2. domača naloga: logistična regresija

Jernej Henigman (kaggle: JernejHenigman)

21. marec 2016

1 Uvod

V 2. domači nalogi je bilo potrebno implementirati logistično regresijo z L2 regularizacijo.

2 Metode

Opomba: v nalogi nisem uporabil modula Orange.

- **Nalaganje podatkov**
  Shranimo 3 "pickle"objekte. Učno množico (trainX), testno množico (testX), razrede učne množice (trainY). Podatki so iz portala Kaggle (Digit Recognizer). TrainX in trainY - 42k primerov. TestX - 28k primerov.

- **Predprocesiranje podatkov**
  Odstranimo nepomembne atribute. Torej tiste stolpce v testni množici (trainX), ki so konstantni, ali pa skoraj konstantni. Pomembno je da odstranimo iste stoplce tudi iz testne množice (testX). Vrednosti v učni in testni množici normaliziramo. Ko naredimo te dve operaciji nad učnimi in testnimi podatki opazimo, da traja učenje na podatkih precej manj časa. Napovedi so tudi bolj točne.

- **Primerjava gradientov**
  Z metodo končnih diferenc preverimo, če je analitično izračunan gradient pravilen. Ugotovimo, da pri istem naboru vhodnih podatkov, obe metodi vrneta enake rezultate.

- **Tehnika eden proti vsem**
  Posamezen primer lahko na naših podatkih (Digit Recognizer) zajame vrednost od nič do devet. Torej obravnavamo problem, ki vsebuje 10 različnih razredov. Logistična regresija lahko razlikuje samo med dvema razredoma (0 ali 1). Zato uporabimo tehniko eden proti vsem, kjer v 10-ih iteracijah izračunamo 10 tabel parametrov (theta). V vsaki iteraciji učenja (računanje thet (fmin-l-bfgs-b)) enemu razredu dodelimo vrednost 1 ostalim devetim pa vrednost 0. Ko delamo predikcije,

```
for i in range(n_classes):
        y_p[i] = testX.dot(thetas[i])
```

pogledamo za vsak primer, v katerem stolpcu v matriki `y_p` se nahaja vrednost z najvišjo vrednostjo. V našem primeru je številka stolpca (z najvišjo vrednostjo za posamezne primer), kar napovedan razred naše logistične regresije.

- **Iskanje parametra pri L2 regularizaciji**
  Najbolj ustrezen parameter 'alpha' za regularizacijo izberemo v več fazah. Za 6 vrednosti alpha (od -60 do 40, s korakom 20) preverimo, kje je točnost napovedi najvišja. Pri alpha = 0, dobimo najvišjo točnost napovedi. Določimo nov nabor parametrov alpha v okolici predhodno izračunan alphe (od -0.3 do 0.2, s korakom 0.1). Tokrat dobimo najvišjo točnost napovedi pri 0.1. Ponovimo postopek še za vrednoti alpha med 0.04 in 0.12. Rezultati so prikazani v spodnji tabeli.
  Pri napovedovanju uporabimo 3-kratno prečno preverjanje na učnih podatkih. Parameter alpha iščemo na podmnožici učnih podatkov. Velikost množice, lahko razberemo iz tabele v poglavju rezultati.

3 Rezultati

alpha	n = 1000	n = 10000	n = 20000
-60	0.107	0.755	0.775
-40	0.0107	0.395	0.715
-20	0.683	0.717	0.789
0	0.79	0.8164	0.825
20	0.681	0.8064	0.825
40	0.63	0.780	0.802
-0.3	0.669	0.7682	0.7811
-0.2	0.422	0.651	0.871
-0.1	0.756	0.778	0.792
0.1	0.838	0.898	0.903
0.2	0.831	0.894	0.8993
0.04	0.846	0.900	0.906*
0.06	0.848	0.898	0.904
0.08	0.842	0.898	0.904
0.10	0.838	0.898	0.903

Tabela 1.1. Napovedane točnosti logistične regresije, pri različnem naboru parametra alpha, za 3 različne velikosti učnih podatkov (n primerov). Uporablja se 3-kratno prečno preverjanje.

Na koncu najboljši parameter uporabimo pri napovedovanju na celotni testni množici (Digit Recognizer, 28k primerov). Predikcije oddamo na portalu kaggle.