Classification of Birds Images

by Chris Nelson

Introduction

For my project I chose to work with a dataset of images with included numerical features. The dataset consists of 11,788 images of different species of birds, each of which falls under 1 of 200 species classifications. The included features for each image are a set of 312 binary attributes that specify whether or not a particular feature is present in that image. My goal was to correctly identify each bird in a photo given the features it had and didn't have.

Motivation

The main motivation behind deciding to do this project was in large part due to my senior design project. For my senior design project I am working with my team to create a system to process video of basil leaves and pinpoint the number and position of the leaves. To be able to do this, I wanted to get some solid experience in dealing with image and image feature data to get a better sense of how these things work. Following this project, I feel like I have a much better idea of how to deal with image data and how to manipulate features to create shortcuts and advantages for machine learning systems.

In addition to the Senior Design motivation of this project, I additionally found out some interesting things about image classification of animals while doing some research. I found that using machine learning and AI models is an area of interest and growth in the biology community. This is due to the fact that there have been many projects documenting and recording animal behavior using things life underwater microphones, wildlife cameras, and other sensors placed in and around habitats. With these recordings, comes a very large amount of data, that as raw data can be overwhelming for humans to deal with and classify. Luckily, scientists have actually been turning to AI/machine learning models in order to automate the classification process. So in regards to my project and the dataset, finding ways to tag images with the species that are in them would relieve researchers of hours of tedious work spent simply classifying the data they collect. Using machine learning for biological research purposes is not something I would have ever thought of, but it is fascinating to see how ML can have such wide reaching beneficial use cases.

Methodology

When starting this project and looking at my chosen dataset, I was given a variety of data in numerous files. In order to start making some sense of all of it, my first task was figuring out what data out of the dataset I actually wanted to use, as different pieces of information would allow me to do different things and pose different challenges.

In the end I decided to try to use just the features present in each of the images along with a confidence factor in order to train my model. At first I was planning to actually import all of the photos, but that ended up being more data than my computer would handle, so I decided to stick with just the features. As for the features, there are 312 of them, each having to do with whether or not an attribute, like a yellow beak for example, is present on the bird in the image.

After importing all the data and arranging it into understandable numpy arrays, I then separated the values into testing and training data. The dataset actually came with a file that specifies which images it recommends for testing and training so for simplicity sake I decided to just follow the recommendation and split the arrays up as specified.

At this point, I ran a few machine learning models on the data, those being LDA and QDA to start as those were the first two classification algorithms we learned in class. Neither of these performed particularly well, both below 50% in testing. As a result, I figured I should try and use some of the other data I had to preprocess the target data a little bit. I used provided data that specified how common certain features were in each class to scale the binary feature data from the images. Using this data, features that were more prevalent for certain classes, were more weighted than those that were less common. This allowed for a fairly significant jump in performance of all models as can be seen in the following section.

Testing Results

Baseline

For my baseline, I did my best to write a few rules in which I thought might provide some level of predictive capability. After testing a few different configurations, the best I was able to obtain was about a 6% accuracy score. This is obviously not great, although I thought it served as a better baseline than just random guessing. With 200 possible categories, a random guess would have a theoretical accuracy of around 0.5%. In regards to random guessing, my predictive ability using basic rules does not look as bad, but my version was very slow and more advanced solutions can obviously do much much better.

Rule Based Accuracy: 6%

LDA

All told, for being the first method that I ran, LDA did not do too poorly. The training accuracy was around 80% with testing around 70%. While 70% is not in the 97% and up that I have come to expect from lab results, it was a pretty good starting point due to the fact that the feature data is only binary with some scaling, meaning there is not a lot of information for the model to work with. This was even noted by the sklearn LDA implementation, as it warned that some of the variables were collinear, which again was bound to happen with the attributes being binary originally.

Training Accuracy: 80.08%
Testing Accuracy: 70.26%

QDA

Seeing as LDA did fairly well, I expected QDA would result in somewhat similar performance. This was not the case at all. Clearly for this dataset, in which the data was not very numerically varied, the model constructed was completely overfit, as the training accuracy came out to 1.0 or 100%. As a result I knew testing accuracy would not be good, and it was not. It was an abysmal 1.2% or 5 times worse than my rule based model.

Training Accuracy: 100%
Testing Accuracy: 1.2%

Logistic Regression

Logistic Regression provided one of the best results of any of the models with a 78% accuracy on the testing set. Despite this good performance, it is still roughly 12% less than the training accuracy of about 90%. Knowing how Logistic regression works to some extent, I expected this to be one of the better models, as it goes beyond simply linear but does not suffer the same overfitting problems that QDA can face.

Training Accuracy: 90.02%
Testing Accuracy: 78.24%

Decision Tree

The Decision Tree model was the first model I used that was outside of our in class learning material, but after doing some research about using binary features, comments suggested a decision tree would most likely do the best job. This turned out to be accurate for me, as it performed the best out of all the models I chose to test. What was interesting to me was that despite the training accuracy being 100%, the model was not overfit and performed very well on the testing set. Obviously Decision Tree models have a different underlying behavior that does not cause overfitting to happen, at least not in this case.

Training Accuracy: 100%
Testing Accuracy: 92.16%

Support Vector Machine

SVM was the last machine learning model I applied to this data. Again, I do not know the underlying principle of how SVM's work, but one thing I noticed about the results are the testing and training results were the closest of any of the models. This means that the model fits both the training and testing data roughly the same level, which in both cases is not astonishingly well, but still better than 60% for both training and testing.

Training Accuracy: 66.33%
Testing Accuracy: 60.51%

Summary, Comparison, and Conclusion

As a wrap up to my testing, it is important to highlight which model performed the best and which did the worst. The Decision Tree model was clearly the best performing model at ~92% accuracy. Logistic Regression was then a fairly distant second at 78%. The worst by far was Quadratic Discriminant Analysis (QDA) at only 1% testing accuracy. Working on this project and seeing the vast difference in results between models reaffirmed what we learned in class: that there is no one canonical machine learning method and that each do well in particular scenarios. Following this project I would like to look more into decision trees to understand what

made that model particularly effective in this scenario. Overall I am happy with the results I obtained and hope that I can apply what I have learned going forward in my senior design project the rest of the year.

References

Dataset:

Wah C., Branson S., Welinder P., Perona P., Belongie S. “The Caltech-UCSD Birds-200-2011 Dataset.” Computation & Neural Systems Technical Report, CNS-TR-2011-001.
http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

Other Sources:

https://www.zooniverse.org/projects/zooniverse/snapshot-serengeti

https://www.whaleshark.org

https://www.youtube.com/watch?v=tSoqJpisKIg