2) filling in the NA values with the mean, 3) filling in the NA values with the imputed score from k-means clustering estimation, 4) an optimally regularized model, and 5) a model with the additional engineered features. The performance of our models are given in the Table 1.

Logistic Iterations	Train F1	Val F1
Dropped NA*	0.68	0.60
Mean NAs	**0.71**	**0.66**
Imputed NAs	0.69	0.64
Regularized	0.69	0.62
Feature Engineered	0.66	0.62

Table 1. Train and Accuracy Scores for Iterative Logistic Regression Models. *the baseline model.

As reported in Table 1, the validation F1 scores give no clear indication of a best logistic model. The model with the mean imputation of the NA values outperforming all other models is likely due to noise.

5.3. SVM

We fit four different variants of the SVM. We fit a Radial Kernel SVM both with and without the additional engi-

neered features, as well as a Polynomial Kernel SVM, both with and without the additional features.

Including the additional features in our models improved performance. The best results are given in Table 2.

5.4. Decision Trees

We fit four different types of decision trees with cross-validation: a Random Forest with and without the engineered features, and Gradient Boosted Trees, with and without the engineered features.

We found that including the engineered features increase the performance of the models. The specific F1 values are reported in Table 2.

6. Model Comparison

Table 2 summarizes the performance of our models, with both the Train and the Validation micro-averaged F1 score reported.

	Model	Train F1	Val F1
Logistic	Baseline	0.68	0.60
	Mean NA	0.71	0.66
	Imputed NA	0.69	0.64
	Regularized	0.69	0.62
	Feature Engineered	0.66	0.62
SVM	Radial Kernel	0.79	0.65
	Polynomial Kernel	0.71	0.61
Decision Tree	Random Forest	1.00	**0.70**
	Gradient Boosted Tree	0.83	0.69

Table 2. Comparison Table of all Models Performances using micro-averaged F1 score

Our Logistic Regression and SVM models are comparable in their performance. Decision Trees, however, outperform the other models, particularly when the engineered features are added to the model. The two best performing models on our data, using cross-validation, were the Random Forest and the Gradient Boosted Tree, both with the additional features included. These models reported a 0.70 and 0.69 F1 validation score, respectively.

This difference in performance between the decision trees and the linear models can be interpreted in different ways. Decision trees offer greater flexibility in a way, as they carve up the input space into discrete regions with corresponding predicted values. This increased flexibility allows the model to make piece-wise constant predictions of the response variable. The results from this study suggest that the true function we are approximating is better fit with this discrete piece-wise model.

7. Error Analysis

After imputing NAs and adding a regularization term to our logistic regression, we conducted an error analysis to better understand our model's results and limitations. Overall, we noticed that the model's percentage of correct classifications increased with its prediction confidence (see the calibration plot in Figure 3).

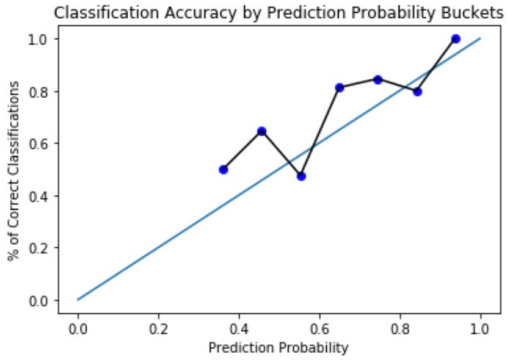

Figure 3. Calibration Plot for Logistic Regression

When we looked at the top 20% of observations that the model was most confident in predicting, they were almost all classified correctly. There were observations from all 4 regimes in this group, although the percentage of Regime 1 observations that qualified (i.e., 27% of Regime 1 observations from the validation data) was slightly higher than the percentages from other regimes. This high confidence is encouraging as R1 is the "degraded" regime that we are particularly interested in identifying. Upon digging further into the predictor values for this subgroup, we noticed trends that may explain the model's higher confidence. For example, Regime 1 examples in this subgroup had especially high values for annual catch by fisheries, which heuristically aligns with the regime's reputation as "degraded".

We also looked at the top 20% of misclassified observations that the model was most confident in predicting, and noticed that a disproportionate number came from Regimes 3 and 5: 38% of Regime 5 observations fell into this group and likewise 20% of Regime 3 observations. Furthermore, if we look at the confusion matrix in Figure 4 for the logistic regression model, we can see that that the model often mixed up Regimes 3 and 5. As classification of Regime 1 is of higher importance, we were less concerned about these errors, although they do hurt our overall metric.

7.1. PCA

In order to understand why our model performance couldn't seem to break 70% F1 micro-averaging, and why some of our models predicted Regimes 1 and 2 better than Regimes 3 and 5, we reduced the dimensions using Princi-

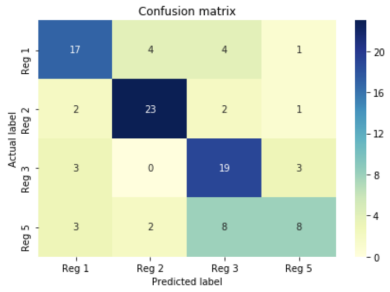

Figure 4. Confusion Matrix of Logistic Regression

pal Component Analysis. By projecting the 25 dimensional dataset onto a 2-dimensional plot, we wanted to see how the regimes are clustered, as well as find if Regimes 3 and 5 were closer in the projected space than Regimes 1 and 2. The plot can be seen in Figure 5.

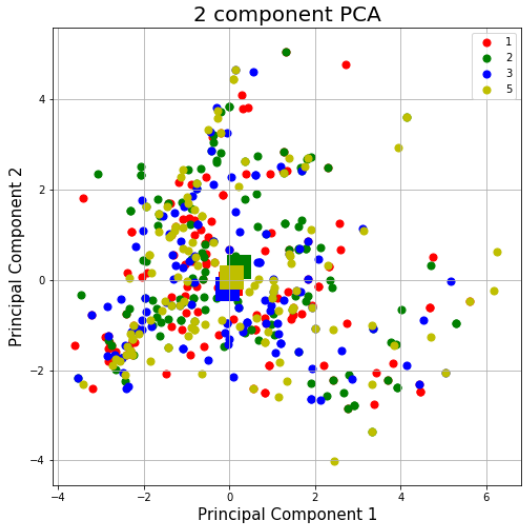

Figure 5. Plot of First 2 Principal Components with Class Means. The observations and class mean centers are not well separated.

We were surprised to see in the top 2 principal components that the classes are not well separated at all, and the class means are all quite similar. This confirmed that linear combinations of our data's predictors that maximize the variance are not all that helpful in predicting the classes.

7.2. Best Model

After training our best model, the Random Forest with engineered features, on the entire training set, we computed predictions for the test data set for our final estimate of our model's performance. We obtained a test error of 0.70, which is reassuring as it shows that we did not overfit on our training data.

The confusion matrix of our Random Forest Model test set predictions is given in Figure 6.

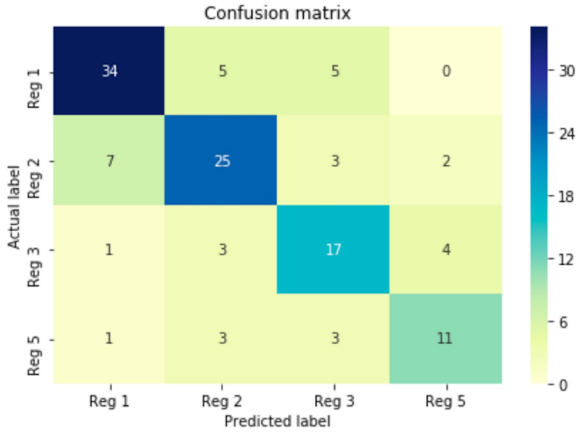

Figure 6. Confusion Matrix of the Best Model (Random Forest) the Test Data

We are encouraged from a few takeaways from the confusion matrix. First, the confusion matrix is diagonally heavy - the most frequent predictions are true estimates different classes. Some interesting findings are that while most of our models often misclassified coral observations from Regime 3 and 5, this model most often confused Regime 1 and 2. This is of concern because we previously saw models that predicted class 1 most accurately as desired, but this model reverses that trend.

The confusion matrix also shows that there are misclassifications in almost every possible way between the 4 classes. The model performs much better than chance, but seems to confuse coral regimes with one another somewhat interchangeably.

8. Conclusion/Future Work

One takeaway from this analysis is that the features used to predict the classes are not strongly related to the coral regimes. This became clear after our models consistently scored between 0.65 and 0.70, and finally when we found that the classes are not well separated in the principal components space. However, our best model seems to generalize well as it achieves a comparable micro-average F1 score (i.e. 0.70) in both the validation set and the test set.

One way of possibly finding more predictive power from these variables is to engineer additional features beyond the 5 we created with help from domain experts. Completely new predictors from joining to other data sets may be needed for increased performance too.

Finally, additional error analysis can be done on the best model to offer insight on why the model confuses the classes with one another, although this may be sufficiently explained by the PCA findings.

8.1. Contributions

Austin - Train/Test split, imputation of 'complexity' column, exploratory data analysis, tree based approaches, PCA.

Sallie - Deeper dive into previous methodology of authors referenced. Feature engineering, cross-validation for hyperparameter selection, and results synthesis.

Tiffany - Sanity checking of data analysis done by [3], exploratory data analysis, and initial baseline models, error analysis, SVM.

References

[1] D. et. al. Combining fish and benthic communities into multiple regimes reveals complex reef dynamics. *Scientific Reports*, (1):16943. 1

[2] G. J. et. al. Coral reef benthic regimes exhibit non-linear threshold responses to natural physical drivers. *Marine Ecology Progress Series*, 522:33–48, 2015. 1, 2

[3] J. J.-B. et. al. Parsing human and biophysical drivers of coral reef regimes. *Proceedings of the Royal Society B: Biological Sciences*, 1896:286, 2019. 1, 2, 6

[4] W. L. M. et. al. Advancing the integration of spatial data to map human and natural drivers on coral reefs. *PLOS ONE*, 13(3):1–29, 03 2018. 1

[5] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[6] J. D. Hunter. Matplotlib:a 2d graphics environment, computing in science engineering, 2007–. [Online; accessed Jun 12 2019.

[7] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed Jun 12 2019].

[8] W. McKinney. Data structures for statistical computing in python. In S. van der Walt and J. Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

[9] T. Oliphant. NumPy: A guide to NumPy. USA: Trelgol Publishing, 2006–. [Online; accessed Jun 12 2019].

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.