Matthew Daniel

CS 415

March 30, 2017

Project 2 Write-Up

**Genetic Algorithm Description:**

I wrote a genetic algorithm in Python that creates a population of size 50 comprised of strings of length 50 whose characters are selected randomly from the amino dictionary in scoringMatrix.py from lecture. Each creature is assigned a fitness score based upon how many English words of 2 or more characters it contains. To get the number of English words, I search through the string character-by-character, checking to see if the character forms an English word of length 2 to the end of the string using a package called “Enchant.” This process takes O(n2) time, so the algorithm runtime is around 70 seconds. I used a mutation rate of 20% to attempt to escape local maxima. This mutation rate seemed to be successful during assignment 1, so it makes sense to use it here. Crossover occurs in the same way as the in-class algorithm, swapping two characters between two strings with a probability of 0.5.

**Data & Analysis Description**

Each iteration, the most fit creature is written to a file. These creatures are strings of length 50, and there will be 1000 of them, one from each iteration. One file contains the creatures only, while another contains the iteration number as well for comparison later. A third file contains the first and final most-fit creatures that the algorithm produces. A separate program, analyze.py, reads in the creatures from the unlabeled file and reorders them. I will apply a global alignment algorithm to the unlabeled creatures and compare them with creature from the original population. I will attempt to use the Blossom-50 scoring matrix to assess the alignments. Since my fitness function has nothing to do with biological processes, this may be unsuccessful, and I may have to engineer another scoring matrix. I will assume that the lower the alignment score a creature gets, the later it was produced (the higher its relative age). I will sort the creatures by alignment score descending, then compare my array to the file that contains the original iteration information. I will measure the effectiveness of the alignment by the average distance a creature is calculated from where it ought to be by comparison with the unaltered data.

**Potential Issues**

I made sure that my GA codes is easily modifiable to alter the number of iterations, number of creatures outputted to the files, etc. so that I can make changes should my analysis prove unsuccessful. One change that I might make entails modifying my mutation function to favor amino acids with higher comparative scores when mutating a character. I suspect that this may increase the effectiveness of Blossom-50 if I encounter problems with the alignment scores yielding useful results.

**Initial Alignment Testing:**

Upon initial testing, I found that by applying a slightly modified version of the global alignment algorithm from the class, the resulting alignment scores from Blossom-50 were nearly identical across all 1000 samples. Before attempting to match up the reordered results with the original data to see how my algorithm had done, I decided to alter the mutation function in the GA as I had planned earlier.

**Updates to the Genetic Algorithm**

I altered the mutation function in the GA to favor mutations that yield higher Blossom-50 scores. I accomplished this by selecting a random number between 0 and 10, and mutating to the first amino acid whose score exceeded this random value. While this method was not grounded in anything other than intuition that it might better utilize the Blossom-50 matrix better than random amino selection, it did yield more varied alignment scores.

**Measurement of Alignment Algorithm Success**

I wrote code to track the effectiveness of my alignment algorithm’s ability to reorder the sequences by generation from newest to oldest. The oldest organisms would potentially have the highest alignment scores when compared to the original organism. I compared my array of strings ordered by alignment score to the original array of strings ordered by generation. As the measure of effectiveness, I took the minimum of the absolute values of the difference between indices of equal strings. This meant that I was calculating the minimum distance away a creature was from its actual generation. I then took the mean of the sum of these distances as a measure of how far away the average creature was based on alignment algorithm from where it should have been.

**Results**

Over 20 trials, I found that the average distance from a creature to its proper generation was 252.96, and the standard deviation was 86.66. If the creatures were just randomly reordered, we would expect the average deviation to be 500, half of the number of generations. The algorithm does perform better than random, but does not get close to a perfect reordering.

**Conclusion and Suspected Issues**

I suspect that the two main areas that caused the algorithm to perform poorly were the use of the Blossom-50 matrix, and the similarity between alignment scores. My crossover and mutation functions were not intended to mirror biological processes. Therefore, I was not using Blossom-50 with sequences that it was intended to be used for. This certainly had the potential to affect alignment scores and make them a poor measure for how similar two different organisms are. Reordering based on alignment score would not make the generations reorder correctly in this case. Additionally, even though I altered my mutation function, alignment scores were still relatively similar. For example, if 50 creatures have the same alignment score, my algorithm comparing their relative positions could have been up to 50 units of distances off. This flaw certainly made my alignment process less effective.

I will tentatively conclude that my algorithm performs about halfway between randomly recording the creatures and an algorithm that would reorder them perfectly. I would have to make some major alterations to my alignment scoring method to achieve an algorithm that reorders the organisms well. The experiment does not work well enough to have this be a viable method for figuring out an organism’s generation based on its alignment score.