Project: Avian Asker

Supervising TAs: Liang Wang, Carlos Gonzalez
Base on handout by Noah Jakimo

October 17, 2010

1 Introduction: Guessing Games/Watching Birds

The *Oxford English Dictionary* contains ~500,000 words. Its size is only a little less than 2^{19}, and a binary search can therefore solve the 20 Questions game rather easily. This is why we move to a more interesting and feathery problem.

2 System Requirements: Python 2.6+ (Any Platform)

Avian Asker requires the Python programming language. For tutorial, the respective CS 11 track's website[1] can bring any programmer up to speed.

Feel free to work within the operating system of your choice. Unless that is Windows, you should already have an appropriate version of Python pre-installed. Nonetheless, everyone should check they have a Python version which is 2.6 or higher[2], as well as software that can unpackage the data to get started.

3 Data & Extraction: Caltech-UCSD Birds 200[3]

Images[4]

There are 6,033 images from 200 bird species - listed in classes.txt - within our collection. Kindly ignore the half that test.txt enumerates. Please only work with those in the train.txt file.

Note, our ornithology of JPEG files are catalogued into subdirectories according to species. As you begin bird watching, the script below will display one fledgling at random:

```
import random, Image
infile = open("lists/train.txt","r")
train = infile.readlines()
rndbrd = train[random.randint(1,len(train))]
Image.open('images/'+rndbrd.rstrip()).show()
```

[1] http://www.cs.caltech.edu/courses/cs11/material/python/
[2] http://www.python.org/download/
[3] Welinder P., Branson S., Mita T., Wah C., Schroff F., Belongie S., Perona, P. "Caltech-UCSD Birds 200". California Institute of Technology. CNS-TR-2010-001. 2010.
[4] http://www.vision.caltech.edu/visipedia-data/CUB-200/images.tgz

Attribute Annotations[5]

Each time an image was viewed, 1 of 1,577 MTurk participants provided 5 of 228 attribute responses. All possible attributes are listed in attributes.txt and all responses have been logged in labels.txt (Please refer to README.txt for their format). Labels may be collected into a Python list as follows:

```
labels = []
infile = open("labels.txt","r")
for line in infile.readlines():
   entry = map(int, line.split())
   labels.append(entry)
```

The file is large (almost 9 million lines), and constantly searching through it will surely waste your time. Instead, consider building Python dictionaries while you read in, and saving those with the `cpickle` Python module.

4 The Interface: You ask, we answer.

You will write a method, `myAvianAsker`, that accepts a list of answers and returns the index of either an attribute in question or bird species. Please define your method inside a class that has the name of your group, *i.e* A1, or A12.

Species are numbered from 1 to 200 as specified in the file `specie_name.txt`, and attributes are numbered 1 to 283 in `attributes.txt`. Your function should return 1 to 200 if it is guessing the number of the species or 201 to 483, if it is asking about attribute 1 to 283 respectively. Your function can guess incorrectly, and our program will simply keep on asking for more questions.

Our function returns 0 - "no", 1-"yes" or 2 -"uncertain". For the milestone, no "uncertain" answers will be given. Here is how the script looks like in general. For the exact file, please download `gameplay.py` in `milestion.zip`

```
QAs = []
while True:
        Q = myAvianAsker(QAs)
        if Q == rndbrd:
                print("You have guessed correctly, the bird is "...\n")
                break
        elif Q <= nspecies and Q != rndbrd:
                A = "0" #incorrect guess
        else:
                A = #answer to your question in the format 0,1,2
QAs.append([Q, A])
```

5 Milestone: Perfect Accuracy on a Simple Set

Please download the archive `milestion.zip` where you will find the following files:

- `dataset.txt`: This corresponds to training data for the 200 species and 283 attributes. The file has in each line the picture id number, the attribute number, and the answer in binary format. There is exactly one answer for each attribute, for each species.
- `gameplay.py`: This python file runs the Q & A game described before. A sample `myAvianAsker` function is provided which asks questions chosen at random and makes random guesses, so that you can see how your function should work. (Not a clever function like yours indeed!).

[5] http://www.vision.caltech.edu/visipedia-data/CUB-200/attributes.tgz

- `specie_names.txt`: This file relates the picture id with the species number, its name and the picure name (in this order). Clearly, since the random bird we provide is identified with the species number, you cannot use this information to hard code which bird this corresponds to. However, we use this file in our code to provide answers to the questions generated by your function so you should keep it in the directory. Use this file to relate the picture id with the species id.
- `attributes.txt`: This file describes each of the 283 attributes you can ask questions to.

6 Goal: Noisy Classification on a Withheld Test

Ultimately, we will return our attention to the data in labels.txt. Your Avian Asker will then have to handle conflicting attribute responses and use the fewest questions to identify a bird's species.

7 Closing Comments:

Computer Vision Possibilities

Further along, we plan to pass the bird's image into your asker. You can try to reduce your line of questions through processing its plumage.

Two-Parameter Objective Function

Grades will be decided by some function of accuracy and efficiency.