Fakultet elektrotehnike i računarstva Sveučilišta u Zagrebu
Strojno učenje, ak. god. 2011/12. – Upute za korištenje Rapid Minera (v1.0)

Upute za korištenje Rapid Minera

1 Uvod

Rapid Miner jedan je od najpopularnijih slobodno dostupnih alata za strojno učenje i dubinsku analizu podataka. Koristi se za edukaciju, brzo razvijanje prototipa, ali i razvoj profesionalnih aplikacija.

Kratak opis prvog pokretanja i sučelja Rapid Minera može se naći u ovom videu:

http://rapid-i.com/videos/tutorials/01_rm5_gui/rm5_gui.html

2 Uvoz podataka

Da bismo neki skup podataka mogli koristiti u Rapid Mineru, najprije moramo napraviti uvoz podataka. To se može napraviti na više načina, kao što je objašnjeno u ovom videu:

http://rapid-i.com/videos/tutorials/02_rm5_data_import/rm5_data_import.html

U nastavku je opisan postupak uvoza podataka iz datoteke u kojoj su vrijednosti odvojene zarezima (CSV-datoteka). Prvo je potrebno pokrenuti čarobnjaka za uvoz CSV-datoteke (*Import CSV Wizard*) odabirom opcije *Import CSV File* u alatnoj traci prozora *Repositories*. Uvoz se zatim provodi u pet koraka:

1. Odabire se CSV-datoteka iz koje se uvoze podaci.
2. Definira se kako treba interpretirati ulaznu datoteku. Najvažniji skup postavki je *Column Separation*, gdje se određuje na koji način su podatci odvojeni. Umjesto zareza može se koristiti tabulator, točka-zarez ili neki regularni izraz.
3. Ovaj korak omogućuje uvoz nekih podataka o stupcima, ako oni postoje. Primjerice, ako prvi red skupa podataka sadrži imena stupaca, to je moguće ovdje definirati.
4. Sada je potrebno detaljnije opisati skup podataka: za svaki od stupaca potrebno je odrediti ime, tip i ulogu.
   (a) Ime stupca – daje ime stupcu (npr. visina, dob, boja očiju). Ako smo u prethodnom koraku definirali redak koji sadrži imena svih stupaca, odgovarajuće vrijednosti bit će upisane automatski.
   (b) Tip stupca – opisuje tip podatka koji poprima pojedini atribut; mogući tipovi opisani su u dijelu 3.
   (c) Uloga stupca – opisuje ulogu koju pojedini stupac ima u skupu podataka. Neke od češće korištenih uloga su:
      - *Attribute* – stupac ima ulogu značajke i koristi se u učenju;
      - *Label* – stupac ima ulogu ciljne vrijednosti, tj. definira klasu kojoj pripada pojedini redak;
      - *Id* – stupac ima ulogu identifikatora i ne koristi se u učenju (npr. ime osobe).

Velik broj ovih podataka popunjava se automatski, pogađanjem na temelju vrijednosti iz skupa za učenje. Premda su automatski određene vrijednosti najčešće ispravne, ipak ih treba provjeriti prije konačnog uvoza podataka.

5. Na kraju, potrebno je odabrati ime repozitorija u koji se sprema skup podataka (obično je to lokalni repozitorij definiran pri instalaciji) i ime novog skupa podataka koji će nastati uvozom. Nakon uvoza, skup podataka može se naći u prozoru *Repositories* i povući na radnu površinu te dalje koristiti.

3 Predobrada podataka

U Rapid Mineru svaki stupac u skupu podataka ima definiran svoj tip. Neki od važnijih tipova su:

1. *Numerical, Real, Integer* – brojevi (npr. *visina*);
2. *Polynominal, Nominal* – značajke s konačnim brojem diskretnih vrijednosti, odnosno značajke koje odgovaraju *multinomijalnim* varijablama (npr. *boja* – *crvena*, *zelena* i *plava*);
3. *Binominal* – značajke s točno dvije diskretne vrijednosti, odnosno značajke koje odgovaraju *Bernoullijevim varijablama* (npr. *istina* i *laž*).

Prije primjena metoda strojnog učenja ponekad je potrebno podatke pripremiti na odgovarajući način. U nastavku su opisani neki često korišteni operatori za pretvorbu podataka. Svaki od navedenih operatora mogu djelovati na sve atribute ili na odabrani podskup atributa.

3.1 Pretvaranje nominalne značajke u numeričku

Neke klasifikacijske metode (npr. SVM) mogu raditi isključivo s numeričkim vrijednostima. Ako su na raspolaganju nominalne (diskretne) značajke, treba ih pretvoriti u numeričke. To je moguće učiniti na dva načina:

1. Uporabom bloka *Nominal to Numerical*, koji svakoj od mogućih vrijednosti značajke dodjeljuje neki cijeli broj. Problem s ovom metodom jest što će inicijalno jednako udaljene nominalne vrijednosti pretvoriti u cijele brojeve koji više nisu jednako udaljeni. Npr., ako se vrijednosti varijable *boja* preslikaju tako da *crvena*$\rightarrow$ 1, *zelena*$\rightarrow$ 2, *plava*$\rightarrow$ 3, onda će u ulaznome prostoru crvena biti bliža (a time sličnija) zelenoj nego plavoj, što u model uvodi neželjenu pristranost. Zbog toga nije dobro pretvorbu provoditi na ovaj način.
2. Uporabom bloka *Nominal to Numerical*, uz postavljenu opciju *dummy coding*. Ovaj blok svaku nominalnu značajku sa V vrijednosti pretvara u V-dimenzijski vektor binarnih numeričkih značajki. Budući da se svi parovi tako dobivenih vektora međusobno razlikuju u točno dvije komponente, vrijednosti su jednako udaljene u ulaznome prostoru i ne uvodi se nikakva dodatna pristranost. Ovo je preferirani način pretvaranja nominalne značajke u numeričku.

3.2 Pretvaranje numeričke značajke u nominalnu

Neke klasifikacijske metode (npr. Bayesov klasifikator) mogu raditi isključivo s nominalnim značajkama. Ako skup podataka sadrži numeričke značajke, treba ih pretvoriti u nominalne, odnosno treba ih diskretizirati. U tu svrhu može se koristiti nekoliko operatora diskretizacije od kojih se najčešće koriste *Discretize by Binning*, *Discretize by Size* i *Discretize by Frequency*.

3.3 Normalizacija podataka

Često je važno i korisno prije primjene metoda strojnog učenja koje rade s numeričkim ulazima (npr. SVM) skalirati podatke tako da oni budu centrirani oko nule ili da budu ograničeni na zadani interval. Za to se može iskoristiti blok *Normalize* koji u svojim opcijama nudi nekoliko algoritama za normalizaciju.

4 Učenje i vrednovanje modela

Učenje modela ostvaruje se tako da se skup podataka dovede na ulaz bloka koji na svome izlazu daje naučen model (npr., blok koji odgovara modelu stabla odluke). U slučaju da model ne može raditi s više klasa, može ga se omotati u blok *Polynomial by Binomial Classification*, koji implementira višeklasnu shemu *jedan-naspram-jedan* ili *jedan-naspram-ostali*.

Primjena naučenog modela za označavanje skupa podataka obavlja se pomoću bloka *Apply Model*. Točnost modela može se izmjeriti uporabom bloka *Performance*.

4.1 Unakrsna provjera

Unakrsna provjera – k-struka višestruka provjera (engl. *k-folded cross validation*) ili provjera "izdvoji jednoga" (engl. *leave-one-out cross validation*) – može se jednostavno izvesti pomoću bloka *X-Validation*. Funkcionalnost tog bloka opisuje sljedeći pseudokod:

Podijeli skup za učenje $\mathcal{D}$ na k podskupova (preklopa) $\mathcal{D}_1, \ldots, \mathcal{D}_k$
za $i = 1$ do k
    Treniraj model na skupu $\bigcup_{j=0, j \neq i}^{k} \mathcal{D}_j$
    Ispitaj točnost modela na skupu $\mathcal{D}_i$
Vrati prosjek točnosti modela po svim iteracijama (preklopima)

Blok *X-Validation* je ugniježđen i sastoji se od dva potprocesa:

1. potproces *Train*, u kojemu se model uči na skupu za učenje;
2. potproces *Test*, u kojemu se naučeni model vrednuje na skupu D_i.

Ako je potrebno, rezultati svake pojedine iteracije mogu se pogledati tako da se postavi prekidna točka (engl. *breakpoint*) na neki od blokova unutar bloka *X-Validation* (u kontekstnom izborniku bloka na koji želimo staviti staviti prekidnu točku).

Na izlazu *mod* dobiva se model naučen na cijelom ulaznome skupu. U slučaju da unakrsnu provjeru ne želimo provoditi višestruko (u petlji), već samo jednom (metoda izdvajanja), treba upotrijebiti blok *Split Validation*. Ilustracija ovdje iznesenog uz konkretne primjere može se vidjeti u sljedećem videu:

http://rapid-i.com/videos/tutorials/05_rm5_evaluation/rm5_evaluation.html

5 Odabir modela

Kod modela koji imaju hiperparametre (npr. SVM i k-NN) potrebno je provesti optimizaciju tih hiperparametara kako bi se odabrao model optimalne složenosti. U Rapid Mineru to se može ostvariti uporabom bloka *Optimize Parameters (Grid)*, koji iscrpno pretražuje sve zadane kombinacije parametara odabranih unutarnjih blokova. Blok funkcionira na način da ispituje svaku kombinaciju parametara i vraća onu koja se pokaže najboljom. To je ugniježđen blok u čiji jedini potproces treba smjestiti nešto što će izračunavati mjeru točnosti. U tu svrhu mogu se koristiti blokovi *Split Validation* ili *X-Validation*. Kombinacije parametara koje se ispituju mogu se specificirati u opcijama bloka *Optimize Parameters* (obično se variraju parametri klasifikatora u unutarnjem bloku).

Blok *Optimize Parameters (Grid)* na svom izlazu *par* daje optimalne parametre. Dodatno, do izlaza bloka možemo provući izlaz *mod* iz unutarnjeg bloka. Tako ćemo kao dodatan izlaz dobiti model učen na ukupnom ulaznom skupu s optimalnim parametrima (blok *Optimize Parameters* na svoj izlaz prosljeđuje unutarnje izlaze baš za onu kombinaciju parametara koja se pokazala najboljom).

Osim *Optimize Parameters (Grid)* dostupni su i drugi blokovi koji ne pretražuju iscrpno nego koriste npr. evolucijske strategije – *Optimize Parameters (Evolutionary)*.

5.1 Ugniježđena unakrsna provjera

Kada bismo odabir modela proveli na cijelom skupu podataka, dobili bismo nerealno optimistične rezultate. Za realnu procjenu učinkovitosti modela postupak odabira modela ne smije se provoditi na ispitnome skupu, već treba koristiti unakrsnu provjeru. Budući da se jedna unakrsna provjera već izvodi za odabir modela, ovo će biti druga unakrsna provjera na višoj razini, čime zapravo dobivamo ugniježđenu unakrsnu provjeru (engl. *nested cross validation*). Ona se u Rapid Mineru može ostvariti tako da se cijeli blok *Optimize Parameters* omota u *X-Validation*. Prisjetimo se, na izlazu bloka *Optimize Parameters* možemo dobiti model učen optimalnim parametrima na cijelom ulaznom skupu te taj model možemo koristiti u potprocesu *Training* bloka *X-Validation*. Tako smo zapravo implementirali vanjsku petlju ugniježđene unakrsne provjere. Petlja po hiperparametrima predstavljena je blokom *Optimize Parameters*, dok je unutarnja validacijska petlja predstavljena unutarnjim blokom *X-Validation* (unutar bloka *Optimize Parameters*).

6 Odabir značajki

Često je korisno od mnogo značajki dostupnih za neki problem strojnog učenja odabrati one koje su najpogodnije. Za to koristimo algoritme odabira podskupa značajki (engl. *feature subset selection algorithms*). Dva su osnovna pristupa odabiru značajki: odabir filtrom (engl. *filter selection*) i odabir omotačem (engl. *wrapper selection*).

6.1 Filtar

Odabir pomoću filtra zasniva se na promatranju skupa podataka i ocjene važnosti atributa temeljem neke mjere. Mjere koje se koriste obično su zasnovane na statistici ili teoriji informacija. Nedostatak ovakvog pristupa jest što ocjenjuje svaku značajku samu za sebe, pa ne uzima u obzir međuovisnosti među značajkama (ovaj je problem moguće riješiti uporabom multivarijatnih filtra). Drugi problem jest što se filtrom značajke odabiru neovisno o upotrijebljenom modelu, ne uvažavajući specifičnosti pojedinih modela.

6.2 Omotač

Kod omotača postupak odabira značajki "omotan" je oko modela: za svaki odabrani podskup značajki provodi se učenje i ispitivanje modela. Postupak optimizacije vođen je mjerom točnosti modela, pa su odabrane značajke potpuno prilagođene modelu. Nalaženje optimalnog podskupa značajki iziskivalo bi ispitivanje svih podskupova početnog skupa značajki, a to najčešće nije vremenski izvedivo. Zbog toga se koriste heuristički postupci pretrage.

U Rapid Mineru odabir značajki može se ostvariti uporabom bloka *Optimize Selection*. Taj blok može raditi na dva načina:

1. *Odabir unaprijed* (engl. *forward selection*) – Krenuvši od praznog skupa značajki, u svakoj se iteraciji odabire ona značajka čije uključivanje u skup značajki dovodi do najvećeg povećanja točnosti modela. Postupak se ponavlja sve dok više nema značajki koje bi dale značajno poboljšanje točnosti;

2. *Odabir unazad* (engl. *backward selection*) – Krenuvši od skupa svih značajki, u svakoj se iteraciji odabire ona značajka čijim se uklanjanjem iz skupa značajki najmanje kvari točnost modela. Postupak se ponavlja sve dok više nema značajke čijim se uklanjanjem točnost modela ne bi značajno smanjila.

Blok *Optimize Selection* na svom izlazu *wei* daje težine optimalnih značajki (1 ako je značajka odabrana, 0 inače), a na izlazu *exa* cijeli ulazni skup podataka filtriran tako da su zadržane samo odabrane značajke. Ovo je ugniježđen blok u čiji jedini potproces treba smjestiti nešto što će voditi pretragu, tj. računati točnost modela. Kao unutarnji proces najčešće se koristi blok *X-Validation*. Konkretan primjer opisan je u sljedećem videu:

`http://www.youtube.com/watch?v=JlhoTAk1ow8`

Osim *Optimize Selection* dostupni su i drugi blokovi za odabir optimalnog skupa značajki koji pretražuju sve moguće podskupove značajki – blok *Optimize Selection (Brute Force)* – ili koriste evolucijske strategije za pretragu – blok *Optimize Selection (Evolutionary)*.

6.3 Odabir značajki i unakrsna provjera

Slično kao i kod odabira modela, kada bismo odabir značajki proveli na cijelom skupu podataka, dobili bismo nerealno optimistične rezultate. Za realnu procjenu pogreške modela postupak odabira značajki ne smije se provoditi na ispitnome skupu, već treba koristiti unakrsnu provjeru. Kombinacija odabira značajki i unakrsne provjere u Rapid Mineru je implementirana u bloku *Wrapper-Split-Validation*. Taj se blok sastoji od tri potprocesa:

1. *Weight* – U ovom potprocesu traže se optimalne značajke. Izlaz ovog potprocesa mora biti vektor težina (engl. *feature weights*). Taj se vektor može dobiti kao izlaz bloka *Optimize Selection* (odabir omotačem) ili kao izlaz nekog od drugih blokova za odabir značajki (odabir filtrom);

2. *Train* – U ovom potprocesu model se uči na skupu za učenje koristeći optimalne značajke;

3. *Test* – U ovom se potprocesu naučeni model s optimalnim značajkama vrednuje na skupu za ispitivanje.

Ugniježđena unakrsna provjera može se, kao i obična unakrsna provjera, provesti višestruko, odnosno u k preklopa. Ta je varijanta implementirana u bloku *Wrapper-X-Validation*. Uočite da, ako se u prvom potprocesu (*Weight*) koristi odabir omotačem, efektivno dobivamo ugniježđenu k-struku unakrsnu provjeru (engl. *nested k-fold cross validation*).