I. General Idea

Receive 1 disease case, then compare that case with cases in the database to find the case that is most similar to the case that needs to be predicted, the label of that case is the search result. . (If 2 cases are more similar, the degree of similarity is greater. If 2 cases have the same degree of similarity, take any)

II. CBR (Case-based reasoning) method

- We create a weight vector for all habits

$$w_h = (w_{h_1}, w_{h_2}, \ldots, w_{h_{23}})$$

where $h_1, h_2, \ldots, h_{23}$ are 23 habits of disease case

- Create weight vectors for all symptoms

$$w_s = (w_{s_1}, w_{s_2}, \ldots, w_{s_{27}})$$

where $s_1, s_2, \ldots, s_{27}$ are 27 symptoms of disease case

- Create 2 weights for all habits and symptoms

$$W_h, W_s$$

- Suppose we have case a as a predicted case, case b as a confirmed case in the database.

- The habits in case a are:

$$h^a = (h_1^a, h_2^a, \ldots, h_n^a) \, where \, n \leq 23$$

- Symptoms in case a are:

$$s^a = (s_1^a, s_2^a, \ldots, s_n^a) \, where \, n \leq 27$$

- The habits in case b are:

$$h^b = (h_1^b, h_2^b, \ldots, h_n^b) \, where \, n \leq 23$$

- Symptoms in case b are:

$$s^b = (s_1^b, s_2^b, \ldots, s_n^b) \, where \, n \leq 27$$

- The degree of similarity between the attributes of the two cases

$$c_{h_k} = \begin{cases} 1 & if \, h_k \in h^a \, and \, h_k \in h^b \\ 0 & else \end{cases} when \, 1 \leq k \leq 23$$

$$c_{s_k} = \begin{cases} 1 & if \, s_k \in s^a \, and \, s_k \in s^b \\ 0 & else \end{cases} when \, 1 \leq k \leq 27$$

- We have a function to compare the similarity between 2 cases

$$\text{similar(a, b)} = W_h \cdot \sum_{k=1}^{23} w_{h_k} \cdot c_{h_k} + W_s \cdot \sum_{k=1}^{27} w_{hs_k} \cdot c_{s_k}$$

- We use genetic algorithm to optimize the weights

$$w_h,\ w_s,\ W_h,\ W_s$$

III. Genetic Algorithm

a) Gene

We create genes using a vector consisting of a weight vector

$$g = (w_h,\ w_s,\ W_h,\ W_s)$$

b) Fitness function

x_i is an attribute of case i

$\tilde{\text{y}}_i$ is the predicted label of CBR for case i

y is the actual label of the cases

$$\tilde{\text{y}}_i = CBR(gene,\ x_i,\ dataset - \{x_i\}\ )\ for\ i = 1,\ 2,\ \ldots,\ size(dataset)$$

$$acc = accuracy(\tilde{\text{y}},\ y)$$

- acc is the output of the fitness function

IV. Implementation steps

1. Initialize the first generation (250 genes)

2. Selective

- Use the fitness function for all genes in generation, then take the 10 genes with the highest fitness to enter parents.

3. Create a new empty generation, select all genes in parents into the new generation

4. Hybridization

- Use random crossover to randomly mix genes in parents to create 60 genes and add them to the new generation.

5. Mutate the genes in the parent to create 60 new genes and add them to the new generation

6. Calculate the mean of the weights of the genes in the parents, then randomly take 60 genes according to the normal distribution with the mean just calculated and put them in the new generation.

7. Generate 60 completely random genes added to the new generation.

8. The new generation has 250 genes. Continue repeating the above steps until fitness improvement level <= epsilon

- In the initial step of creating the first generation, if we randomly initialize all the weights in the gene in a very large space (23 + 27 + 2 = 52 dimensions). This will create a generation that easily falls into local extremes and is difficult to converge. Therefore, when initializing, it is necessary to propose genes with a high probability of having a reasonable ratio of weights

V. Initialization methods

1. Initializing gene by bayesian method

a) General idea

- Habits or symptoms that play an important role in predicting the disease (have low chaos), will have higher weight than habits or symptoms with high chaos.

b) Method

- Call the habit or symptom to be considered x
- Calculate the conditional probability of each disease given habit or symptom x

$$p(d_i \mid x) \, with \, i \, = \, 1...8$$

- Calculate the disorder of habit or symptom x

$$entropy(x) \, = \, -\sum_{i=1}^{8} p(d_i \mid x) \cdot \, \log\left(p(d_i \mid x)\right)$$

- Calculate the weight of habit or symptom x

$$w_x = \frac{1}{entropy(x) + \epsilon}, with\, \epsilon \leq 0.1$$

- After calculating the weight of habit or symptom x, we use the same method to calculate other weights and create genes. Finally, we use min-max normalization to bring the weights in the gene to the domain [0, 1].

$$w_x = \frac{w_x - w_{x_{\min}}}{w_{x_{\max}} - w_{x_{\min}}}$$

$$Đặt\, W_h = W_s = 0.5$$

=> We obtain gene bayes

- From the gene bayes, we create 50 new genes by sampling from the normal distribution whose mean is the weight of the gene bayes. Note that when sampling from a normal distribution, we must set the clip function to limit it to the region [0, 1].

2. Initializing gene by equal method

a) General idea

- Habits and symptoms play the same role

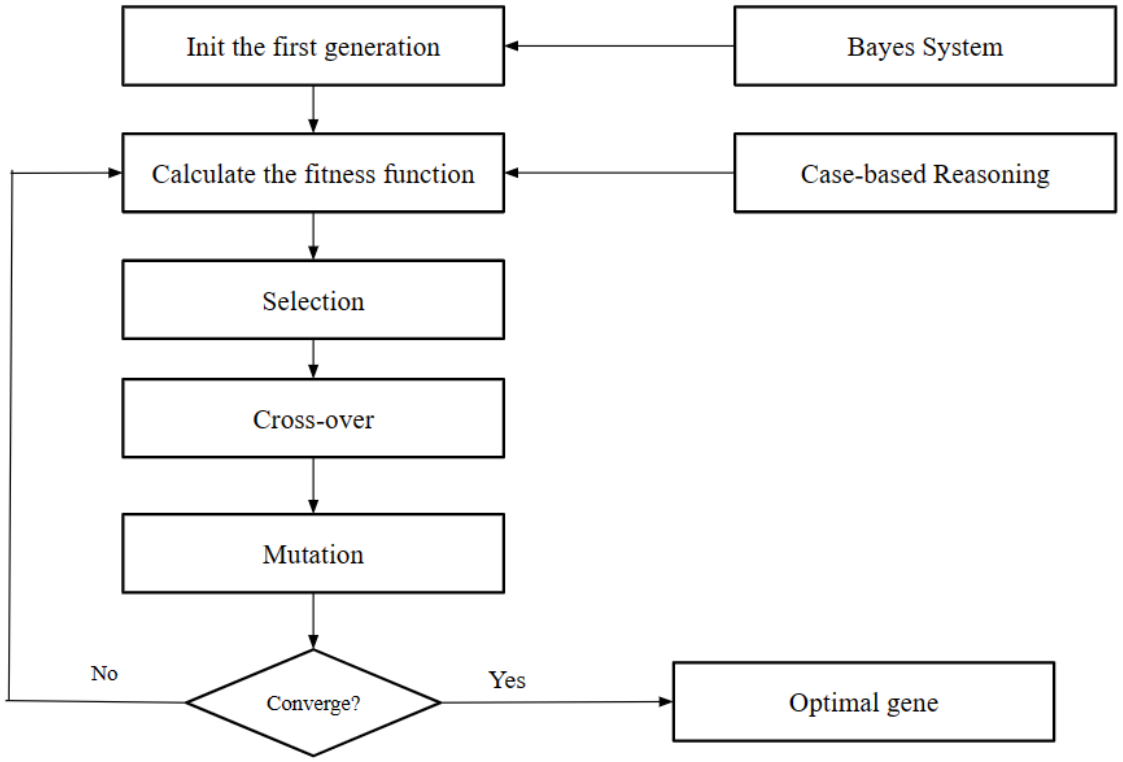
b) Method

$$w_h = w_s = \alpha \cdot (1, 1, \ldots, 1)\, with\, \alpha \sim 0.5$$

$$W_h = W_s = 0.5$$

- From the balanced gene, we create 48 new genes by sampling from the normal distribution whose mean is the weight of the fair gene. Note that when sampling from a normal distribution, we must set the clip function to limit it to the domain [0, 1] and the standard deviation of the distribution is approximately alpha / 2 = 0.25.

VI. Block diagram of the system


Init the first generation
Bayes System
Calculate the fitness function
Case-based Reasoning
Selection
Cross-over
Mutation
No
Converge?
Yes
Optimal gene