# Uvod v odkrivanje znanj iz podatkov



## 4. domača naloga: logistična regresija

**1. (40 točk)** Implementirajte logistično regresijo. Uporabite priloženo ogrodje ter ga dopolnite z:

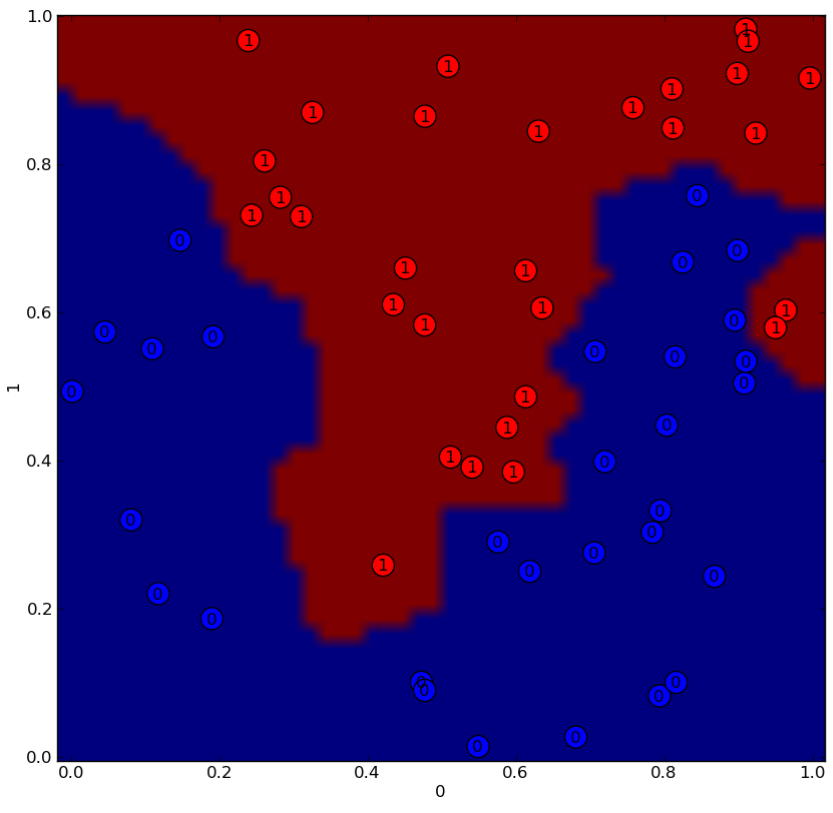
- izračunom verjetnosti ciljnega razreda za posamezni primer, *h*,
- cenilno funkcijo, *cost*,
- analitičnim gradientom cenilne funkcije, *grad* in
- numeričnim gradientom, *num_grad*.

Gradnja napovednega modela in napovedovanje razreda posameznim primerom sta ločeni: razred *LogRegLearner* iz učnih podatkov zgradi napovedni model tipa *LogRegClassifier,* ki lahko nato za poljuben vektor značilk napove verjetnosti obeh razredov. Ogrodje rešuje optimizacijski problem s funkcijo fmin_l_bfgs_b (oglejte si preprost primer uporabe), zato morate le implementirati funkcije *cost*, *grad* in *num_grad*. Vašo implementacijo shranite v eno samo datoteko in jo poimenujte solution.py.

Preverite pravilnost vaših implementacij:

- Ker vemo, da lahko cost in grad množimo s poljubno konstanto, prilagodite vašo rešitev testom. Najverjetneje bo treba deliti s številom primerov. Datoteko s testi shranite v direktorij z vašo rešitvijo.
- Če gradnjo modela logistične regresije brez regularizacije poženete na celotnih podatkih reg.data, vam mora zgrajen model vse primere uvrstiti prav tako, kot je zapisano v reg.data, napovedna točnost na učnih primerih pa mora biti enaka 1.0.

**2. (25 točk)** Uporabite kodo za izris napovedi. Na celotnem podatkovnem naboru reg.data (brez regularizacije) morate dobiti spodnji izris, kjer točke označujejo primere posameznih razredov, barva ozadja pa verjetnost napovedi v tistem delu prostora. Na abscisi je vrednost prve značilke (indeks 0), na ordinati pa druge (indeks 1).




Raziščite vpliv regularizacije na napovedi, tako da spreminjate vrednosti parametra `lambda`. Izrišite tri zanimive in v rezultatih različne stopnje regularizacije in jih shranite v samostojnem enostranskem pdf dokumentu.

**3. (35 točk)** Implementirajte k-kratno prečno preverjanje kot funkcijo `test_cv(learner, X, y, k)`, ki vrne napovedi za vse primere (napovedi so verjetnosti za oba razreda) v enakem zaporedju kot so v X, le da nikoli ne uporablja istih primerov za napovedi in učenje. Razvijte še mero napovedne točnosti kot funkcijo `CA(real, predictions)`. Funkciji naj se uporabljata takole:

```
learner = LogRegLearner(lambda_=0.)
res = test_cv(le, X, y)
print "Tocnost:", CA(y, res) #argumenta sta pravi razredi, napovedani
```

V poročilo dodajte tabelo ki za širok nabor vrednosti lambda (npr. za 10 različnih vrednosti lambda) poroča o točnosti, kjer točnost merite:

- s 5-kratnim prečnim preverjanjem in (funkcija `test_cv`) in

- z gradnjo modela in napovedovanjem istih (vseh) primerov, torej učenje na učnih podatkih (funkcija `test_learning`).

**POMEMBNO**: Pri razvoju metod vam bodo pomagali testi.

**Dodatno (+20 točk)**: Uspešnost klasifikacije merite še s površino pod krivuljo ROC. Implementirajte jo kot funkcijo AUC(real, predictions). Implementirajte jo sami, brez uporabe že izdelanih ROC ali AUC.

**Oddaja**: Oddajte .zip arhiv, ki vsebuje:

1. Datoteko solution.py, kjer ste implementirali manjkajoče funkcije pri logistični regresiji in testiranju. Datoteka mora delovati s testi.
2. Datoteko rezultati.py, kjer ste implementirali izpise in izrise za 2. in 3. podnalogi. V datoteko ne kopirajte kode iz solution.py, ampak izdelane funkcije le uvozite.
3. Poročilo ki vsebuje samo 3 slike iz drugega dela in tabelo iz tretjega dela.

Poročilo naj bo napisano s predpisano predlogo. Vključite zgolj naslednja razdelka:

1. **Regularizacija.** Vključite izrise za tri zanimive stopnje regularizacije kot ga zahteva 2. podnaloga.
2. **Točnosti.** V eni tabeli poročajte o točnosti za različne stopnje regularizacije za obe metodi testiranja (v sklopu 3. podnaloge).

Dolžina poročila naj ne presega ene strani.

Pri domači nalogi boste potrebovali knjižnjice numpy, scipy in matplotlib.

## Status oddaje naloge

| | |
|---|---|
| Status oddaje naloge | Neoddano |
| Stanje ocen | Neocenjeno |
| Rok za oddajo | petek, 7. december 2018, 23:55 |
| Preostali čas | 13 dni 6 ure |
| Zadnja sprememba | - |
| Komentar oddaje | Komentarji (0) |

Oddaj nalogo

Niste še oddali naloge

◄ 3. domača naloga: napovedovanje prihodov a Skok na... Anketa o predmetu Uvod v odkrivanje znanj iz podatkov ►