Environmental analysis of images

and their military applications

Author Adviser

Jiayi Lou Xin Gao

215595796 xingao@mathstat.yorku.ca

[Jiayilou318@gmail.com](mailto:Jiayilou318@gmail.com)

Department of Math & Stats

York Univercity

**Abstract**

This project uses open online resources as train data to analyze the environment in the images. The resource is made up of 800 photos and a data table. These pictures and dataset are available from follows net address: <http://www.ee.columbia.edu/trustfoto> , In this report, We describe how the AI analyzes the image and how to determine the environment in the image. In addition, We discussed the use and future development of this technology in the military.

**1.Introduction**

German spies and intelligence agents during world war ii, without today's drone technology, analyzed environmental conditions by surveying newspaper pictures of the target's location. After 70 years, Computer intelligence analysis replaces manual analysis. Analyzing the environment with images is an important part of the reconnaissance field. In particular, reconnaissance images from drones used in military operations can be used to analyse whether the local weather is suitable for combat. The analysis of the battlefield environment influenced the final result of the battle. Sometimes fighting is not in the wild, attacking indoor terrorists is a good example.

Before the computer makes a prediction, We need data sets to build the prediction model. In this report, I will use the RBG data of the pictures as the analysis object. Photo source is from Columbia University team. By logistic regression and neural network, I will show how the computer analyzes the picture data and gives the prediction.

**2.Dataset and pictures**

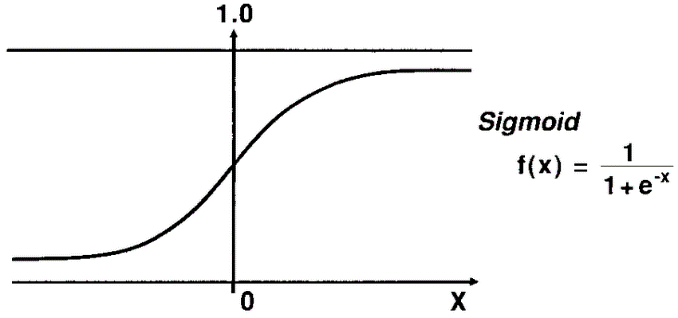
The picture package is from Columbia University team. The resource is available on: http://www.ee.columbia.edu/trustfoto. There were three photographers taking 800 pictures in two different cities (new york and boston). The cameramen used two different cameras-- nikon D70 and canon 10D. The 800 images were divided into eight categories which are artificial, indoor-dark, indoor-light, natural, outdoor-dawn-dusk, outdoor-day, outdoor-night, outdoor-rain-snow. Sunny day outdoor pictures are our main objects of recognition, so I'll use a dichotomy to separate out the labels with outdoor-day.

**3.Methodology**

In the processing and analysis of the picture, I use the following two methods to process the data.

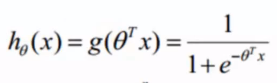
**3.1 logistic regression**

Logistic regression is one of the classical algorithms. Many computers prefer logistic regression because it is fast and efficient. We need to understand the Logistic regression before we can understand sigmoid function.

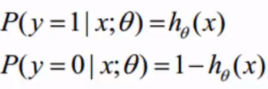
**3.1.1 sigmoid function**

Sigmoid function can map a real number to an interval of (0,1), which can be used for dichotomies. It works best when the feature differences are complex or not too large. It’s upper limit is 1 and it’s lower limit is 0, These two values mean the two options(yes or no). When we pick any number as input, sigmoid function outputs a probability. In general, this function does classification task.

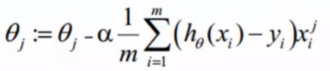
**3.1.2 details of** **logistic regression**

 Firstly, we show the prediction function [1]:

the input of the function is a linear regression form[1]:

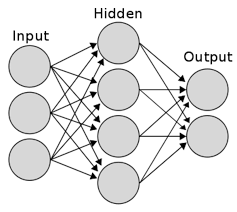
if we give some input we can get the result like this[1]:

If we combine the above results, we can get a simplified formula[1]:

Then we use log likelihood to convert the formula to take the derivativ. So after we do that, we can get a training formula for the theta[1]:

After many iterations of the computer, we finally came up with a better Θ. Finally we put ΘtX in to sigmoid function, we can get a predict value to do classification task.

**3.2 neural network**

The structure of the neural network is shown in the following picture, which is generally composed of three parts: the input layer, the hidden layer (one layer is given in the figure for convenience,[2] but in practice there may be multiple layers) and the output layer.For each layer, it is composed of several units (neurons).Neurons in two adjacent layers are fully connected, but there is no connection between neurons in the same layer, somethimes we call hidden layer black box.

**3.2.1 working process of** **neural network**

First, we give some input data into the input layer, by relying on linear combination caculation, we can get the value of each neuron in the adjacent layer of the input layer. Then we use the current layer’s value as the input to act on the next layer, one after another, until reaching the output layer. But if you just take the linear transformation, the resulting output is just a linear representation of the input. This kind output does not meet our requirements (we need to get a nonlinear expression of the input), so usually after each neuron is combined linearly, we need use activation function to transform and the result of the activation function is used as the value of the corresponding neuron.

In general, we use sigmoid function as activation function.

**4.Analysis and model**

We will use R as the modeling software. Because R is a professional data analysis software, it has a large number of analysis packages for us to use. We will use following analysis packages:

1. JPEG package

This package can read pictures and represent them numerically, so we can use these values as our input data.

1. NEURALNET

This package can simplify the computation code of neural network.

1. PORC

This package can give us the accuracy of a model.

**4.1 Logistic regression model**

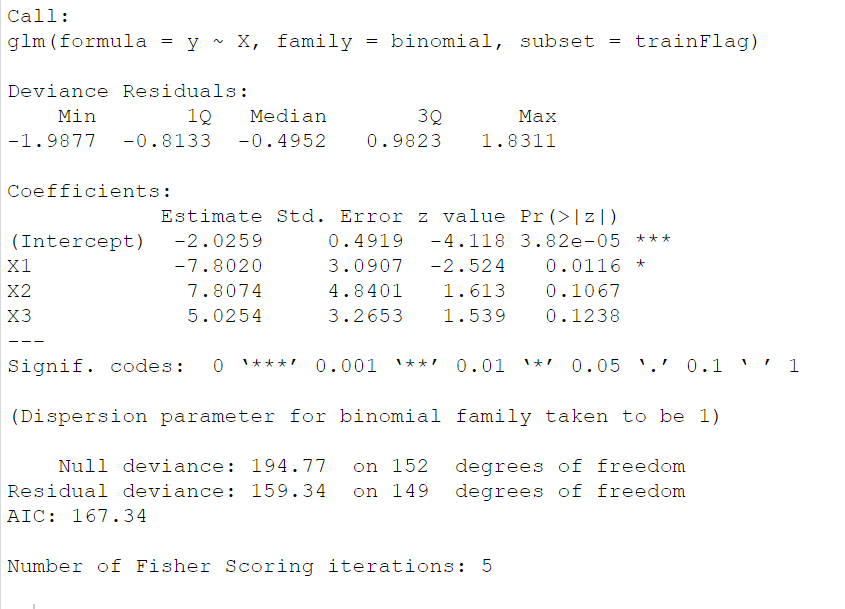
Suppose a military operation requires good weather. There are eight categories in the data. The outdoor-day in these eight categories meet the conditions for military action. So we mark all the data with outdoor-day label as 1, otherwise the mark left data as 0.

First, we get the data in 800 images by JPEG package’s function in R. The data we get from each picture is composed of three kinds values which is linked to image’s 3 channels (RED, GREEN, BLUE). Because of the size and pixels of each image, the amount of each channel data in each image is different. To unify the data, I take the median of each channel data in each image as input variable. In R, I use ‘readJPEG’ code to gain input variable.

Two sets of data are needed to build the model. They are training data and verification data. In general, we use 80 percent of the data as training data, the left 20 percent of data is verification data. In 800 pictures, we randomly selected 640 images without replacement as training data, the left 240 images is verification data. In R, to extract 80 percent of data evenly, I use ‘runif>0.8’ code as tool.

Each image that has been read has three pieces of data, there are red channel median value, green channel median value and blue channel median value. These three values are the basic data of a picture, so I use them as variables of the model.

From train data set, I take each picture’s category (outdoor-day or not outdoor-day) as y(1 or 0) and take 3 different channel data as x(variable). In R, we use code ‘glm’ to finish Core module building. Because of the dichotomies and Future verification , we chose the binomial as ‘glm’ code’s precondition and the subset should be our train data set.

By computer calculation, we can get following table:

|  |  |  |  |
| --- | --- | --- | --- |
| Θ0 | Θ1 | Θ2 | Θ3 |
| -2.0259 | -7.8020 | 7.8074 | 5.0254 |

The total model function is y=e(-2.0259-7.8020X1+7.8074X2+5.0254X3)

**4.1.1How to predict the result?**

Take the first picture as a example, the 3 channel’s median value is (0.42745098,0.400000000,0.376470588) If we put this 3 values into our model function, we can get that:

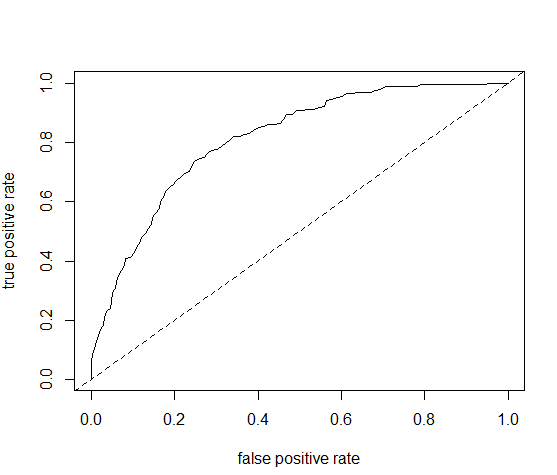
y = e^(-2.0259-7.8020\*0.42745098+7.8074\*0.4+5.0254\*0.376470588)

=0.543

Because 0.543>0.5, we say the result is close to 1, which means we predict that the first picture is outdoor-day picture. We can also say that the first picture have 54.3% probability to be outdoor-day picture.

**4.1.2** **Accuracy**

In R, we can use ‘PORC’ packages, by the code ‘ROC’, we can get a picture:

We can also write own code to achieve the same function to plot ROC picture

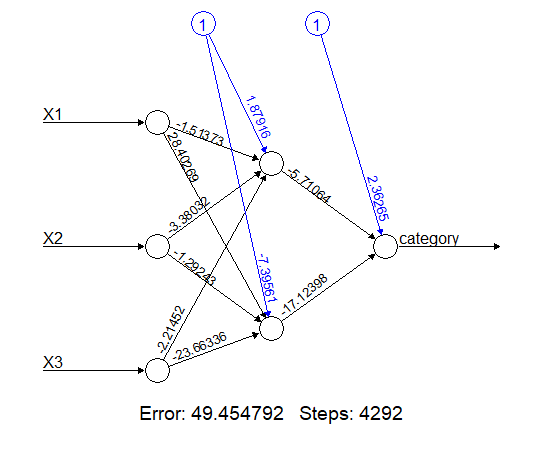
The top left curve tends to a right Angle, which means this model is good, we do not need to change any variables. The area of the lower half of the arc means it’s accuracy. We can calculate the accuracy by code, the accuracy of this mode is o.816.

**4.2 neural network**

To facilitate the comparison between the two models, we use the same data set which is 3 channels’ median value. Again, we use 80% of the data as training data. But this time, we use code ‘sample’ to randomly select 640 pictures.

Before we do the modeling, we always need to normalize the data set. But ‘For machine learning, every dataset does not require normalization. It is required only when features have different ranges.’[3]

In R, we use the code ‘neuralnet’ from NEURALNET package to build our model. Each picture’s 3 channels’ median values are (x)variable in our model. In general, there is no direct correlation between the three channels, so we set the model’s variables have no linear correlation and we assume that the effect of independent variables on dependent variables is nonlinear. In addition we set model’s threshold to be 0.01, which means that if the error change during the iteration is less than 1%, the model will not be further optimized. Finally, we set up two hidden layers in the black box of the model. We can get such code ‘network<- neuralnet(category~X1+X2+X3,train,hidden=c(2), linear.output=FALSE, threshold=0.01)’. ‘The question of how many hidden layers and how many hidden nodes should there be always comes up in any classification task of remotely sensed data using neural networks. Until today there has been no exact solution.’ [4] The reason why we choose two hidden layers is to reduce the computing burden on the computer.

After thousands of iterations we can get the following neuron details:

From the figure, we find that the computer has carried out 4292 iterations.

**4.2.1 how to use this model**

We take the first picture as example, the 3 channel’s median value is (0.42745098,0.400000000,0.376470588) If we put these 3 values into our model function, we can get following calculation:

**The first hidden layer:**

s1=0.427\*(-1.514)+0.4\*(-3.380)+0.376\*(-2.215)+1.880=-0.951

We convert the raw hidden layer values into activation functions,

h1=1/(1+e-0.951)=0.721

**the second hidden layer：**

s2=0.427\*(28.402)+0.4\*(-1.292)+0.376\*(-23.663)-7.396=-4.862

h2=1/(1+e-4.862)=0.991

**the final output:**

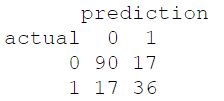
o=0.721\*(-5.711)+0.991\*(-17.124)+2.363=-18.725

c=1/(1+e-18.725)=0.999

By calculation, we figured out that the first picture havs 99% probability to be outdoor-day picture. So we can predict that the first picture is a outdoor-day picture.

**4.2.2 Accuracy**

In order to test our training model, we need to use the validation data to determine the accuracy. We have 160 pictures as validation data set. By computer calculation, 为can get following table:



We found that the accuracy of predicting non-outdoor-day photos is 84%(90/107), but the accuracy of predicting outdoor-day photos is very low(68%). The total correct rate is 78%((90+36)/160).

**5. An Overview of two model’s result**

By comparing the overall accuracy, we find that logistic regression model(82%) is better than neural network model(78%). In detail, by comparing the ability for predicting non-outdoor-day photos, Neural networks are more efficient than logistic regression. However, Our goal is to identify outdoor-day photos (not non-outdoor-day), The neural network model has only 68 percent accurate in this regard which is not good model. Therefore, we are sure that logistic regression model has greater advantages.

**6.Conclusion**

For logistic regression model, the specificity and sensitivity of the model are directly linked to the number of available variables [5]. In this report, we find that logistic regression is good at two-category data. Neural networks do not seem to have the edge in this field. But if we ask the model to judge multiple categories (Not a dichotomy), I believe the neural network is the best choice.

**7.Reference**

[1] XIN GAO (2020), formula of logistic regression power point, Retrieved April 10, 2020, from https://xingao.info.yorku.ca/?p=1398.

[2] Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. IEEE transactions on pattern analysis and machine intelligence, 12(10), 993-1001.

[3] Lakshmanan, S. L. S. (2019, May 16). How, When and Why Should You Normalize / Standardize / Rescale Your Data? Retrieved April 10, 2020, from https://medium.com/@swethalakshmanan14/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff

[4] D. Stathakis (2009) How many hidden layers and nodes?, International Journal of Remote Sensing, page1, 30:8, 2133-2147, DOI: 10.1080/01431160802549278

[5] Coughlin, Steven & Trock, Bruce & Criqui, Michael & Pickle, Linda & Browner, Deirdre & Tefft, Mariella. (1992). The logistic modeling of sensitivity, specificity, and predictive value of a diagnostic test. Journal of clinical epidemiology. 45. 1-7. 10.1016/0895-4356(92)90180-U.

**8.Professor's Logistic regression code**



**9.my network code**

