Pēteris Račinskis
pr20015

Mājas darbs 4: PCA, kNN

Uzdevums 1

a. PCA - kvadrātiskas sakarības

Vispirms dati (faili pielikumā) tika noģenerēti ideāli pēc dotā vienādojuma (bez trokšņa) un WEKA pēdējo īpašvektoru neatrada vispār, izvadā uzrādot tikai 3:

```
eigenvalue      proportion      cumulative
 76.49037         0.95613         0.95613       0.914z+0.384xsquared+0.118x+0.049y
  3.40778         0.0426          0.99873       -0.721xsquared+0.581y+0.301z-0.228x
  0.10185         0.00127         1             0.967x-0.217xsquared+0.131y-0.04z

Eigenvectors
 V1      V2      V3
 0.1178 -0.228   0.9665 x
 0.3842 -0.7209 -0.2168 xsquared
 0.0487  0.581   0.1311 y
 0.9144  0.3013 -0.0404 z
```

Nejauši pieliekot dažiem *z* kādu ciparu aiz komata kopā parādās troksnis un WEKA atrod visus īpašvektorus:

```
eigenvalue      proportion      cumulative
 77.06421         0.95673         0.95673       0.915z+0.383xsquared+0.117x+0.048y
  3.3837          0.04201         0.99873       -0.72xsquared+0.584y+0.3  z-0.227x
  0.10166         0.00126         1             0.967x-0.22xsquared+0.126y-0.039z
  0.00034         0               1             0.801y+0.536xsquared-0.267z+0.007x

Eigenvectors
 V1      V2      V3      V4
 0.1175 -0.2274  0.9667  0.0067 x
 0.3831 -0.7197 -0.2195  0.5359 xsquared
 0.0482  0.5836  0.1258  0.8008 y
 0.9149  0.2998 -0.0388 -0.2674 z
```

Algebriski pārveidojot pēdējo īpašvektoru (un ignorējot niecīgo x komponenti) atklājas sākotnējā sakarība:

$$\bar{x^2} = 3.5; \bar{y} = 1.5; \bar{z} = -3.5$$

$$0.536(x^2 - 3.5) + 0.801(y - 1.5) - 0.267(z + 3.5) \approx 0 \Rightarrow$$

$$\Rightarrow 0.536x^2 + 0.801y - 4.012 = 0.267z \Rightarrow$$

$$\Rightarrow 2.007x^2 + 3.00y - 15.026 = z$$

Pēteris Račinskis
pr20015

Uzdevums 2

a. kNN parametri WEKA programmatūras pakotnē

Parametri, kuri ietekmē rezultātu:

- knn: tuvāko kaimiņu skaits, kuru klases tiek ņemtas vērā klasificējot punktu;
- distance weighting: sver kaimiņus pēc to distances metrikas (divas iespējas - 1-distance un 1/distance);
- window size: ņem vērā tikai pēdējos x datu punktus, veicot klasifikāciju (paredzēts darbam ar datu plūsmām, nav īpaši aktuāls šajā gadījumā un samazina rezultātu precizitāti pie maziem logiem);
- search algorithm: izmantotais algoritms, ietekmē izpildes ātrumu un izmantotās atmiņas apjomu, taču (vismaz dotajām datu kopām) rezultāta precizitāti neietekmē.

b. kNN - Iris

Šai datu kopai rezultāti ir samērā labi. Jau ar sākotnējiem parametriem, veicot krosvalidāciju, redzams, ka precizitāte ir abtuveni 95%:

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         143               95.3333 %
Incorrectly Classified Instances         7                4.6667 %
```

To var pacelt līdz 96%, palielinot knn parametru no 1 līdz 10:

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 10 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
```

Precizitātes vērtības atkal sāk kristies ap knn=30. Pārējos parametrus mainot novērots, ka distance weighting nedaudz samazina precizitāti pie nelieliem knn, taču palielina to pie lieliem:

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 100 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
```

c. kNN - ionosphere

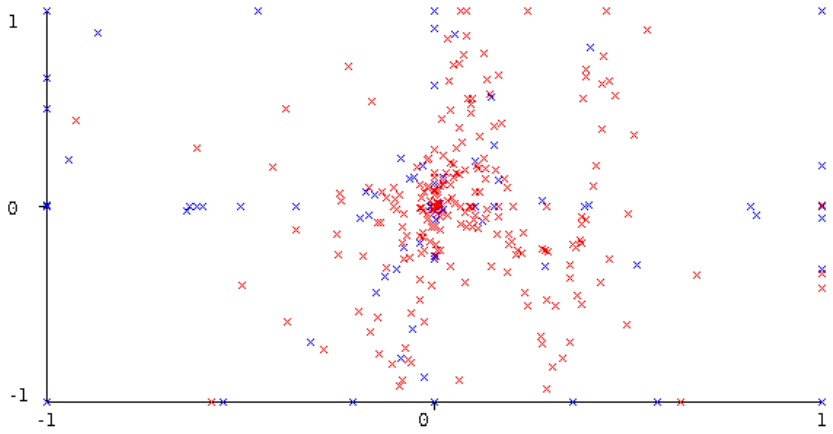
Šeit iegūtie rezultāti jau ir ievērojami sliktāki - ap 86% pie nelieliem knn, ar maksimālo precizitāti ap 88% pie knn=2:

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 2 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         311               88.604  %
Incorrectly Classified Instances        40               11.396  %
```

Lielākiem knn īpaši nepalīdz arī distance weighting - labākajā gadījumā 1-2% pie ~70% precizitātes. Citu parametru maiņa uzlabojumus nesniedz.

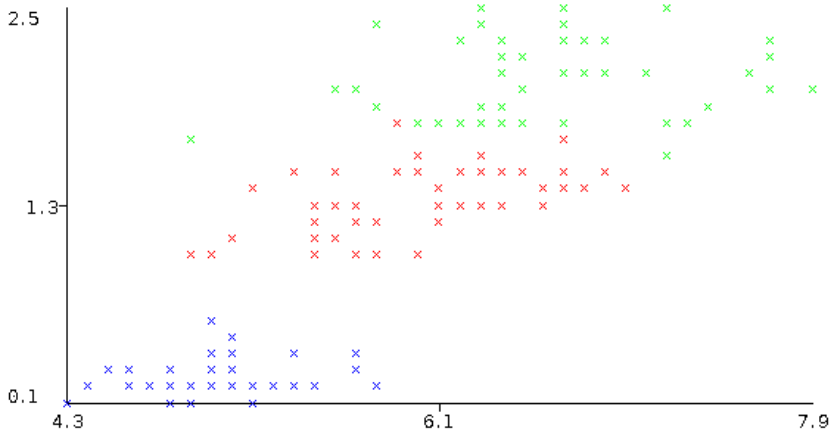
d. Secinājumi

Intuitīvs izskaidrojums atšķirībām klasifikācijas precizitātē varētu būt fakts, ka dati *ionosphere* ir ļoti "sajaukti" - jau vizuāli apskatot datu kopas projekciju jekburā nejauši izvēlētā plaknē redzams, ka dispersija dominē pār starpklašu atšķirību:



Šī situācija nozīmē, ka daudzos gadījumos katra punkta kaimiņi var ar diezgan augstu varbūtību piederēt katrai no klasēm.

Savukārt *iris* datos klases veido samērā monotonus blokus - t.i., dominē distance starp klašu vidējām vērtībām, nevis katras klases iekšējā dispersija:



Dati bez trokšņa;

@relation quadratic

@attribute x numeric
@attribute xsquared numeric
@attribute y numeric
@attribute z numeric
@attribute rownum numeric

@data
0,0,0,-15,0
0,0,1,-12,1
0,0,2,-9,2
0,0,3,-6,3
1,1,0,-13,4
1,1,1,-10,5
1,1,2,-7,6
1,1,3,-4,7
2,4,0,-7,8
2,4,1,-4,9
2,4,2,-1,10
2,4,3,2,11
3,9,0,3,12
3,9,1,6,13
3,9,2,9,14
3,9,3,12,15

Pēteris Račinskis
pr20015

Dati ar troksni:

@relation quadratic

@attribute x numeric
@attribute xsquared numeric
@attribute y numeric
@attribute z numeric
@attribute rownum numeric

@data
0,0,0,-15.1,0
0,0,1,-12,1
0,0,2,-9.2,2
0,0,3,-6,3
1,1,0,-13,4
1,1,1,-10,5
1,1,2,-7.05,6
1,1,3,-4,7
2,4,0,-7,8
2,4,1,-4,9
2,4,2,-1.1,10
2,4,3,2,11
3,9,0,3,12
3,9,1,6.2,13
3,9,2,9,14
3,9,3,12,15