Mini-Project 4: Monster Identification CS7637

Josh Adams
jadams334@gatech.edu

Abstract—For this project we are tasked with developing a relatively intelligent agent. This agent will be given a set of positive and negative examples of monsters. Each of the samples have 12 different features. The agent is to learn as much as it can from the subset and then predict if a new instance is an example of a monster or not.

1 HOW THE AGENT WORKS

The most successful method I used was a suggestion by someone in Ed discussions. This suggestion was to only use the sub-sample provided and if the given monster has traits which are not found in the positive examples to return False. This was by far my most successful iteration of the agent as it was able to score above 36 on almost each submission.

My first approach was based on calculating percentages of the traits found within positive and negative examples. This was not working very well so I augmented the results by implementing linear regression and calculating the distance the point would appear in both negative examples and positive examples. This also was not successful.

An alternative method I tried was aggregating all previous examples to provide a much larger knowledge base. I thought this would make my agent more robust. Figure 1 shows the traits and their appearance in positive and negative samples. Each of the non-numerical traits were encoded with a numerical value. For example, looking at the 'size' data, and encoding of 0 actually corresponds to 'huge'. While other such as arm-count which was a numerical value, just corresponds to that value. There were only two traits which were exclusive to just one type of sample. This was 'arm-count' of 0 was only found in negative instances and 'color' with a value of five, which corresponds to the color 'purple'.

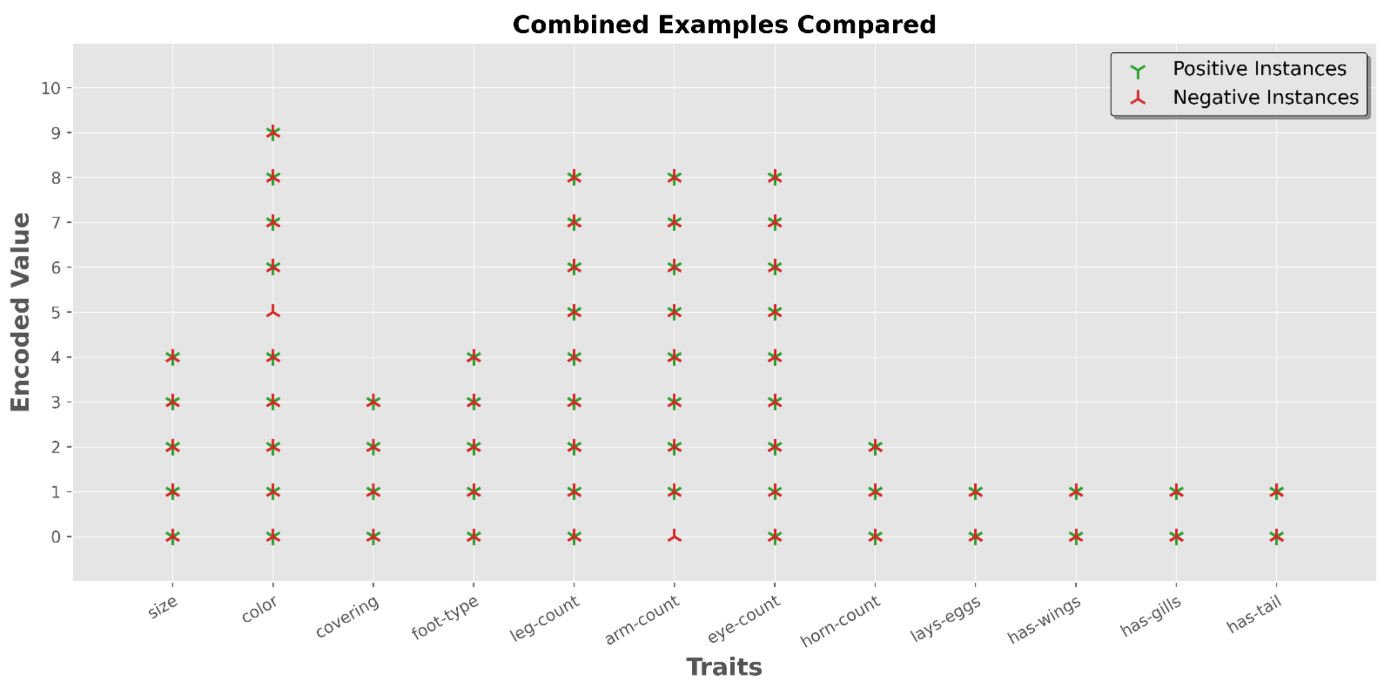

Figure 1 — Example of expanding trait frequencies.

It turns out I was wrong. The reason this was not working was that as my knowledge base expanded, there were less differences to take advantage of to determine if a sample is a monster or not. Figure 1 shows in red the negative examples and green the positive examples. Practically every single value for a trait is shared between both positive and negative examples as the knowledge base increases.

2 AGENT PERFORMANCE

My agent is able to get between 36 and 40 points on Gradescope. But it seems to be dependent on the questions that are generated for the agent. It does not struggle on any particular case because it will just produce an answer no matter what. In terms of struggling with accuracy, the agent struggles when it comes in contact with previously seem samples. The reason it struggles is that it must process that new sample and try to reason through and changes it must make based on that new sample, as well as recalculating all previously calculated heuristics.

3 AGENT EFFICIENCY

Once I remove all of the extra functionality from my agent its efficiency increased significantly. My agent runs O(N) as best, where N is the number of samples given to the agent. The reason it runs this quickly is that the agent only iterates a single time over the samples. During this iteration, the agent will verify its features belong in the positive samples, if it does, then it returns True, else it returns False.

4 AGENT CLEVERNESS

I was not able to solve this mini project using complicated methods. I ended up using a suggestion from someone in Ed discussion. The cleverness boils down to only returning False when an attribute found in the monster is not in the set of positive instance features. I tried many other approaches, and all failed to produce as many correct as this approach.

5 AGENT COMPARED TO HUMANS

With respect to speed my agent is able to solve the problems much faster that most humans. If the data for the monsters was provided as a picture, I would expect a human's response to be faster than the previous human speed. I still would expect my agent to produce an answer much faster than a typical human. A big difference would be the accuracy of my agent versus a human. Currently my agent is able to solve 40/40 of the problems but I ran some tests. It appears that this is only the case when given small subsets of the distributed monsters. I modified my agent to aggregate previous given datasets, so its knowledge base is expanded with each successive iteration. This dramatically reduce my agent's overall performance. The reason is the way my agent works, and the knowledge increases the cleverness benefit begins experiencing significant diminishing returns in terms of accuracy. As for similarity between the way a human would solve the problem and my agent, I would say they are limited in their similarity. The methods my agent takes would only be a small step of the human process to solve.

6 REFERENCES

1. Numpy