Project title: Machine learning algorithms for insightful analysis of complex data structures

Objective: Hardware implementation of rule learning procedures (W3)

Activity: Implementation of interface between FPGA and software realization of machine learning algorithms (Month 1-7)

**Procedural analysis of machine learning procedures** (P. Škoda, završeno u studenom 2014.)

Cilj i metodologija

Cilj provedene analize postupaka strojnog učenja je skupiti saznanja o značajkama koje će pomoći u odabiru najboljih kandidata za njihovu realizaciju u sklopovlju. U tu svrhu analizirana su četiri algoritma koja spadaju među najvažnije algoritme u strojnom učenju/dubinskoj analizi podataka:

* C4.5 - algoritam za konstrukciju stabla odlučivanja,
* k-means - algoritam za grupiranje (clustering),
* Support vector machines (SVM) - algoritam za klasifikaciju.

Svaki algoritam obrađen je teoretski, te je pregledana po jedna programska implementacija algoritma. Posebna pozornost posvećena je ponašanju algoritma i njegove implementacije pri obradi velikih skupova podataka. Razlog tome je da zbog troškova komunikacije između programa i FPGA sklopovlja, što je već dokazano u dosadašnjim istraživanjima, za male skupove nema dobitka u vremenu izvođenja programa/algoritma.

Iz provedene analize navedeni su zaključci o računsko/vremenski najintenzivnijim procesima u navedenim algoritmima, te sličnostima i razlikama tih procesa.

C4.5

C4.5 je jedan od najpoznatijih algoritama za konstrukciju stabla odlučivanja. Algoritam se koristi za učenje klasifikacijskih stabla odlučivanja.

Algoritam rekurzivno dijeli skup za učenje na manje podskupove, prema vrijednosti određenog atributa, dok se ne dostignu čisti podskupovi ili dok podskupovi ne postanu premali za daljnje dijeljenje. Ključan, a i računski najintenzivniji, dio algoritma je odabir atributa prema kojem se skup dijeli. Proces odabira atributa može se u grubo podijeliti na dva dijela:

* izračun matrice frekvencija za svaki atribut,
* izračun informacijske dobiti (information gain) i faktora informacijske dobiti (information gain ratio) za svaki atribut.

Algoritam različito postupa s atributima različite vrste (nominalni, numerički). Kod nominalnih atributa skup se dijeli na onoliko podskupova koliko ima jedinstvenih vrijednosti atributa. Prebrojavanjem primjera određene klase koji imaju određenu vrijednost nominalnog atributa određuje se matrica frekvencija.

Izračun matrice frekvencija u biti je postupak težinskog prebrojavanja parova (vrijednost, klasa) u skupu. To se svodi na pribrajanje težine primjera u ćeliju matrice koja odgovara njegovom paru (vrijednost, klasa). Ta je matrica ulaz u postupak izračuna informacijske dobiti.

Postupak izračuna informacijske dobiti numerički je složeniji od izračuna matrice frekvencija, jer u sebi uključuje operacije dijeljenja, te izračuna logaritma.

Iako je izračun matrice frekvencija numerički jednostavan, zbog ulaznog volumena podataka, za velike skupove on predstavlja većinu računskog vremena. Izračun informacijskog dobitka, unatoč većoj numeričkoj složenosti, uzima manji dio vremena izvršavanja, osim za najmanje skupove.

Izračun matrice frekvencija radi se s cijelim skupom (odnosno podskupom), dok izračun informacijskog dobitka radi s matricom frekvencija, koja je najčešće znatno manja od obrađivanog (pod)skupa.

k-means

K-means algoritam je jedan od najpoznatijih algoritama za grupiranje (clustering), i spada u skupinu algoritama za nenadzirano učenje (unsupervised).

Algoritam se izvršava na skupu vektora (ulazni skup podataka mora biti prikazan kao matrica), a cilj je vektore grupirati u k grupa. Grupe su opisane kao skup vektora koji predstavljaju centre grupa.

Dva glavna procesa algoritma su:

* grupiranje - svi vektori u skupu grupiraju se oko trenutnih centara,
* ažuriranje centara, tj. izračun novih centara iz trenutnih grupa.

Ta dva procesa izvršavaju se iterativno dok se grupe ne konvergiraju u stabilno stanje, ili dok se ne dosegne određeni broj iteracija.

Računski najintenzivniji dio postupka grupiranja je izračun udaljenosti vektora od centara grupa. Izračunate udaljenosti međusobno se uspoređuju, te se vektor dodijeli grupi čijem je centru najbliži. Kao mjera udaljenosti uglavnom se koristi euklidska udaljenost. Druge mjere su moguće, ali se koriste rijetko. U osnovnom algoritmu nužno je izračunati udaljenosti između svakog mogućeg para (vektor – centar), ali postoje modifikacije algoritma (npr. filtriranje pomoću k-d stabla) kojima se smanjuje broj parova za izračun udaljenosti.

Ažuriranje centara svodi se na izračun srednje vrijednosti vektora koji pripadaju istoj grupi. Taj proces je računski znatno manje intenzivan od procesa grupiranja.

Za veće skupove podataka algoritam se ubrzava korištenjem k-d stabala, koja omogućuju izbjegavanje izračuna udaljenosti između svakog vektora i svakog centra, te tako smanjuje obim izračuna. Također, koraci algoritma mogu biti spojeni, tako da se novi centri inkrementalno izračunavaju istovremeno s grupiranjem.

SVM

Strojevi s potpornim vektorima (engl. Support vector machines - SVM), jedan su od najuspješnijih i široko korištenih algoritama za klasifikaciju.

Klasifikator je definiran putem potpornih vektora (support vectors) - minimalnog podskupa vektora iz skupa za učenje - i koeficijenata. Oni zajedno definiraju hiperravninu koja najbolje razdvaja skup na dva dijela. Ukoliko skupovi nisu linearno separabilni, može se koristiti jezgrena funkcija (kernel function) kojom se vektori preslikavaju u prostor u kojem je skup linearno separabilan.

Učenje SVM klasifikatora je problem kvadratne optimizacije i postoji više metoda kojim se rješava. Jedna od najpoznatijih i najefikasnijih metoda je SMO - sequential minimal optimization. U ovoj metodi problem kvadratne optimizacije svodi se na niz minimalnih kvadratnih problema, tj. problema s dva koeficijenta/potporna vektora koji se rješavaju analitički. Odabir (dvaju) koeficijenta koji se optimiraju u svakoj iteraciji algoritma određuje se heuristički, a o odabranoj heuristici ovisi koliko će iteracija biti potrebno da se dođe do rješenja.

Svaka iteracija algoritma sadrži provjeru uvjeta konvergencije unutar koje se izvodi izračun skalarnog produkta, što je računski najintenzivniji dio procesa. Ako se koriste nelinearne jezgrene funkcije, numerička složenost, a njome i računska intenzivnost, tog procesa se povećava. Često korištene jezgrene funkcije (uz linearnu) su polinomijalna, te Gaussova (RBF - radial basis function).

Zaključak

Iz provedene analize vidljivo je da su algoritmi u velikoj mjeri različiti u smislu temeljnih procesa koje koriste pri izvršavanju. Od računski najintenzivnijih procesa, najizgledniji kandidati za izvedbu u sklopovlju su:

* težinsko prebrojavanje parova (vrijednost, klasa),
* izračun udaljenosti između dva vektora,
* skalarni produkt,
* izračun jezgrene funkcije.

Jedina pronađena dodirna točka je specijalan slučaj za SVM s Gaussovom jezgrenom funkcijom i k-means s euklidskom udaljenošću. U tom je slučaju proces izračuna euklidske udaljenosti dva vektora, kao dio izračuna Gaussove funkcije, zajednički za ta dva algoritma.

S obzirom na manjak zajedničkih procesa u analiziranim algoritmima, za identifikaciju temeljnih procesa korištenih u strojnom učenju potrebno je učiniti analizu koja uključuje znatno veći broj algoritama. No ni to nije garancija da bi identifikacija i izvedba tih procesa u sklopovlju dala željeni učinak u smislu ubrzanja izvršavanja algoritma. Isti procesi kod različitih algoritama izvršavaju se u različitim kontekstima koji mogu imati značajne posljedice na efikasnost njihova izvođenja u sklopovlju. Dodatno, izvedbu dijela algoritma u sklopovlju najčešće prate i znatne izmjene u implementaciji algoritma, a nerijetko i promjene u samom algoritmu.

Iz navedenih je razloga ustaljena praksa razvijati sklopovlje ciljano za pojedini algoritam. Pri tome se samo za temeljne matematičke procese mogu koristiti unaprijed razvijene jezgre (cores), ako to dopušta razvojni sustav. Stoga je teško za očekivati da će procedure implementirane u ranoj fazi projekta biti izravno korištene za implementaciju heterogenog sustava za *rule induction*, ali se očekuje stjecanje znanja koje će doprinijeti boljoj konačnoj izvedbi sustava.