

RI::Projekat

Using evolutionary algorithms to optimize the placement of fuzzy rules in a fuzzy logic system

Dušan Mirčić, Luna Mirčić

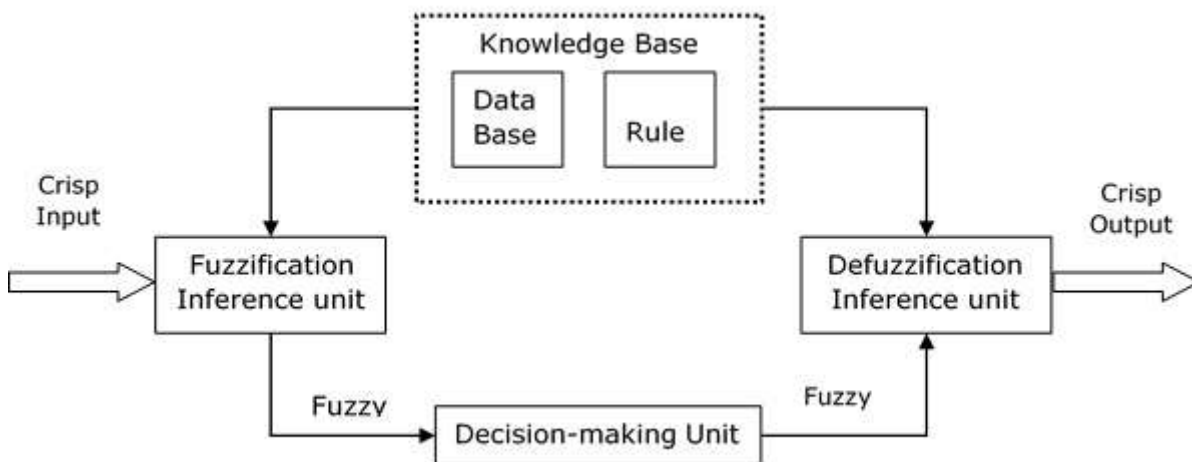
Uvod

apstrakt. Fuzzy logika je predstavljena svetu 1965. godine od strane matematičara po imenu Lotfi Zadeh. 1973, uveo je princip lingvističke promenljive - kojom je bilo moguće predstaviti značenja prirodnog jezika poput visok, nizak, topao, hladan, malo, mnogo i odrediti ulaznim vrednostima stepen pripadnosti i asocijacije ka ovim konstruktima. U ovom radu ćemo objasniti rad sistema koji na osnovu ulaznih i izlaznih vrednosti pokušava da odredi pravila koristeći evolutivnu metodu.

Problem

Ovo je primer jednog fuzzy sistema zaključivanja.

Sastoji se od realne ulazne vrednosti koja ulazi i prolazi kroz proces fazifikacije, primene izabranih pravila, defazifikacije i izlazi napolje kao realna vrednost.



U sklopu ovog rada, bavićemo se samo postavljanjem, korišćenjem i izvođenjem pravila.

Pravila u fazi sistemu mogu biti postavljena na 3 načina:

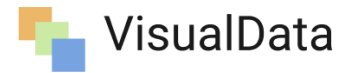
- Ljudski ekspert u polju koji zadaje fuzzy pravila
- Automatizovanim procesom
- Hibridnim pristupom

Fuzzy logika je odličan izbor za implementaciju unutar industrijskih procesa, kontrolora uređaja u domenu bele tehnike, predviđanje nivoa rizika...

Poprilično je univerzalna i široko primenljiva metoda za korišćenje i implementaciju. Jedna od najvećih vrlina sistema baziranih na ovom pristupu je da dobro barataju nepreciznim ulazima.

Medjutim, postiže se isti nivo preciznosti za precizne ulaze, čime nisu idealni za one izlaze/sisteme za koje je veći stepen preciznosti potreban.

Setovi podataka



UC Irvine
Machine Learning
Repository



Kako je fokus ovog rada primarno na postavljanju pravila, možemo koristiti bilo koji set podataka za koji unapred znamo pravila.

U ovom slučaju ćemo koristiti FIS sa predavanja Matematičkog fakulteta u Beogradu iz 2024. godine.

1. Za ovaj primer su nam poznati ulazi, izlazi, kao i pravila.
2. Za nasumičan realan ulaz X , zabeležicemo rezultat defazifikacije, i sacuvati je unutar seta podataka.
3. Obrisaćemo pravila i pokušati da sintetišemo naša.

	A	B	C
1	potrosnja	pouzdanost	vrednost
2	4.8518775873686	9.22930376565318	29.3487395997693
3	2.07230613629045	8.54307419006345	35.7946381447677
4	12.9800348481117	19.5590531126341	15.7531823248492
5	9.34521894785807	4.39578898353388	22.1373771072882
6	9.59446394735581	0.369714139816286	21.2556685023909
7	7.75662536261913	2.8921780575998	31.192155823298
8	13.114500535939	17.2572593655426	15.1704976775978

Sada kada imamo set podataka, kao i metodologiju, moramo osmisлити reprezentaciju za nase resenje. Reprezentacija do koje smo dosli je sledeca:

Reprezentacija gena

Opsti oblik jednog gena $[A_1, A_j, B_i, B_j, O_i, O_j, op]$

U sledećoj tabeli je objašnjeno šta predstavlja svako od ovih polja.

indeks mod 7	naziv	moguca vrednost
0	Leva roditeljska kolona C_1	[0 .. inf]
1	Indeks fuzzy inputa C_1	[0 .. inf]
2	Desna roditeljska kolona D_1	[0 .. inf]
3	Indeks fuzzy inputa D_1	[0 .. inf]
4	Izlazna kolona O_1	[0 .. inf]
5	Indeks posmatranog fuzzy outputa O_1	[0 .. inf]
6	Operator	{0,1}

Kako je resenje koje trazimo skup pravila koji se moze sastojati od jednog ili vise pravila, jedna od mutacija za nasu jedinku ce biti dodavanje novog pravila na kraj postojeceg.

Genetski algoritam

Koraci genetskog algoritma koji ovaj projekat prati su:

1. Inicijalizacija populacije, stvaramo potpuno nasumicne jedinke sa jednim pravilom
2. Za svaku jedinku se racuna fitness na skupu podataka za treniranje
3. Jedinke se sortiraju
4. Izdvoji se najboljih [elitizam] broj jedinki
5. Koristeci najbolje jedinke, formira se nova populacija
6. Ostatak nove populacije se popunjava tako sto se bira iz selekcije metodom turnira
7. Njihovim ukrstanjem dobijamo decu (svako dete ima neku verovatnocu za mutaciju)
8. Kod nas su mutacije brisanje dodavanja ili izmena nekog pravila
9. Dobijenoj deci se racuna fitness i integrisu se u populaciju
10. Crossover - Izaberemo nasumican gen iz jednog roditelja koji ima manje gena (pravila od drugog), prvo dete prepisuje sve do kraja nasumicnog gena iz prvog roditelja, ostatak gena od drugog roditelja se nastavlja od te druge pozicije do kraja
11. Izlaz iz genetskog algoritma je jedinka koja ima najmanji rezultat na kraju izvršenja fitness funkcije

Eksperimentalna resenja

Jedno eksperimentalno resenje dobijeno za gore-sintetisani skup podataka bi bilo zapisano na sledeci nacin:

```
Epoha 424 fitness 4811.674523816739 [0, 2, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0]: 14% | 421/3000  
[00:08<00:58, 44.01it/s]
```

[0,2,1,1,0,0,1,1,0,0,1,0,0,0] (jedinka sa 2 pravila)

Ovo je u stvari kompozicija 2 pravila

[0,2,1,1,0,0,1] i **[1,0,0,1,0,0,0]**

```
Epoha 600 fitness 1859.057811156602 [0, 0, 1, 0, 0, 2, 0, 0, 2, 1, 1, 0, 0, 0]:  
[01:31<00:00, 6.54it/s]  
mala potrošnja AND visoka pouzdanost => velika vrednost  
velika potrošnja AND niska pouzdanost => mala vrednost
```

Izlaz iz programa:

	<bound	method	NDFrame.head	of	potrošnja	pouzdanost	vrednost
0	4.851878		9.229304	29.348740			
1	2.072306		8.543074	35.794638			
2	12.980035		19.559053	15.753182			
3	9.345219		4.395789	22.137377			
4	9.594464		0.369714	21.255669			
..			
221	13.343046		9.688012	13.500000			
222	14.793234		9.806577	9.739985			
223	7.229389		1.938796	36.761858			
224	0.770151		8.387611	37.069054			
225	8.135461		2.030960	28.261232			

[226 rows x 3 columns]>

```
Epoha 700 fitness 1859.057811156602: 100% | 700/700 [00:35<00:00, 19.80it/s]  
visoka pouzdanost AND mala potrošnja => velika vrednost  
velika potrošnja AND niska pouzdanost => mala vrednost  
Epoha 700 fitness 2209.007270440958: 100% | 700/700 [00:40<00:00, 17.41it/s]  
niska pouzdanost OR velika potrošnja => mala vrednost  
visoka pouzdanost AND srednja potrošnja => srednja vrednost  
Epoha 700 fitness 1855.4114977869522: 100% | 700/700 [00:40<00:00, 17.44it/s]  
velika potrošnja AND niska pouzdanost => mala vrednost  
mala potrošnja AND visoka pouzdanost => velika vrednost  
Epoha 589 fitness 2209.007270440958: 84% | 589/700 [00:33<00:06, 16.95it/s]
```

Zaključak

Ovo rešenje daje jako dobre rezultate za skupove sa malim brojem vrednosti, ali se jako rano dodje do konvergencije. Umesto u svrhe automatizacije celog procesa biranja pravila, prisutan je veci potencijal za koriscenje tokom hibridnog pristupa. Ovaj program u velikoj meri pati od nedostatka optimizacija, paralelizacije poslova putem grafickog procesora.

Projekat se moze naci na <https://github.com/duki/FuzzyEvolution>

Reference

#	naziv	izdavač
1	Computational Intelligence: An Introduction	WILEY
2	Fuzzy Logic Applications in Computer Science and Mathematics	WILEY
3	Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications	CRC Publishing
4	Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases	WSPC
5	Genetic algorithms and fuzzy multiobjective optimization	Kluwer Academic Publishers