COMP-8920 - Summer 2019

Genetic Algorithm & Keystroke Dynamics

MINI PROJECT 01

DONE BY:

1. SAMEER AKHTAR SYED (Student ID: 105152677)

2. MUHAMMAD ANWAR SHAHID (Student ID: 104711240)

Keystroke Detector and Classification of *Sheeps, lambs, goats and wolves* :

Output generated:-

- For each subject, we print his ID, his success rate on his signature and then whether he is classified as Sheep or Goat.

- Then We're printing FRR which is simply his success rate on his signature subtracted from 1
- Then we print FAR for each subject. (Since there are too many samples, we randomly select 1000 samples not belonging to the subject and get the success rate on them. This will be FAR for that subject.)

- Then for each subject, we find out what signatures he can imitate/what subjects can imitate his signature. We're printing the statistics related to them for each user. We only used 100 samples to check if a subject can imitate a signature successfully due to computational constraints.

- After that we print all the observations regarding each subject, whether he is a wolf/lamb or not. A subject can be both a wolf and a lamb, as well as neither of them. (Most subjects are neither.)

- Then we print data related to each generation of GA.

Observations:-

- There was a lot of variance in the data within a user. Same user might take 8 seconds or 16 seconds to type the same password. To identify patterns within such vastly diverse data, We scaled all of them to 1 seconds in total by dividing them by the total time.

- For signature, We're storing the mean of 10 attempts. If x = Euclidean distance from match, we take match % = 1/(1+2x) . This way when x=0, we get match = 100% for a perfect match and when x is very large, match = 0% for no match. We tried x = Manhattan distance, and match = e^(-x) as well, but the results are more consistent and meaningful with the choices explained above. (eg e^(-x) caused underflow in some cases.)

- In an ideal situation, we will expect all subjects to be sheep. However, 18/51 subjects are goats. This means that 18 users failed to login to their account more than 30% of the time.

- We considered a subject 1 can imitate subject 2 if 70% of his attempts to login as subject 2 are successful. A subject is classified as a wolf when he can imitate at least 5 other subjects, and a lamb when at least 5 other subjects can imitate him. Thus these categories are neither exclusive nor complement to each other.

- There are 14 lambs and 14 wolves. That means there are 14 users which are vulnerable and 14 users with potential access to other's accounts. Also there are 4 user (5,17,29,30) which are lambs as well as wolves.

Genetic Algorithm:-

Fitness score:

- We took match % as fitness score directly.

Chromosomes:

- The durations are directly taken as chromosomes.

Initialization:

- We randomly generated 10 samples for each user(total 520) and saved them in random_num1.csv. We considered them as the initial population.

Selection:

- At each generation, 50% population with max fitness survives and rest are eliminated. We tried random sampling with weights proportional to fitness but we could not get convergence using that. Population is restored by adding children one at a time, by randomly selecting it's two parents from the survivors.

Crossover:

- Like in biological evolution, child should inherit properties from both mother and father. To implement this, We took random values for the child's chromosomes in the range between his parents' chromosome values.

Mutation:

- With probability 0.25, a child may also get chromosome values for a single random chromosome outside the range of its parents' values. We took a random chromosome, took an interval thrice the size of its parents and picked the chromosome's value uniformly from it. This allows the population to have some diversity even after many generations.

Termination:

- When a large portion of population (70%) can successfully attack the target subject (s002 in this case), we stop the GA.

Observation:

- Convergence in 10 generations. Since we have a relatively moderate target (85%) for a successful match and initial population averaged at 74% match already, it didn't take many generations to achieve the target success rate. Interestingly, once a few points got a successful match for the first time, the population only took 6 generations to reach a success rate of 70%. This is also probably due to the moderate target for successful match.