

Decision tree dengan Genetic Algorithm

Kriesna Bayu Pratama / 1301170183 / IF41-11

1. Abstraksi

Decision tree merupakan suatu metode pencarian dengan memberikan konsekuensi lanjut setelah suatu keputusan diambil. Decision tree ini akan merujuk pada satu hasil akhir yang menentukan konsekuensi akhir dari keputusan yang diambil. Decision Tree pada kali ini diterapkan pada Genetic Algorithm yang memberikan data secara acak yang kemudian dicari nilai fitness terbaik sehingga menghasilkan hasil terbaik.

2. Datatrain dan Datatest

Diberikan suatu datatrain yang berisikan 80 data yang berisi aturan tertentu untuk diujikan pada rule yang dibuat secara acak. Sedangkan datatest merupakan kumpulan data yang belum memiliki hasil 'Ya' atau 'Tidak'

3. Kromosom dan Populasi

Kromosom tersusun atas bilangan biner dengan Panjang 30 bit yang berarti dua rule. Satu rule merepresentasikan satu set data dengan 4 atribut dan satu hasil 'Ya' atau 'Tidak'. Populasi terdiri dari 20 kromosom yang di generate secara random.

4. Decode Datatrain

Datatrain di decode menjadi bilangan biner dengan ketentuan tertentu. Setiap jenis atribut merepresentasikan satu bit bilangan biner, jadi misal waktu, siang maka akan data tersebut akan berupa 0100 dan untuk malam maka 0001. Jika satu atribut memiliki tiga bentuk seperti normal, tinggi dan rendah maka atribut tersebut akan memiliki 3 digit bilangan biner, sehingga terbentuk 15 digit bilangan biner, yang berisi 3 bit suhu, 4 bit waktu, 4 bit kondisi, 3 bit kelembapan dan 1 bit untuk menentukan terbang atau tidak.

5. Fitness

Nilai fitness ditentukan dengan cara melakukan pengujian dari kromosom yang telah di generate dengan semua data pada datatrain. Pencocokan ini dilakukan dengan cara melakukan mencocokkan tiap atribut dan untuk menentukan hasil kecocokan dari kromosom dengan data train. Apabila pada rule ditemukan nilai 101 pada atribut dan pada data train ditemukan nilai 001 maka, maka hasil akan tetap true dikarenakan 101 pada rule berarti tinggi atau rendah sedangkan pada data train berarti rendah yang mana masih cocok dengan rule. Secara algoritma, perbandingan dilakukan dengan cara melakukan AND antar bit rule dengan data train sehingga menghasilkan satu kromosom/rule baru. Kemudian kromosom tersebut di decode per atribut, kecuali satu bit terakhir. Setelah itu, tiap bit yang ada di dalam atribut di OR satu sama lain sehingga menghasilkan satu bit per atribut. Setelah itu hasil dari OR tersebut di AND satu sama lain sehingga menghasilkan satu bit terakhir. Apabila bit terakhir tersebut 0 maka, langsung dicocokkan dengan bit ya/tidak, jika cocok value+1, jika tidak value tidak bertambah. Apabila satu bit terakhir sama dengan bit terakhir(ya/tidak) dari rule awal maka bit terakhir tersebut dicocokkan dengan bit ya/tidak dari data train. Apabila cocok, maka value + 1 yang mana nantinya akan dijumlah dan dibagi dengan jumlah data untuk menemukan fitness dari kromosom tersebut.

6. Pemilihan Orang Tua

Pada algoritma ini menggunakan Teknik pemilihan orang tua Tournament Selection. Tournament selection ini bekerja dengan cara mengambil suatu *pool* dari populasi yang mana nantinya dari *pool* tersebut akan fitness kromosom satu dengan fitness kromosom yang lain hingga menemukan satu orang tua dengan fitness terbaik. Pool ini berjumlah kurang dari jumlah kromosom dalam populasi sehingga akan mendapat pool yang berisi kromosom acak yang jika dibandingkan akan menghasilkan individu yang berbeda. Tournament Selection ini akan terus menghasilkan parent dengan fitness terbaik. Berbeda dengan roulette wheel yang melakukan pemilihan parent berdasar probabilitas yang akan berubah secara acak, tournament selection memberikan kromosom dengan fitness terbaik dari pool yang diambil.

7. Crossover dan Mutasi

Crossover dilakukan dengan cara memilih 2 titik yang berbeda pada parent 1 dan 2 melakukan pemilihan 2 titik lagi untuk parent kedua. Untuk pemilihan titik dilakukan dengan cara melakukan penentuan 4 titik yang dilakukan dengan menggunakan gap antara 2 titik dan hasil mod dari gap tersebut yang dikombinasi sedemikian rupa sehingga menghasilkan empat pasang titik yang akan dipilih secara acak untuk dilakukan crossover. Hasil crossover bisa menghasilkan kromosom yang lebih panjang ataupun lebih pendek tergantung pada titik yang telah dipilih Mutasi dilakukan dengan cara melakukan invers nilai dari kromosom yang telah dipilih dari parent selection. Jika 0 akan diubah menjadi 1 dan juga sebaliknya. Mutasi ini terjadi dengan probabilitas yang sangat kecil sehingga jarang terjadi.

8. Metode pemilihan Generasi baru

Pada algoritma ini, metode pemilihan generasi baru menggunakan Generational Replacement. Generational Replacement bekerja dengan cara mengambil dua kromosom dengan fitness terbaik dari populasi (Elitism). Elitism dilakukan untuk memastikan tidak terjadi penurunan nilai fitness pada generasi yang baru. Kemudian kedua kromosom tersebut dimasukkan ke dalam list populasi baru yang akan diisi dengan kromosom hasil crossover dan mutasi sedemikian rupa sehingga jumlah kromosom pada populasi baru sama dengan jumlah kromosom pada populasi sebelumnya. Kemudian kromosom pada populasi baru dibandingkan dengan populasi sebelumnya untuk diambil kromosom dengan fitness terbaik sejumlah kromosom pada populasi sebelumnya. Dengan demikian generasi yang baru ini akan berisi kromosom dengan fitness yang terbaik dari populasi lama dan populasi baru. Generational Replacement akan dilakukan berulang kali sampai dengan batas yang telah ditentukan..

9. Pencocokan Rule dengan Fitness terbaik

Pada algoritma ini generasi akan berhenti pada generasi ke sekian ditentukan user. User menentukan proses pemilihan generasi baru akan diulang berapa kali. Variasi generasi akan menghilang saat menemukan fitness terbaik dari populasi awal dan dari hasil crossover dengan cara melakukan perbandingan pada algoritma generational replacement.

Hasil Akhir Program

```
Akurasi : 77,5 %
Data ke : 1 Ya
Data ke : 2 Ya
Data ke : 3 Tidak
Data ke : 4 Ya
Data ke : 5 Ya
Data ke : 6 Tidak
Data ke : 7 Tidak
Data ke : 8 Ya
Data ke : 9 Tidak
Data ke : 10 Ya
Data ke : 11 Ya
Data ke : 12 Tidak
Data ke : 13 Ya
Data ke : 14 Tidak
Data ke : 15 Tidak
Data ke : 16 Ya
Data ke : 17 Ya
Data ke : 18 Tidak
Data ke : 19 Ya
Data ke : 20 Tidak
```