Pēteris Račinskis
pr20015

Mājas darbs 6: Klasterizācija

Uzdevums 1

a. glass - hierarhiskā klasterizācija

Kopumā hierarhiskā klasterizācija šai datu kopai ir izteikti neveiksmīga. Sākot ar izejas parametriem un nomainot tikai paredzēto klasteru skaitu uz 6 (īstais klašu skaits, kas hierarhiskās klasterizācijas gadījumā ir vienkārši viens slānis konstruētajā kokā) un ļaujot programmatūrai piešķirt (vai nepiešķirt) katram klasterim kādu no klasēm, kļūdas procents ir 63.09%. Salīdzinājumam, ja visa datu kopa tiktu uztverta kā viens klasteris ar biežāko klasi, kļūdas procents būtu $\frac{n_{total} - n_{maxclass}}{n_{total}} = 64.49\%$ - kas liek domāt, ka iegūtais rezultāts ir aptuveni ekvivalents šai darbībai, t.i., lielākais klasteris ietver gandrīz visus datus, ticis apzīmēts ar biežāko klasi un daži izlēcēji ir saņēmuši savai klasei atbilstošu apzīmējumu. Nedaudz papētot rezultātu izdruku, redzams, ka precīzi tā arī ir noticis:

```
Clustered Instances

0      208 ( 97%)
1        1 (  0%)
2        1 (  0%)
3        1 (  0%)
4        1 (  0%)
5        2 (  1%)


Class attribute: Type
Classes to Clusters:

  0  1  2  3  4  5  <-- assigned to cluster
 70  0  0  0  0  0 | build wind float
 75  0  0  1  0  0 | build wind non-float
 17  0  0  0  0  0 | vehic wind float
  0  0  0  0  0  0 | vehic wind non-float
 10  1  0  0  0  2 | containers
  8  0  1  0  0  0 | tableware
 28  0  0  0  1  0 | headlamps

Cluster 0 <-- build wind non-float
Cluster 1 <-- No class
Cluster 2 <-- tableware
Cluster 3 <-- No class
Cluster 4 <-- headlamps
Cluster 5 <-- containers

Incorrectly clustered instances :       135.0    63.0841 %
```

Redzams, ka 97% datu pieder vienam klasterim. Izmēģinot citas starp-klasteru attāluma noteikšanas metodes (ar to pašu Eiklīda distances metriku), labāko rezultātu var atrast pie “Complete” varianta:

```
Clustered Instances

0      137 ( 64%)
1        3 (  1%)
2       29 ( 14%)
3       27 ( 13%)
4       17 (  8%)
5        1 (  0%)


Class attribute: Type
Classes to Clusters:

  0  1  2  3  4  5  <-- assigned to cluster
 61  0  9  0  0  0 | build wind float
 52  3 15  0  6  0 | build wind non-float
 14  0  3  0  0  0 | vehic wind float
  0  0  0  0  0  0 | vehic wind non-float
  1  0  2  3  7  0 | containers
  5  0  0  0  3  1 | tableware
  4  0  0 24  1  0 | headlamps

Cluster 0 <-- build wind float
Cluster 1 <-- No class
Cluster 2 <-- build wind non-float
Cluster 3 <-- headlamps
Cluster 4 <-- containers
Cluster 5 <-- tableware

Incorrectly clustered instances :       106.0    49.5327 %
```

Redzams, ka vēl joprojām dominē viens klasteris, taču vismaz “headlamps” klase ir diezgan precīzi atdalīta no pārējiem datiem, papildus dažiem šķietami nejaušiem izlēcēju klasteriem. Tālāk pamainot distances metrikas parametru “dontNormalize” uz “True” (lai tiktu izmantotas asu vērtības tiešā veidā, bez normalizēšanas uz vienu mērogu) precizitāte nedaudz krītas:

```
Clustered Instances

0      162 ( 76%)
1       24 ( 11%)
2       11 (  5%)
3       12 (  6%)
4        3 (  1%)
5        2 (  1%)


Class attribute: Type
Classes to Clusters:

  0  1  2  3  4  5  <-- assigned to cluster
 70  0  0  0  0  0 | build wind float
 65  0  7  4  0  0 | build wind non-float
 17  0  0  0  0  0 | vehic wind float
  0  0  0  0  0  0 | vehic wind non-float
  2  0  4  4  1  2 | containers
  4  2  0  3  0  0 | tableware
  4 22  0  1  2  0 | headlamps

Cluster 0 <-- build wind float
Cluster 1 <-- headlamps
Cluster 2 <-- build wind non-float
Cluster 3 <-- tableware
Cluster 4 <-- No class
Cluster 5 <-- containers

Incorrectly clustered instances :       110.0    51.4019 %
```

Mainot klasteru skaitu, vispirms var izpētīt, kas notiek ieviešot tikai 2. Rezultāts visnotaļ paredzams ir tas pats kas no sākuma - precizitāte ir vienkārši biežākās klases īpatsvars:

```
Clustered Instances

0      203 ( 95%)
1       11 (  5%)


Class attribute: Type
Classes to Clusters:

  0  1  <-- assigned to cluster
 70  0 | build wind float
 69  7 | build wind non-float
 17  0 | vehic wind float
  0  0 | vehic wind non-float
  9  4 | containers
  9  0 | tableware
 29  0 | headlamps

Cluster 0 <-- build wind float
Cluster 1 <-- build wind non-float

Incorrectly clustered instances :       137.0    64.0187 %
```

Reģionā 4-8 novērojama precizitāte tuva labākajai iepriekš novērotajai ar nelielu kritumu (50+% kļūda), taču interesantu parādību var novērot pie 9-10 klasteriem - starp mazākajiem klasteriem vēl joprojām dominē viens lielākais, taču tagad parādās daži lokāli apgabali, kur precizitāte atkal palielinās, un rezultāts nav sliktāks, kā pie 6 klasēm:

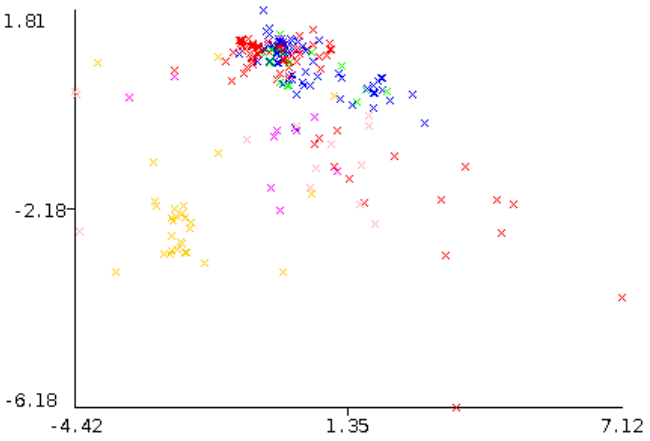
```
Clustered Instances

0      139 ( 65%)
1       23 ( 11%)
2        2 (  1%)
3       23 ( 11%)
4       12 (  6%)
5        9 (  4%)
6        1 (  0%)
7        3 (  1%)
8        2 (  1%)


Class attribute: Type
Classes to Clusters:

  0  1  2  3  4  5  6  7  8  <-- assigned to cluster
 54  0  0 16  0  0  0  0  0 | build wind float
 63  0  2  2  4  5  0  0  0 | build wind non-float
 14  0  0  3  0  0  0  0  0 | vehic wind float
  0  0  0  0  0  0  0  0  0 | vehic wind non-float
  2  0  0  0  4  4  0  1  2 | containers
  4  1  0  0  3  0  1  0  0 | tableware
  2 22  0  2  1  0  0  2  0 | headlamps

Cluster 0 <-- build wind non-float
Cluster 1 <-- headlamps
Cluster 2 <-- No class
Cluster 3 <-- build wind float
Cluster 4 <-- tableware
Cluster 5 <-- containers
Cluster 6 <-- No class
Cluster 7 <-- No class
Cluster 8 <-- No class

Incorrectly clustered instances :       106.0    49.5327 %
```

Izvēloties citas distances metrikas:

- Čebiševa dstiance - maksimālā no starpībām gar kādu no vektoru asīm - ir mazliet (52%) sliktāka;
- Manhetenas distance - starpību summa pa vektoru dimensijām - ir ievērojami sliktāka (58%)
- Minkovska distance ģeneralizē visas šīs distances vienā izteiksmē - Eiklīda distance ir pakāpe 2, Mahetenas - pakāpe 1 - bet Čebiševa - pakāpe +inf. Mainot pakāpes vienīgas rezultāts, kas ir nedaudz labāks

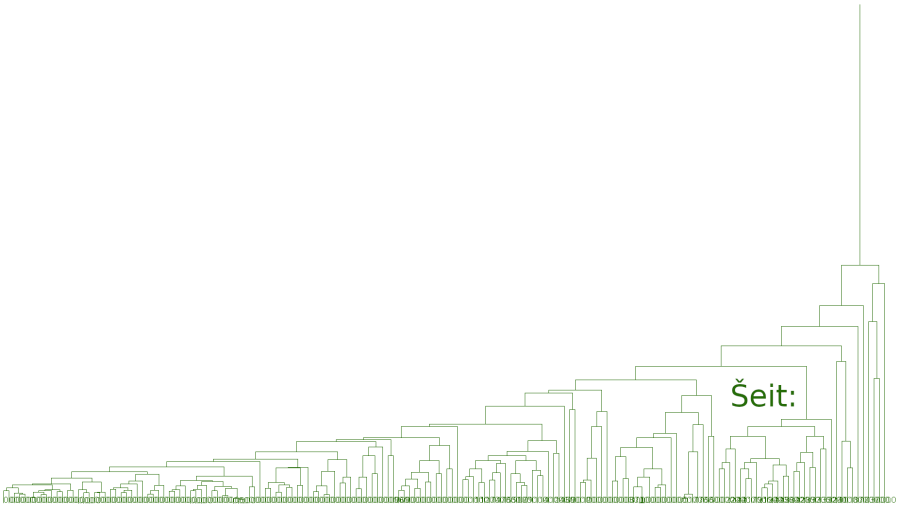
par 2, ir 1.5, taču par ļoti nelielu starpību (<0.5%) un grūti spriest, ko tas nozīmē - pakāpēm 1.25 un 1.75 rezultāti ir sliktātki.

Protams, kopējais rezultāts vēl joprojām ir diezgan slikts. Lai mēģinātu inuitīvi saprast, kāpēc tā, var mēginat vizualizēt datu kopu ar dimensiju redukcijas metodēm. Izvēloties PCA, pirmie divi īpašvektori ietver tikai 50% kopējas dispersijas, taču tajos redzamā aina ir līdzīga arī citās projekcijās un sniedz labu priekštatu par neveiksmes iemesliem:



Redzams, ka (vismaz šajā projekcijā) liela daļa datu atrodami vienā sajauktā klasterī ar salīdzinoši lielu skaitu šķietami nejauši izkliedētu izlēcēju. Vizuāli uzkrītošs ir tikai viens papildus klasteris, kas ir arī visnotaļ homogēns, un to klasterizācijas metodes arī sekmīgi atrod. Tātad, ja uzdevums būtu klasificēt tikai starp klasi “headlamps” un pārējām, iegūtais rezultāts būtu adekvāts, bet citādi - nē.

Vizualizējot rezultātu dendrogrammu grūti redzēt kādas uzkrītošas sakarības. Atsevišķais “headlamps” klasteris, iespējams, ir atzīmētais vai pa labi no tā, taču saprast, kuri zari attiecas uz kuriem datiem ir atjautības uzdevums - marķējumi nepalīdz un uz linux sistēmas pat sākotnējais attēls nav saskatāms, jo zīmēts gandrīz baltā krāsā un jāapstrādā pirms vispār kaut ko var saskatīt:



Uzdevums 2

a. glass - klasterizācija ar k-means

Izmantojot k-means, pirmā sakarība, ko var pamanīt, ir tas, ka Manhattan distance pie pareizā klasteru skaita (6) dod labāku preciziāti (kļūda 48.60%) salīdzinot ar Eiklīda distanci (57.01%). Taču pamainot "seed" parametru (sākotnējo koordināšu nejaušā ģeneratora ieejas parametru), ātri vien atklājas, ka pastāv ievērojama variācija rezultātos - Eiklīda gadījumā tie visai stabili turas starp 54% un 60% ar distanču summu (optimizējamo izteiksmi) 18-20, savukārt Manhetenai starp 45% un 60% ar distanču summu 103-123. Grūti spriest, vai ar citiem izejas parametriem to precizitāte pārklājas, taču intuitīvi varētu spriest, ka Manhetenas distance ir jutīgāka pret atšķirīgiem sākuma nosacījumiem.

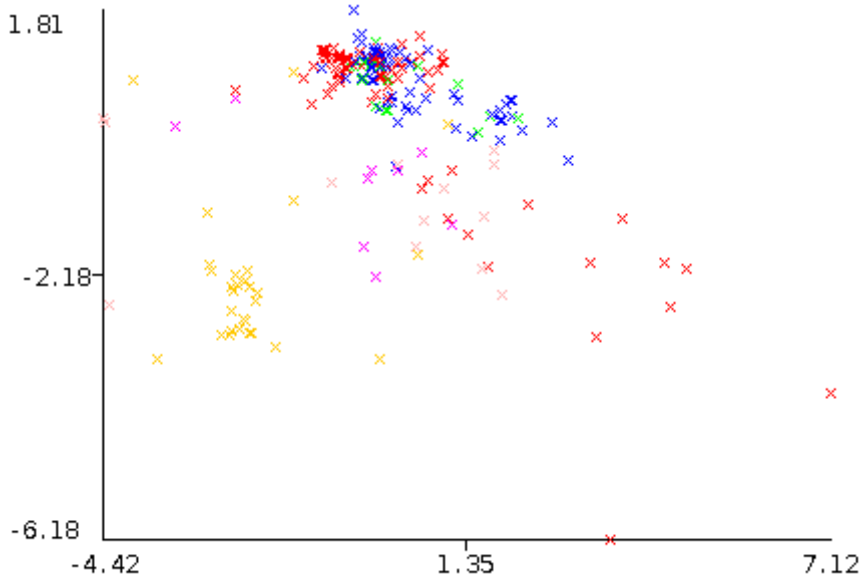
Vērtējot "elkoņa likumu" Eiklīda distancei ar fiksētu "seed" parametru, sākot ar 2 klasteriem, novērota virkne aptuveni 34-29-26-21-19-18-18-.., kas pareizo klasteru skaitu tik tiešām nosprauž ap 6, taču rezultātat precizitāte tāpat ir slikta, un apsverot iepriekšējā uzdevumā viekto datu analīzi rodas šaubas, vai tas tiešām ir saistīts ar datu klašu skaitu.

```
Clustered Instances

0       97 ( 45%)
1       29 ( 14%)
2       34 ( 16%)
3       24 ( 11%)
4       19 (  9%)
5       11 (  5%)


Class attribute: Type
Classes to Clusters:

  0  1  2  3  4  5  <-- assigned to cluster
 40  0 11 19  0  0 | build wind float
 43  0 19  2 11  1 | build wind non-float
 12  0  3  2  0  0 | vehic wind float
  0  0  0  0  0  0 | vehic wind non-float
  0  3  1  0  7  2 | containers
  2  3  0  0  1  3 | tableware
  0 23  0  1  0  5 | headlamps

Cluster 0 <-- build wind non-float
Cluster 1 <-- headlamps
Cluster 2 <-- vehic wind float
Cluster 3 <-- build wind float
Cluster 4 <-- containers
Cluster 5 <-- tableware

Incorrectly clustered instances :       116.0    54.2056 %
```

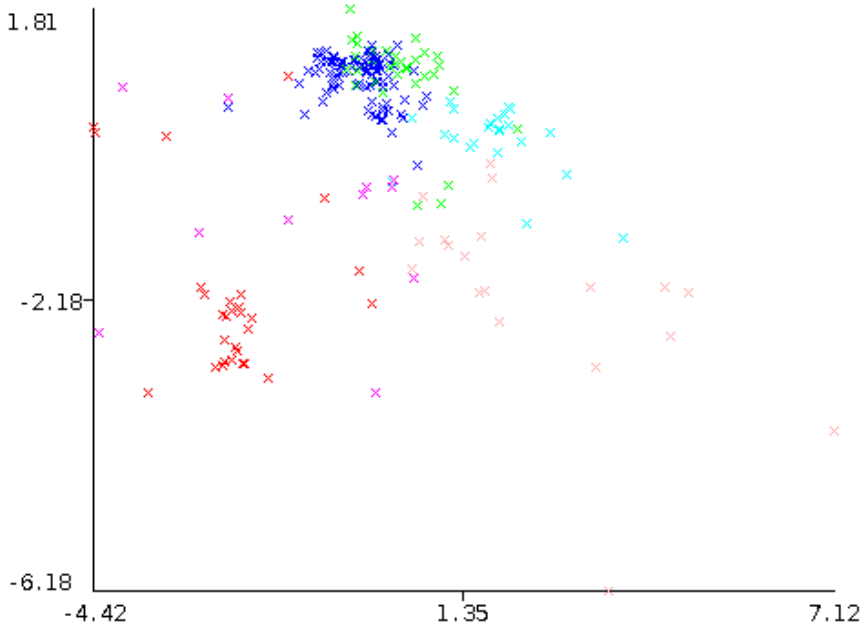
Redzams, ka atrasti tie paši divi galvenie klasteri un atkal sagrupētas izlēcēju kopas.

Iznesot datus .arff formātā, izdzēšot visas ar @ iesāktās un tukšās rindas iegūst "comma separated values" failu. Slinkākajā variantā to tad var pārsaukt par <nosaukums>.csv, un MS excel vai kāda ekvivalenta programmatūras pakotne to jau spēj nolasīt. Šādā formātā par datiem ar daudzām skaitliskām dimensijām un bez nekādas izpratnes par to nozīmi gan grūti kaut ko spriest. Autoram vienkāršāk un

interesantāk likās saglabātos datus vēlreiz ielādēt WEKA rīkā, atmest klases un aizstāt tās ar klasteru kolonnu, veikt dimensiju redukciju un paskatīties, kā klasteri atšķiras no klasēm. Vispirms var blakus nolikt to pašu projekciju un salīdzināt:

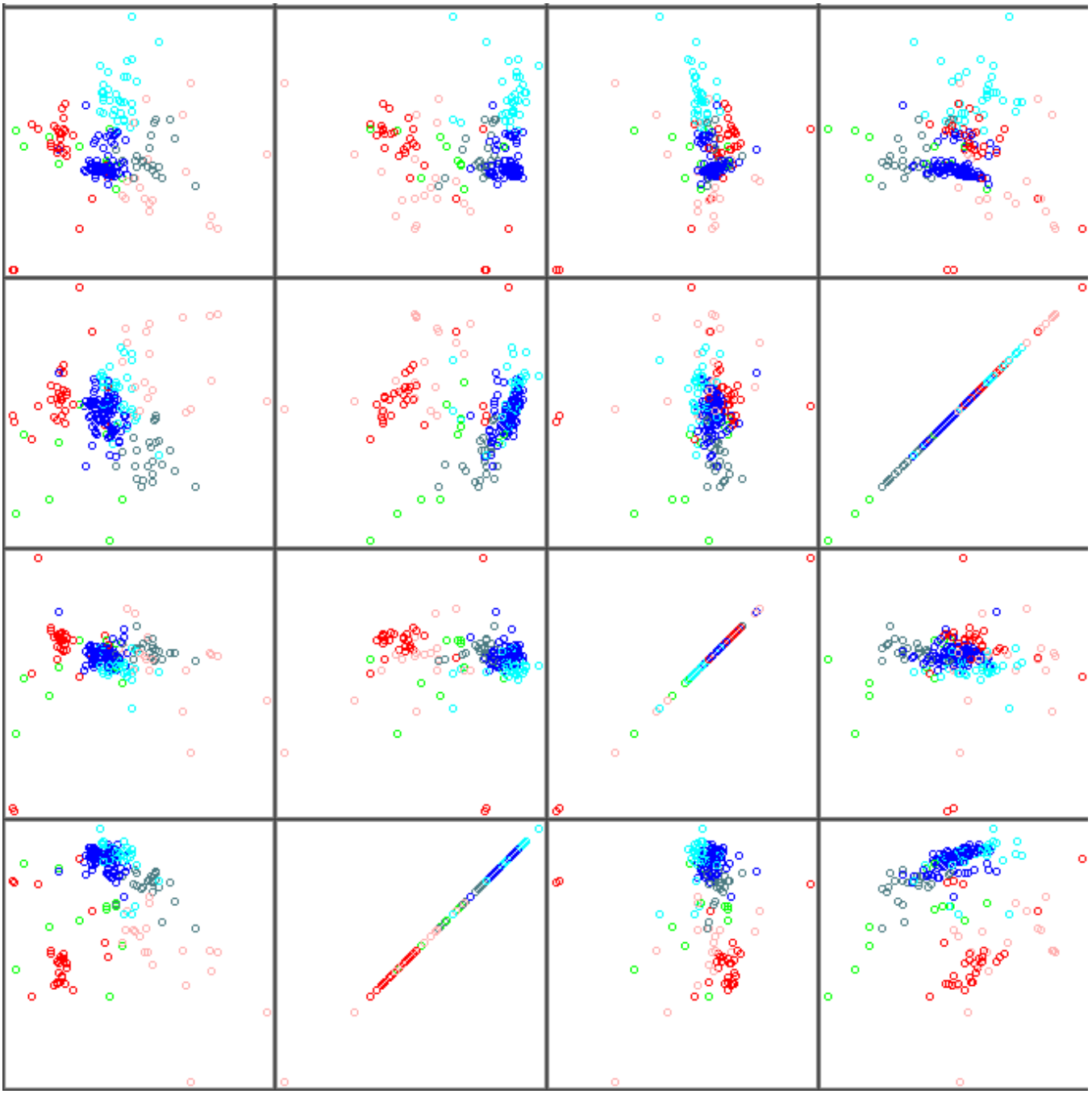


klases

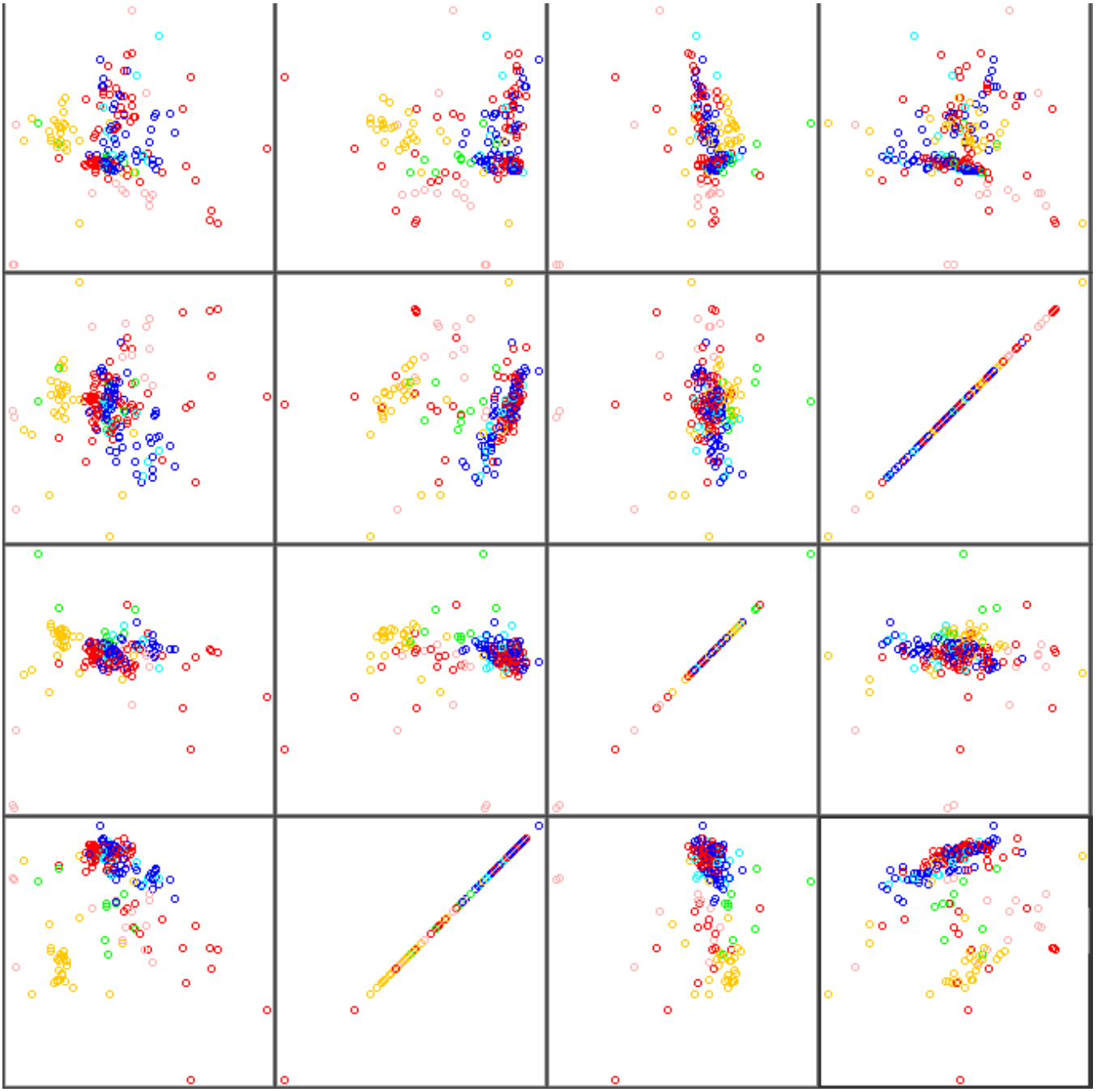


klasteri

Skaidri redzams, ka vismaz “headlamps” (dzeltens, sarkans klasteris) klase ir skaidri nošķirta arī kā klasteris, taču “galvenajā” klasterī, kurā hierarhiskā metodei bija tendence grupēt gandrīz visus datus, ieviesti klasteri, kuriem īpašas nozīmes nav. Izlēcēji veido atsevišķus klasterus. Lai gūtu pilnīgāku izpratni, iespējams apskatīt vēl citas projekcijas:



klasteri



klases

(visām nepietiek vietas). Skatoties uz 10 “stiprākajām” projekcijām (viss zem diagonāles ir duplikāts), var redzēt, ka visās ir novērojams viens klasteris (zilais), no kura “nošķelti” citi, ko var manīt kādās no projekcijām - taču sākotnējos datos klašu izkārtojums ar šo struktūru nesakrīt - galvenajā klasterī ir sajauktas divas lielākās klases (kas kopā sastāda ievērojami vairāk nekā pusi datu kopas). Turklāt vienai no tām - “build wind non-float” (sarkans) ir liela dispersija un/va liels skaits izlēcēju, atrodami starp potenciāliem mazāko klašu klasteriem. Par to, ka mazākās klases klasterus potenciāli varētu veidot, var pārliecināties, salīdzinot cluster4/”containers”

(sk. izvades datus 5.lpp,), kas ir vienā un tajā pašā rozā krāsā abos attēlos. Klasterī ir 7 no 10 “containers” punktiem, taču precizitāti grauj fakts, ka ieskaitīti arī 11 attālāki punkti no “build wind non-float” (sarkans, viena no dominējošajām klasēm datu kopā). Pastāv iespēja, ka, ja datu kopas būtu vienādā izmērā, atrastais klasteris labāk atspoguļotu datu dalījumu klasēs. Tomēr iespējams arī, ka, ja “build wind non-float” ir liela dispersija un tā nosedz pārējās klases, pārklāšanās dēļ klasteri nekad labi neatbilstu klasēm.