Analaiza algoritma uniclass

Jure Ham - 63080514

May 30, 2011

Kazalo

1 Opis programa

Program uniclass je celovit paket namenjen testiranju in nadgradji algoritma uniclass. Grafični vmesnik omogoča nalaganje testnih primerov, spreminjanje osnovnih nastavitev algoritma ter sproten prikaz delovanja. Ker je algoritem računsko zahteven, je program implementiran v javi, ki omogoča dober kompromis med hitrostjo izvajanja in hitrostjo razvoja.

1.1 Branje podatkov

Program podpira nalaganje testnih primerov v formatu tab, ki je osnoven format programskega paketa orange. Implementacija je omejena na zvezne in neurejene diskretne vrednosti ter na eno samo meta vrednost, ki postane ime entitete.
Manjkajoči podatki so obravnavani kot posebne vrednosti, kar pomeni, da je razdalja med entiteto z vsemi podatki in entiteto brez podatkov največja, razdalja med dvema entitetama brez podatkov pa je najmanjša. Razlog za to odločitev je zelo enostavna implementacija.

1.2 Računanje matrike razdalj

Razdaljo med dvema primeroma izračunamo kot vsoto vseh razdalj med atributi. V primeru, da je aribut diskreten in neurejena, je razdalja diskretna in sicer 0, 'ce sta vrednosti enaki in 1, če sta vrenosti različni. V primeru, da je atribut zvezen, pa se razdalja inračuna kot

$$dist = \left| \frac{value_1 - value_{min}}{value_{max} - value_1} - \frac{value_2 - value_{min}}{value_{max} - value_2} \right|$$

Kjer je $value_1$ vrednost atributa pri prvem primeru, $value_2$ vrednost atributa pri drugem primeru, $value_{max}$ največja vrednost atributa in $value_{min}$ najmanjša vrednost atributa.
Vse razdalje se pomnožijo še s pomembnostjo atributa, ki je izračunana z algoritmom information gain in se lahko spreminja v grafičnem vmesniku, ter s pomembnostjo vrednosti, ki je definirana kot unikatnost vrednosti. Unikatnost izračunamo na diskretiziranih vrednostih in sicer tako, da izračunamo povrečno število primerov z enako vrednostjo, nato pa to število delimo s številom primerov, ki imajo enako vrednost kot primer, ki nas zanima.

$$uni = \frac{\frac{\| E \|}{\| V_u \|}}{\| E_i = v \|}$$

Kjer je E množica vseh entitet, V_u množica vseh unikatnih vrednosti za atribut in v vrednost za katero računamo unikatnost.
Ko primerjamo dva primera, je unikatnost določena kot produkt unikatnosti prve in druge vrednosti.

1.3 Simuliranje sil

Algoritem uniclass deluje tako, da primere predstavi kot delce v dvodimenzionalnem prostoru. Vsem delcem dodeli enako maso, nato pa med njimi računa sile, ki delce premikajo.

1.3.1 Vztrajnost

Za razliko od algoritma MDS, uniclass upošteva tudi vztrajnost delcev. Vztrajnost računamo tako, da ima vsak delec poleg atributa pozicija tudi atribut hitrost, ki se spreminja glede na sile, ki nanj delujejo. S spreminjanjem mase delcev lahko uravnavamo kaotičnost sistema, saj je vpliv sil na hitrost delca obratno sorazmeren z njegovo maso. Začetna masa delcev je sorazmerna z vsoto vseh sil med delci, tako da kaotičnost sistema ni odvisna od testnih podatkov.

1.3.2 Izračun sile med delcema

Algoritem deli delce na podobne in na ne podobne. Podobni delci so tisti, med katerimi je moč povezave manjša od določene meje, ki je na začetku definirana kot povprečna moč povezave med vsemi delci, kasneje pa jo lahko preko uporabniškega vmesnika tudi spreminjamo. Moč povezave je definirana kot $1 - razdalja$.
Podobni delci se privlačijo s silo, ki je definirana kot $\frac{(f-k)*d^2}{C}$, kjer je f moč povezave, k je prej omenjena meja, d je razdalja med delcema v prostoru, C pa konstanta .
Delci, ki si niso podobni, se odbijajo po formuli $\frac{(f-k)*C}{d}$.
Tako izračunana sila se nato še deli z maso delca. Ker so razdalje lahko zelo majhne, je določena tudi največja možna sila, kar prepreči kaotičnost sistema.

1.3.3 Zaletavanje delcev

Ker algoritem predstavi primere kot delce z maso in radijem, ki je večji od 0, se delci med seboj tudi zaletavajo. S tem prepŕcimo to, da bi se več delcev zbralo na isti oziroma zelo podobni poziciji v prostoru, hkrati pa omogočimo združevanje podobnih delcev, kar bi bilo v nasprotnem primeru zaradi vztrajnosti skoraj nemogoče.
Da se delci, ki si niso podobni, nebi združevali, definiramo togost delcev pri odboju kot obratno verdnost njune podobnosti. Tako dva zelo podobna delca po trku potujeta v skoraj enako smer, dva zelo različna pa vsak v svojo.

1.3.4 Meje polja

Delci, ki z ostalimi primeri nimajo močnih povezav, nam bodo hitro pobegnili izven vidnega polja, saj jih bo večina delcev odbijala. Ta problem vsaj delno rešimo s tem, da okoli polja postavimo mejo, od katere se vsi delci odbijajo. Tako take delce ujamemo na robu polja, kjer se bodo poskušali čim bolj približati delcem, ki so jim podobni in čim bolj oddaljiti od delcev, ki jih odbijajo. Da delci nebi ostali popolnoma prilepljeni na mejo, v algoritem uvedemo krčenje prostora, ki v vsakem ciklu računanje delce malo pomakne proti sredini. Razdalja za katero se delci premaknejo, je odvisna od oddaljenosti od središča.

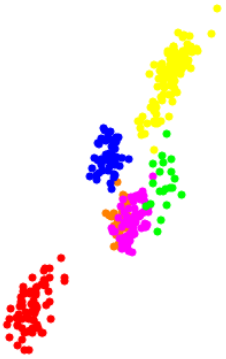
1.4 Barvanje delcev

Za boljšo preglednost delovanja delce pobarvamo glede na razred kateremu pripadajo. Razrede pobarvamo tako, da jih čim bolj enakomirno razporedimo po robu barvnega kroga.

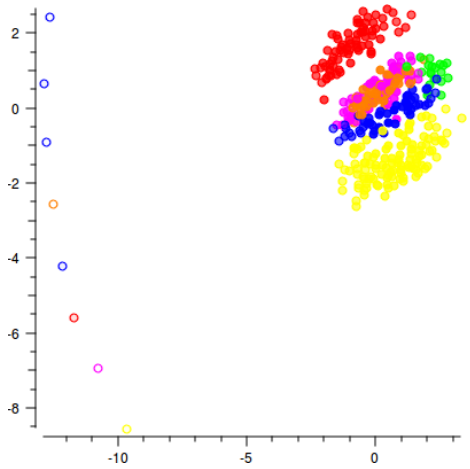
Poleg barvanja razredov delcem dodelimo še dve vizualni lastnosti. Prva lastnost je temnost, ki je odvisna od velikosti sile, ki deluje nanj, druga lastnost pa je rdeča obroba, ki se pokaže, kadar algoritem kNN iz pozicije delca v prostoru ni pravilno določil njegovega razreda.

2 Vizualizacija podatkov

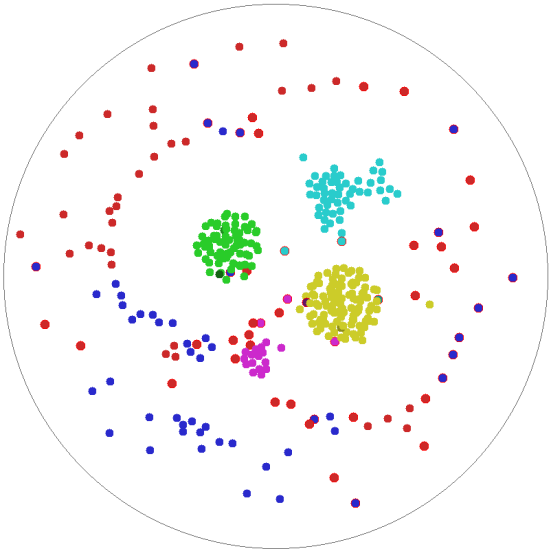
Ker algoritem uniclass obdeluje podatke v dvodimenzionalnem prostoru, je zelo primeren za vizualizacijo podatkov. 1



(a) FreeViz



(b) MDS



(c) uniclass

Slika 1: To je caption

3 Klasifikacija

4 Clustering

5 Zaključek