Use Satellite Data to Track the Human Footprint in the Amazon Rainforest

Genmao Shi
geshi@eng.ucsd.edu

Zhuoran Gu
zhg050@eng.ucsd.edu

1. Summary

Deforestation has become a severe issue in the past decades, resulting in reduced biodiversity, habitat loss, climate change, and other devastating effects. And deforestation in the Amazon Basin accounts for the largest share every year. Better data analysis about the location of deforestation and human encroachment on forests can help governments and local stakeholders respond more quickly and effectively. This project aims to label the satellite image chips of Amazon forest with atmospheric and various classes of land cover/land use.[6] We proposed two main methods: XGBoost and convolutional neural networks and achieved precisoin of 96.3% and F2 score of 0.90.

Based on our results and experiment, we can see that the CNN outperforms XGBoost for this problem, which indicates that the CNN is a powerful tool for visual classificaiton or recognition. But XGBoost is really a handful and fast training model. Besides, we think that use model ensemble and effective image pre-processing may improve our prediction further.

2. Introduction

Every minute, the world loses an area of forest the size of 48 football fields. Deforestation of Amazon Basin accounts for the largest share.[6] Tracking changes in the forest and differenciation human cause or natural cause for deforestation can help people stop deforestation and protect the earth. With the technology developing, hight resolution imagery has already been applied to detect small-scale forest degradation; however, robust methods to diffrenciate human encroachment and natrual factors has not been proposed. In this project, we are given a series of image segments of Amazon Basin and try to label these images with multiple tags accurately and efficiently.

2.1. Problem Formulation

There are 17 labels for the imagery in total, consisting of

- four weather labels: clear, partly cloudy, haze, cloudy;
- six land labels: primary, agriculture, water, cultivation, habitation, road;
- seven rarer labels: slash burn, conventional mine, bare ground, artisinal mine, blooming, selective logging, blow down.

Since each image chip may have multiple labels, like 'partly cloudy' and 'primary', this problem can be defined as a multi-class classificaiton problem opposed to standard multi-class classificaiton problem. We investigate the methods by solving the multi-label classification problem:

$$\hat{\mathbf{y}}_i = f(\mathbf{x}_i) \qquad i = 1, 2, ..., N \tag{1}$$

where N is the number of images in the dataset, $\hat{\mathbf{y}}_i$ is a 1×14 vector containing 1s and 0s, $f : \mathbb{R}^m \rightarrow \mathbb{R}^{14}$ is the decision function, $\mathbf{x}_i$ is the ith images with m features. We label $\mathbf{x}_i$ as 1s and 0s to represent if the image contains the specified tag or not. For example, if a image chip is clear and primary, then the true label should be $[1, 0, 0, 0, 1, 0, 0, ...0]$, and any class vector $\hat{\mathbf{y}}_i$ different from this will be regarded as incorrect.

2.2. Data Overview

The chips for this project were derived from Planet's full-frame analytic scene products using our 4-band satellites in sun-synchronous orbit (SSO) and International Space Station (ISS) orbit. The set of chips for this competition use the GeoTiff format and each contain four bands of data: red, green, blue, and near infrared. Each of these channels is in 16-bit digital number format. The imagery has a ground-sample distance (GSD) of 3.7m and an orthorectified pixel size of 3m. The data comes from Planet's Flock 2 satellites in both sun-synchronous and ISS orbits and was collected between January 1, 2016 and February 1, 2017.[5] Limited by our computer capability, we used the JPEG format image chips which only contains three channels(red, green and blue), the GeoTiff format dataset may provide more information though. The training dataset consisted of 40479 labeled files, which implies that this problem can also be defined as a supervised-learning classification. The test dataset consisted of 61191 files.

Figure 1. Sample chips and their labels

3. Data Engineering

3.1. Data Analysis

Some data exploration and analysis have been done before classificaiton. Given the histogram of labels in Figure 2, we can see that the label "primary" and "clear" have the hightest proportion of labels.

The co-occurence matrix can provide a lot of information for multi-label classificaiton challenge. Figure 3 shows heat map of co-occurence matrix between different labels. If we zoom out the map, we can find that each image segments can only have one weather labels, but the land labels and rarer labels may overlap. And some pairs of labels have trend to occure together, like primary and argriculture, argriculture and water.

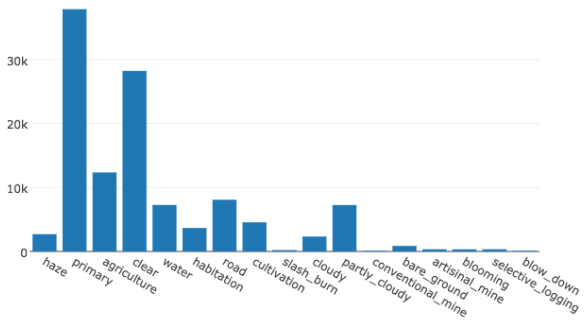

Figure 2. Histogram of labels distribution[1]

3.2. Feature Selection

According to previous reported image classification solutions, we have selected 7 classic statistic properties of 3

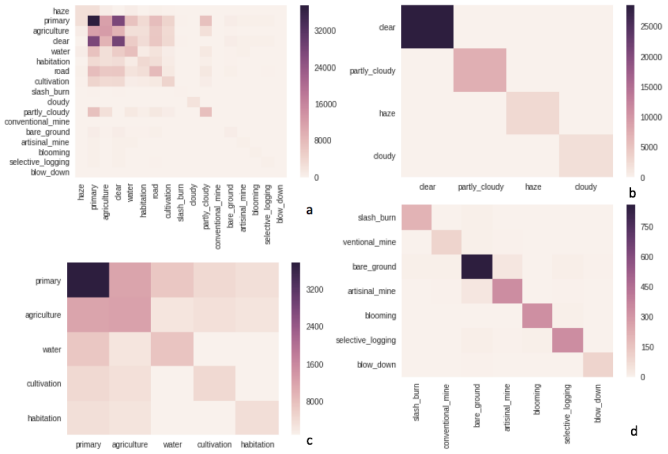

Figure 3. Heat map of co-ocurrence matrix[7]

channels (RGB) as features, i.e.

- mean
- standard deviation
- max_value
- min_value
- trim_mean (pruned top 20% and least 20%)
- kurtosis: The kurtosis is the fourth standardized moment, defined as [2]

$$Kurt[X] = \frac{\mu_4}{\sigma^4} = \frac{E[(X-\mu)^4]}{(E[(X-\mu)^2])^2} \quad (2)$$

where μ_4 is the fourth central moment and σ is the standard deviation. The kurtosis is a measure of the "tailedness" of the probability distribution

- skewness: The skewness of a random variable X is the third standarized moment γ, defined as [2]

$$\gamma = E[(\frac{X-\mu}{\sigma})^3] \tag{3}$$

where μ is the mean and σ is the standard deviation. The skewness is a measure of the asymmetry of the probability distribution about its mean.

There are 21 features in total for each image segment.

4. Methods

We implemented two methods for classification and compare their results.

4.1. XGBoost

The first method we have used is based on XGBoost[3]. XGBoost is short for extreme gradient boosting, and proposed based on this original model. The model of XGBoost is tree ensemble, which is a set of classification and regression trees (CART). We classified the chips into different leaves, and assigned them the score on the corresponding leaves. A little bit different from decision tree, a CART leaf only contains the decision values. Since one single tree is not strong enough, we summed the prediciton score of multiple trees together to get the final results. The model can be represented as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \qquad f_k \in F \tag{4}$$

where K is the number of trees, f_k is the funciont in the functional space, F is a set of all CARTs. Therefore, the objective function is:

$$obj = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{5}$$

where l is the loss function, and Ω is the regularizaiton to evaluate how complexity the model is. For the training part, what we need to learn are those functions f_i containing the tree structure and leaf scores. To learn the ensemble trees at once is difficult; instead, we use addictive strategy: fix what we have learned in the previous steps, and add one tree at each step. So, the prediciton value at step t can be written as:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{6}$$

The objective function at step t is:

$$\begin{aligned} obj^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} +\Omega(f_i) \\ &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{i=1}^{t} +\Omega(f_i) \end{aligned} \tag{7}$$

Here, we implement a model where logistic function as the loss function, and L2 norm as the regularizaiton. To train the mode, we first set the learning rate as 0.1 to tune the approximate optimal parameters, and then shirink the learning rate to 0.02 to finetune our model and achieved more accurate prediction.

4.2. Convolutional Neural Networks

XGBoost achieves pretty good results in a fast training without tricky method. Since this is a image classification problem, we decide to improve the performance of the model further by CNN[4] and tried to get a better results. Unlike ordinary neural networks, CNN makes the explicit assumption that the input are images, which allows us to encode certain properties that make the forward function more efficient to implement and vastly reduce the amount of parameters in the network.

Layer (type)	Output Shape	Param #
conv2d_27 (Conv2D)	(None, 32, 32, 32)	896
conv2d_28 (Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d_14 (MaxPooling	(None, 16, 16, 32)	0
dropout_18 (Dropout)	(None, 16, 16, 32)	0
conv2d_29 (Conv2D)	(None, 16, 16, 64)	18496
conv2d_30 (Conv2D)	(None, 16, 16, 64)	36928
max_pooling2d_15 (MaxPooling	(None, 8, 8, 64)	0
dropout_19 (Dropout)	(None, 8, 8, 64)	0
conv2d_31 (Conv2D)	(None, 8, 8, 128)	73856
conv2d_32 (Conv2D)	(None, 8, 8, 128)	147584
max_pooling2d_16 (MaxPooling	(None, 4, 4, 128)	0
dropout_20 (Dropout)	(None, 4, 4, 128)	0
flatten_6 (Flatten)	(None, 2048)	0
dense_9 (Dense)	(None, 256)	524544
dropout_21 (Dropout)	(None, 256)	0
dense_10 (Dense)	(None, 17)	4369

Total params: 815,921.0
Trainable params: 815,921.0
Non-trainable params: 0.0

Figure 4. Architecture of CNN

Due to the limitation of speed, our network only have 3 stages and each stage contains:

- Two convolutional layers: compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume.
- RELU layer: apply an elementwise activation func-

tion, such as the max(0,x) thresholding at zero. This leaves the size of the volume unchanged.

- Pooling layer: perform a downsampling operation along the spatial dimensions (width, height).
- Drop out layer: avoid overfitting
- Fully-connected layer: a fully-connected layer to compute the class scores

The whole architecture of CNN is [INPUT-(CONV-CONV-POOL-DROP OUT)×3-FULLY CONNECTED]

Figure 5. Samples of prediction

5. Results

Method	F2 score
XGBoost	0.88221
CNN	0.90003

Table 1. F2 score of two methods

The final prediction is evaluated by mean $F2$ score, which measures the accuracy with precision p and recall r. The F2 score is given by

$$(1+\beta^2)\frac{pr}{\beta^2 p + r} \tag{8}$$

where $p = \frac{tp}{tp+fp}$, $r = \frac{tp}{tp+fn}$, $\beta = 2$. tp denotes true positive, fp false positive, fn false negative.

We have achieved a precision at 96.3%. But since $F2$ score emphasizes more on recall, our final mean $F2$ scores of two methods are a little bit lower. Figure 5 shows some samples of our final prediction.

References

[1] anokas. Data exploration and analysis. https://www.kaggle.com/anokas/data-exploration-analysis. Accessed June 13, 2017. 2

[2] S. Brown. Measures of shape: Skewness and kurtosis. https://brownmath.com/stat/shape.htm/. Accessed June 13, 2017. 2, 3

[3] T. Chen. Introduction to boosted trees. http://homes.cs.washington.edu/~tqchen/pdf/BoostedTree.pdf. Accessed June 13, 2017. 3

[4] F. fei Li. Convolutional neural networks(cnns/convnets). http://cs231n.github.io/convolutional-networks/. Accessed June 13, 2017. 3

[5] Planet. Data of the kaggle competition - planet: Understanding the amazon from space. https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/data. Accessed June 13, 2017. 1

[6] Planet. Overview of the kaggle competition - planet: Understanding the amazon from space. https://www.kaggle.com/c/planet-understanding-the-amazon-from-space. Accessed June 13, 2017. 1

[7] Robin. Get started with the data. https://www.kaggle.com/robinkraft/getting-started-with-the-data-now-with-docs. Accessed June 13, 2017. 2