Revised Algorithm Design Document

4our People

Kaleb Tesfay - 101048170
Adam Farah - 100966918
Japinder Sandhu - 101021899
Samson Teklay - 100972872

04/9/19

Introduction

Animal shelters consist of homeless animals waiting to be adopted into homes where they can be treated with love and care. The shelter staff provide care for these animals until a client is chosen to adopt them. Due to animal-client incompatibility issues, animals are not just given to random clients for adoption. Instead, the Carleton University Animal Care System (cuACS) runs an animal-client matching algorithm based on each animal's attribute as well as client's preferences to determine the most compatible match. Thus, it is only after running the ACM algorithm and finding the most compatible match is a client allowed to adopt a pet.

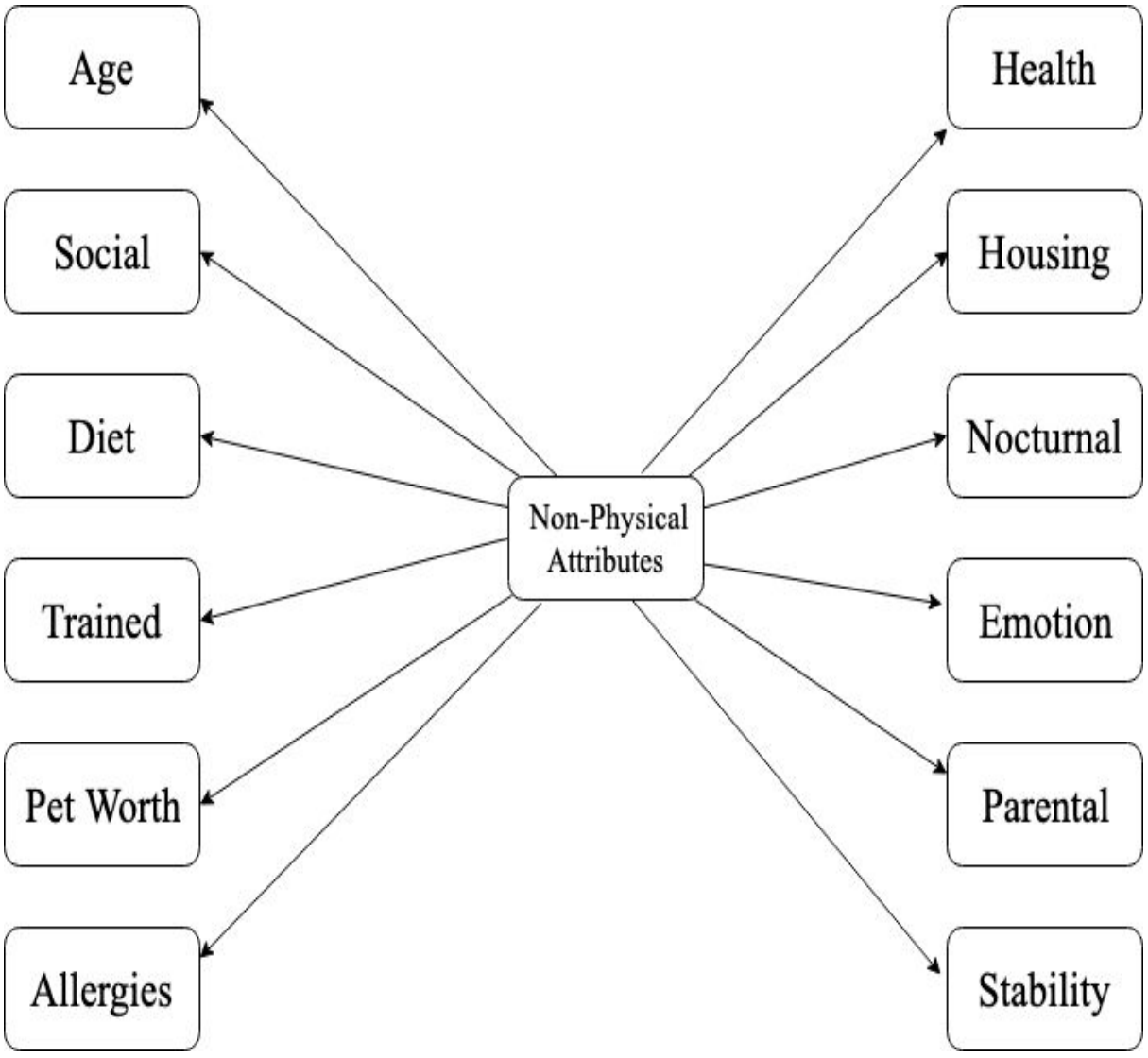
In deliverable #2, our team implementing the cuACS defined twelve (12) fixed non-physical attributes and qualities that are critical in coordinating compatible animal-client matches, possibly resulting in an adoption. Based on the non-physical attributes, we created attributes scales, algorithm rules and algorithm justification to come up with an animal-client matching algorithm that would help us determine compatibility. That being said, in our final design document, we will be revising our ACM algorithm design document from deliverable #2 to reflect all the non-physical attributes as well as algorithm rules used in our implementation of this final deliverable. We will be explaining how our algorithm narrows down the list of available pets for adoption to a few potential pets. Then, we will be discussing, in details, the scoring and the weighting system used to find the perfect one to one unique match between a pet and a client.

ACM Algorithm

Non Physical Attributes

Each animal's profile contains detailed information of the animal that is ready for adoption; including 12 or more physical and non-physical attributes.

Figure 1 - Non-physical Attributes



Algorithm Rules

The shelter staff will be in charge of running the ACM algorithm. Animal's non-physical attributes as well as client's pet preferences are stored in the shelter's database. To run the animal-client matching algorithm, the shelter staff uses an algorithm to find a 1 to 1 compatibility between the client and their matched pet. The system uses a filter based method that allows the algorithm to narrow down the list of animals available for adoption based on the client's preferences. Client's have their value preferences of their ideal pet saved in the system. Using their preferences values and the animals' non-physical attributes' scores, the shelter staff is able to run an animal-client matching algorithm. The ACM should use the filter based process until it is able to generate one-to-one matches between the client and a pet. However, if there is more than one pet matched with a client, then the pets' attribute values are compared with each other. Then, the pet which has values that are the closest to the client's preferences is matched with the client.

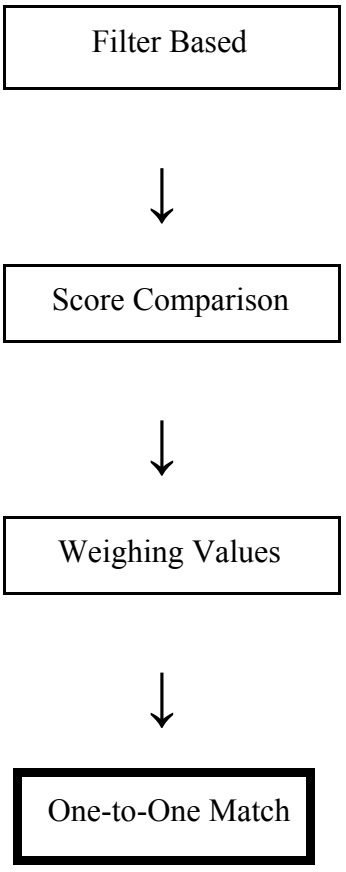


Table 2 - Scaled attributes levels

Age: Age of a pet

1	2	3	4	5

Social: A degree to which a pet associates itself with other pets

1	2	3	4	5

Diet: Diet restrictions levels of each pet

1	2	3	4	5

Trained: A pet could be not trained at all, little trained or well trained

1	2	3	4	5

Pet Worth: Different levels of pet maintenance cost

1	2	3	4	5

Allergy: Levels of allergies that a pet acquires

1	2	3	4	5

Health: Different health levels of a pet

1	2	3	4	5

Housing: Some pets require more space than other pets

1	2	3	4	5

Nocturnal: Some pets are more active at night while other pets are active during the day

1	2	3	4	5

Emotion: Each pet has different emotional levels

1	2	3	4	5

Parental: Some pets in the shelter have family members that they are born or grew up with

1	2	3	4	5

Stability: Independence of a pet

1	2	3	4	5

Algorithm Justification

Scoring Algorithm

The Scoring Algorithm will be a trivial process that occurs behind the scenes. Score refers to the added up and tallied integer points assigned to each attribute. Points assigned to each attribute will come from each client stored in the database and from each animal stored in the database. Each client and each animal will have their own scores.

All 12 of our attributes have a score from 1-5. The scores for the the client and animal are already stored in the database. The database holds scores for each attribute, using SQL we can access these individual attributes.

Line 1686 in mainwindow.cpp

```
client_list -> prepare(“ SELECT * FROM CLIENT WHERE id =’”+val+”’);
```

This command will create a query of all the clients that have an id, which is all the clients. This is used to iterate through all the clients and evaluate the best animal for each client in a one to one match. Unique matches will be created because of the filtering process which will remove already matched animals from the matching process.

While iterating through every client, we use filters to disregard animals which are not in the range of the attributes values requested. Once the filters are completed, the animals are compared using the scoring algorithm. The scoring algorithm adds up the values of the specific animal and separately adds all the attribute values of the client, then compares to see what animal will be the best match for the client in a one to one format.

Attribute Weighting

All of the attributes are scaled from 1-5 but may have different weights. The weight of an attribute is also referred to the “cost” of an attribute. In applications weight may be used to measure the importance of an attribute. Given weighted attributes, we are able to find a more fine tuned result for the animals resulting in higher animal welfare.

Some attributes are weighted higher than others. The attributes and the attribute weights can be seen in the table below. All of the percentages add up too 100%. Certain attributes are

weighted less for example, diet is not weighted largely because the diet of the animal should not be considered too much when choosing an animal. Animals regularly have standard diets. The weight of the attribute Age is weighted regularly as it is a standard condition that is used in all animals. Similarly, Social, PetWorth, Nocturnal and Emotion and standard weights. 5% is considered a standard weight. Allergy weight is slightly larger weight as an allergy is more important. Health attribute weight is weighted the largest as it is very important to know the health condition of the animal, as is Trained. Housing and Stability are also highly weighted as they are important to the matching.

Age	5%
Social	5%
Diet	2%
Trained	15%
PetWorth	5%
Allergy	8%
Health	15%
Housing	12.5%
Nocturnal	5%
Emotion	5%
Parental	10%
Stability	12.5%

Unique Matches

The algorithm will produce unique matches. The matching is produced by evaluating the score and the weightings. By going through a filter process, the animals for each client will reduce until there is only one or two animals left. If there are two animals left only then the score is compared because by that point the filters would have removed any animals which are not applicable. Once the score is compared and the appropriate animal is chosen, the animal will be popped off the stack. This will ensure that the next client being matched will not have a repeated match and the match will then be one to one or unique. Each client will be compared with the animals that remain in the filtered list.

Below is an example of the unique matching with a small db provided example.

Attributes	Client	Animal1	Animal2	Animal3	Animal4	Animal5	Animal6	Animal7
Age	1	1	1	1	1	1	1	1
Social	1	2	5	2	3	4	2	1
Diet	2	3	2	2	4	5	5	2
Trained	3	4	3	3	5	1	3	3
PetWorth	4	5	4	4	5	2	3	4
Allergy	5	4	5	2	1	1	3	5
Health	1	2	1	1	5	2	1	1
Housing	2	3	2	2	5	5	2	2
Nocturnal	3	2	3	3	5	5	3	3
Emotion	4	3	4	4	5	1	3	4
Parental	5	4	5	5	5	5	5-	5
Stability	1	2	1	1	5	4	3	1

Pass1: Age (+/- 1) 1
Potential_choice = [Animal1,Animal2,Animal3,Animal4,Animal5,Animal6,Animal7]

Pass2: Social (+/- 1) 1
Potential_choice = [Animal1,Animal3,Animal6,Animal7]

Pass3: Diet (+/- 1) 2
Potential_choice = [Animal1,Animal3,Animal7]

Pass4: Trained (+/- 1) 3
Potential_choice = [Animal1,Animal3,Animal7]

Pass5: PetWorth (+/- 1) 4
Potential_choice = [Animal1,Animal3,Animal7]

Pass6: Allergy (+/- 1) 5
Potential_choice = [Animal1,Animal7]

Pass7: Health (+/- 1) 1
Potential_choice = [Animal1,Animal7]

Pass8: Housing (+/- 1) 2
Potential_choice = [Animal1,Animal7]

Pass9: Nocturnal (+/- 1) 3
Potential_choice = [Animal1,Animal7]

Pass10: Emotion (+/- 1) 4
Potential_choice = [Animal1,Animal7]

Pass11: Parental (+/- 1) 5
Potential_choice = [Animal1,Animal7]

Pass12: Stability (+/- 1) 1
Potential_choice = [Animal1,Animal7]

Scoring Algorithm with Weights

Attribute	Percentage	Animal1	Animal2

Age	5%	1(0.05)	1(0.05)
Social	5%	2(0.10)	1(0.05)
Diet	2%	3(0.06)	2(0.02)
Trained	15%	4(0.06)	3(0.45)
PetWorth	5%	5(0.25)	4(0.1)
Allergy	8%	4(0.32)	5(0.4)
Health	15%	2(0.3)	1(0.15)
Housing	12.5%	3(0.375)	2(0.25)
Nocturnal	5%	2(0.10)	3(0.15)
Emotion	5%	3(0.15)	4(0.2)
Parental	10%	4(0.4)	5(0.5)
Stability	12.5%	2(0.25)	1(0.125)
total		2.415	2.445

Animal1_Score = 2.415

Animal2_Score = 2.445

Animal2 is selected!

A unique match will be produced because this Animal2 will now be popped off the stack and the only animals other than this animal will be evaluated for selection for the rest of the clients.

Optimization for Maximal Animal Welfare

Prior to our application, Humans and Animals may have been mismatched in a variety of ways, often animals and humans would be matched and not be fully compatible. Animal welfare is crucial component to our application. As this carleton university animal shelter system(cuACS) goal is to increase the animals welfare we took this into consideration from the beginning of the construction. In order to produce matches we use a variety of calculations and filter passes. Using these may favour the client in certain cases, but we ensure to account for animals welfare by using weighting system.

The weighting system ensures animal welfare. This is done by giving certain attribute values more weight or cost, than other attribute values. We even go as far as to decrease certain attribute values to make up for the increase in the important attribute values. An example of an attribute value that has been increased is Health attribute. We ensure that clients with the intentions of caring for an animal will receive animals with higher health issues so that the animals may get the best care they can get. Another attribute we increased the weight on to ensure animal welfare is the Housing attribute. This attribute will increase the cost of the housing attribute because the housing environment is important for certain animals. Certain animals may require larger homes and that is the preference.

These are the ways we are increasing animal welfare.