|  |
| --- |
| A. S. M. Ahsan-Ul-Haque  [ahsanhaquetarique@gmail.com](mailto:ahsanhaquetarique@gmail.com)  +8801521496399 |

Let p\_list be a list of M persons where person X is also included. For now, let’s consider a person to be a class as described:

Class Person {

Set <Activity> Interest;

}

We construct an interest set S from all person in p\_list.

S = Φ

For each p in p\_list

I = p.Interest;

S = S **U** I

End For

Let, S contain total N activities of interest. We will be using these N activities as features during the training phase. We construct a matrix **F** of dimension MxN as defined below:

Training:

We employ either of the two methods mentioned below:

Method 1:

We can employ this method for training when we do not have explicit knowledge about which person is happy and which person is sad.

**We assume that people chose the activities that tend to make them happier.** We assign weight to each activity and construct the weight vector **W** of dimension Nx1 as described below:

Our hypothesis **h** is an Mx1 vector given by

We find a critical value λ from the dataset for person X and define happiness as,

Method 2:

We can use this method if we have a prior knowledge about which person is happy and which person is sad. We then use a label vector **Y** ofdimension Mx1

Unlike in the previous method, now we use training phase to determine the parameter **W.** The hypothesis vector **h** is defined as,

Now that we can train **W**, we do not need a second parameter λ. We simply define

In other words, λ = 0 for this method.

We may use regularized logistic regression to determine **W** using mean squared error as cost function**.** Any other suitable method such as Decision Tree, SVM, Neural Network etc. may also be employed.

After training, we use the parameter **W** and prediction **h**(X) to outline an algorithm to make a sad person X happy.

Algorithm to make sad person X happy

We use an algorithm similar to Simulated-Annealing as described below:

Function Make-Happy (Person X)

For i= 1: maxiter

Remove-Interest(X)

Add-Interest(X)

End For

If (h(X) < ) then this person refuses to be happy. Recommend professional help.

End Function

Function Remove-Interest(Person X)

I = X.Interest

T = 0

For each interest i in I

Try to remove i from I

If it is successful, then calculate h(X).

If h(X) increases after the removal of i then commit.

Else with probability exp(-T\*(random number between [0,1])) commit

Else discard removal of i

T = T + 1

End For

End Function

Function Add-Interest (Person X)

I = X.Interest

If ( size(I) == N) then no more interest to add. Return.

Else

Randomly select set A containing k interests from S, higher W values of the interest will have higher probability of being picked.

I = I U A

End Function

Remarks:

Method 2 does not make any assumption about a person’s interest and it may be more accurate when some interests of the people tend to cause sadness/unhappiness (such as drugs).

Method 1 will be more effective in cases where we want to rely on data to determine whether a person is sad or happy since method 2 requires a prior knowledge of a person’s state of happiness which may be prone to error.