SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6493

OTKRIVANJE STRŠEĆIH VRIJEDNOSTI U PODATCIMA UPORABOM PLATFORME ELKI

Martina Sušilović

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6493

OTKRIVANJE STRŠEĆIH VRIJEDNOSTI U PODATCIMA UPORABOM PLATFORME ELKI

Martina Sušilović

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 13. ožujka 2020.

ZAVRŠNI ZADATAK br. 6493

Pristupnica:	**Martina Sušilović (0036507308)**
Studij:	Računarstvo
Modul:	Računarska znanost
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Otkrivanje stršećih vrijednosti u podatcima uporabom platforme ELKI**

Opis zadatka:

Stršeći podatci (engl. outliers) su podatci koji po svojim značajkama odstupaju od većine podataka u skupu. Otkrivanje stršećih podataka je važno zbog više razloga: uklanjanja šuma u podatcima, ustanovljavanja podgrupa u podatcima i pronalaska rijetkih uzoraka. Za pronalazak stršećih podataka mogu se koristiti tehnike nadziranog i nenadziranog učenja. U ovom završnom radu potrebno je proučiti i opisati platformu ELKI za dubinsku analizu podataka. Platforma ELKI ima implementiranih više postupaka za otkrivanje stršećih podataka. Za potrebe završnog rada potrebno je na nekoliko slobodno dostupnih skupova podataka (npr. iz repozitorija UCI Irvine Machine Learning) provesti analizu stršećih podataka koristeći platformu ELKI te kvalitatitvno i kvantitativno usporediti rezultate dobivene različitim implementiranim postupcima.

Rok za predaju rada: 12. lipnja 2020.

Sadržaj

Uvod

U dubinskoj analizi podataka, stršeće vrijednosti (engl. *outliers*) su zapažanja koja značajno odstupaju od ostalih vrijednosti u skupu podataka. Alternativno, to su vrijednosti koje su nekonzistentne u odnosu na skup vrijednosti kojem pripadaju. Uzrok pojavi stršećih vrijednosti mogu biti pogreške, primjerice pri mjerenju, zaokruživanju vrijednosti ili pri unosu. Takvi su podaci u skupu nepoželjni. Uzrok može biti i ispravno, ali rijetko tj. iznimno ponašanje koje ne predstavlja pogrešku nego predmet posebnog interesa. Prisutnost stršećih vrijednosti može nepovoljno utjecati na interpretaciju podataka, klasifikaciju, modele stvorene na temelju podataka ili donošenje odluka.

Detekcija stršećih vrijednosti podrazumijeva identifikaciju rijetkih i abnormalnih vrijednosti te je često primarni korak u modeliranju i analizi podataka. Postupak detekcije važan je u područjima strojnog učenja, dubinske analize podataka i prepoznavanja uzoraka. Primjenjuje se u otkrivanju zloupotrebe platnih kartica, u medicinskoj dijagnostici, u sigurnosnim sustavima, pri „čišćenju“ podataka (podrazumijeva ispravljanje ili uklanjanje neispravnih podataka), kod otkrivanja upada u mrežama i drugim domenama primjene dubinske analize.

Tri su kategorije tehnika otkrivanja stršećih vrijednosti. Nadzirano učenje anomalija pronalazi stršeće vrijednosti u skupu podataka u kojem su svi primjerci označeni kao normalni ili abnormalni. Nenadzirano učenje podrazumijeva neoznačen skup podataka i funkcionira pod pretpostavkom da je većina podataka ispravna te pronalazi primjerke koje najmanje odgovaraju ostatku skupa. Treća su opcija tehnike polunadziranog učenja koje stvaraju model normalnog ponašanja iz danog normalnog skupa za učenje, a zatim ispituju vjerojatnost da primjerci podliježu stvorenom modelu.

Postoji više različitih pristupa otkrivanju stršećih vrijednosti koji se načelno mogu klasificirati u 5 kategorija: algoritmi temeljeni na raspodjeli, udaljenosti, dubini, gustoći i grupiranju. Platforma ELKI nudi, između ostalog, implementaciju brojnih algoritama za detekciju stršećih vrijednosti kroz Java API ili grafičko sučelje MiniGUI koji omogućuju vizualizaciju skupa podataka i dobivenih rezultata izvođenja algoritama.

1. Platforma ELKI

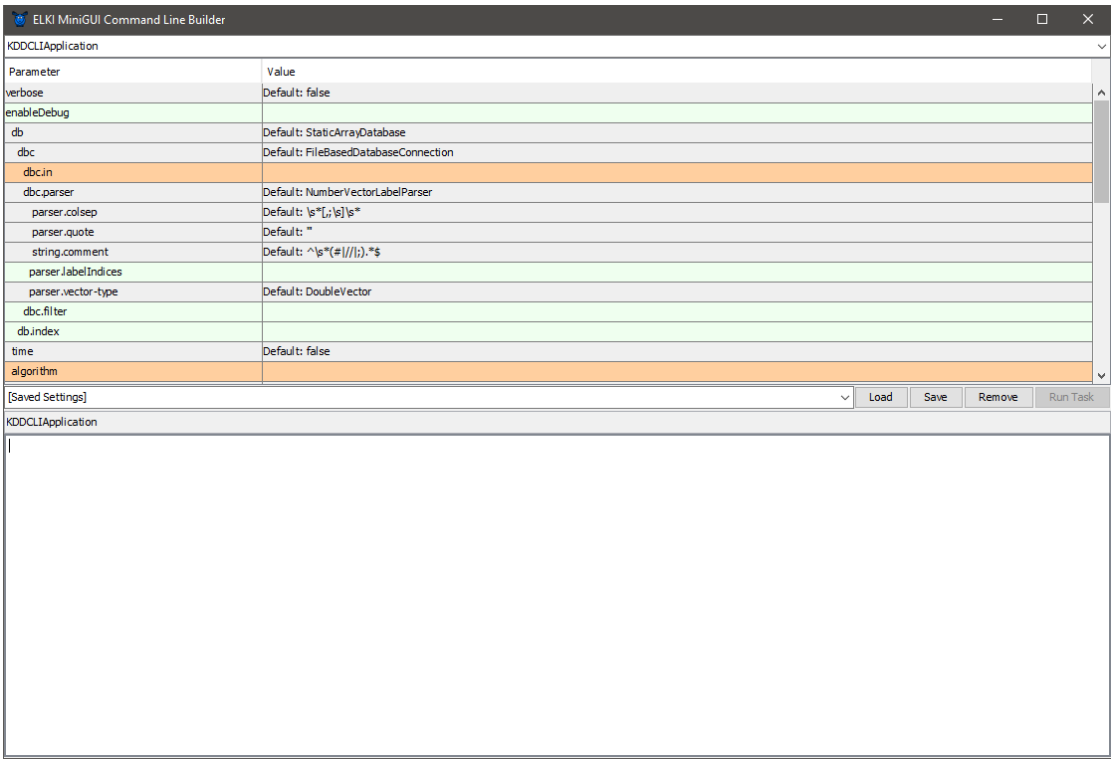
ELKI (engl. *Environment for Developing KDD-Applications Supported by Index-Structures*) je radni okvir otvorenog koda za dubinsku analizu podataka razvijen u svrhe istraživanja i poučavanja. Dostupno je grafičko i sučelje naredbenog retka, kao i standardno Javino aplikacijsko programsko sučelje. Razvoj platforme započeo je na Sveučilištu u Münchenu, a nastavljen na Tehničkom sveučilištu u Dortmundu. Platforma je usredotočena na istraživanje algoritama s naglaskom na nenadzirane metode za grupiranje podataka (engl. *clustering*) te otkrivanje stršećih vrijednosti (engl. *outlier detection*). ELKI je napisan u programskom jeziku Java i dizajniran na način koji omogućuje jednostavnu nadogradnju. Modeliran je oko jezgre inspirirane bazom podataka uz strukture indeksa koji poboljšavaju brzinu izvođenja. Korištenje Javinog sučelja omogućuje korisnička proširenja radnog okvira implementacijom vlastitih tipova podataka, funkcija za računanje udaljenosti, algoritama, parsera ulaznih podataka ili formata rezultata.

1.1. ELKI-jev MiniGUI

ELKI je moguće koristiti kroz jednostavno grafičko korisničko sučelje *MiniGUI* koje se pokreće pri pokretanju .jar datoteke preuzete s [1], i to naredbom `java -jar elki.jar`.

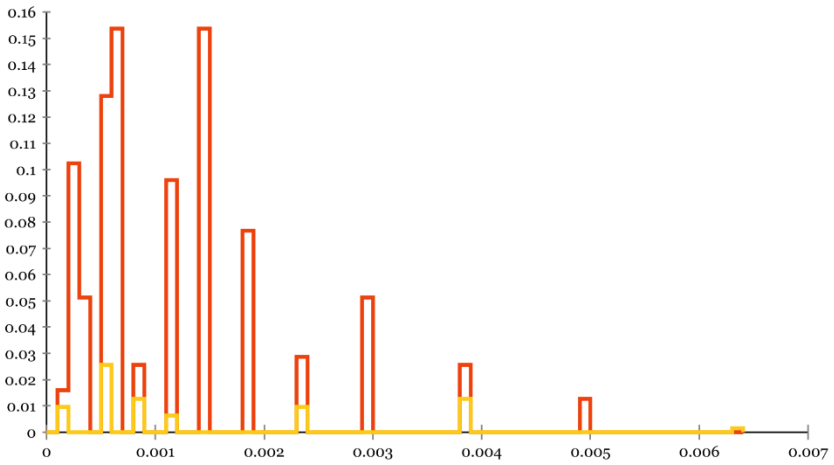
Izgled grafičkog sučelja prikazan je na slici 1.1. Na vrhu se nalazi tablica dostupnih parametara koja se dinamički mijenja u ovisnosti o trenutno odabranim parametrima. Donji dio zaslona služi za ispis pogrešaka u konfiguraciji.

Ulazna datoteka s podacima zadaje se parametrom *dbc.in*. Potrebno je odabrati željeni algoritam, ili više njih, i po potrebi postaviti obvezne tražene parametre. Postavljanjem parametra *resulthandler* na pretpostavljenu vrijednost „AutomaticVisualization“, kao rezultat pokretanja algoritma dobiva se niz grafičkih prikaza za ulazni skup podataka. Osim toga, moguće je dobiti tekstni zapis rezultata izvođenja algoritma postavljenjem tog parametra na „ResultWriter“ i zadavanjem putanje izlazne datoteke u polje označeno s *out*.



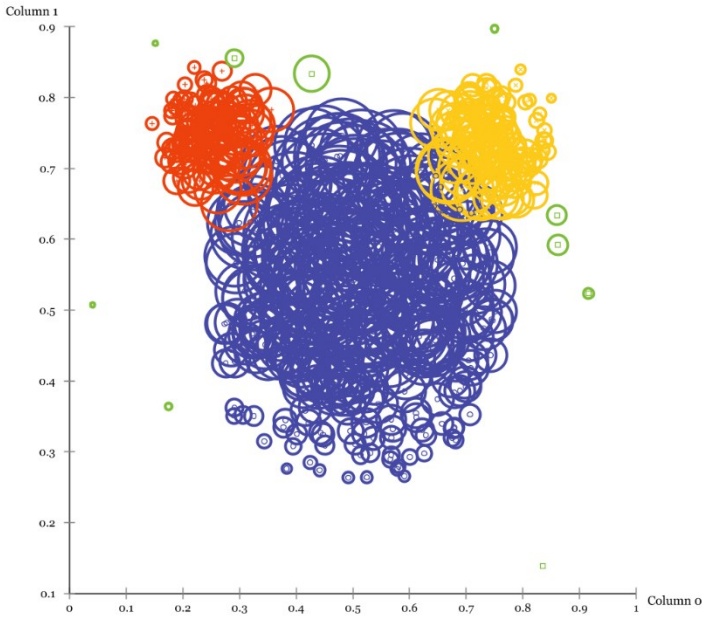
Slika 1.1 Izgled grafičkog korisničkog sučelja platforme ELKI

Za algoritme otkrivanja stršećih vrijednosti, između ostalog, prikazuju se stupčasti dijagram rezultata provođenja algoritama (engl. *outlier score histogram*). Na *x*-osi su vrijednosti iz intervala mogućih rezultata dobivenih za sve primjerke, a na *y*-osi postoci koji određuju udio skupa podataka za koje je dobiveni rezultat unutar pojedinog dijela intervala. Primjer izgleda takvog dijagrama dan je slikom 1.2. Crvenom su linijom označene vrijednosti za anomalije, a žutom za normalne primjerke.



Slika 1.2 Izgled histograma s rezultatima izvođenja algoritma

Dijagram raspršenosti (engl. *scatter plot*) je prikaz točaka koje odgovaraju primjercima skupa podataka na kojem je svakoj točki pridružen krug (engl. *outlier bubble*) polumjera koji je u linearnoj korelaciji s izlaznom vrijednošću dobivenom za tu točku. Izlazna vrijednost je rezultat koji određuje u kojoj je mjeri primjerak anomalija, a postupak dobivanja vrijednosti ovisi o odabranom pristupu. Primjer jednog takvog dvodimenzionalnog prikaza je na slici 1.3. Boje točaka i pripadnih krugova razlikuju se s obzirom na pripadnost klasi, tj. s obzirom na vrijednost zadnjeg stupca u ulaznoj datoteci.



Slika 1.3 Izgled 2D dijagrama raspršenosti

U slučaju podataka s više od dva stupca, prikazuje se jedan trodimenzionalni dijagram s vrijednostima prvih triju stupaca. Osim toga, za skupove s većim brojem atributa prikazuju se dvodimenzionalne projekcije ovisnosti raznih kombinacija parova atributa.

1.2. ELKI-jev Java API

Drugi je mogući pristup korištenju ELKI-ja putem standardnog Javinog aplikacijskog programskog sučelja. Osim korištenja već ostvarenih funkcionalnosti, prikladan je za proširivanje radnog okvira vlastitom implementacijom definiranih sučelja. ELKI je moguće dodati u Java projekt kao *.jar* datoteku preuzetu s [1] ili pomoću ovisnosti (engl. *dependency*) za Maven (programski alat koji omogućuje jednostavnije upravljanje projektima). Ovisnost za Maven dostupna je na [9] i unosi se u datoteku *pom.xml*.

Prije pokretanja bilo kojeg algoritma, potrebno je učitati podatke iz datoteke i stvoriti bazu podataka. Jedan od načina da se to postigne, dan je kodom 1.1.

```
ListParameterization params = new ListParameterization();
params.addParameter(FileBasedDatabaseConnection.Parameterizer.INPUT_ID, filename);
Database db = ClassGenericsUtil.parameterizeOrAbort(StaticArrayDatabase.class, params);
db.initialize();
```

Kôd 1.1 – Program za učitavanje baze podataka

Programsko ostvarenje algoritama za otkrivanje stršećih vrijednosti nalazi se u paketu `de.lmu.ifi.dbs.elki.algorithm.outlier`. Svi su algoritmi modelirani zajedničkim sučeljem `OutlierAlgorithm`.

2. Metode za otkrivanje stršećih vrijednosti

Jedna od mogućih podjela metoda za otkrivanje anomalija u skupovima podataka je na:

- statističke testove,
- metode temeljene na udaljenosti (engl. *distance-based*),
- metode temeljene na dubini (engl. *depth-based*),
- metode temeljene na odstupanju (engl. *deviation-based*),
- metode temeljene na gustoći (engl. *density-based*),
- metode prikladne za podatke velikih dimenzija (s mnogo atributa) [3].

Statistički testovi koriste neku određenu statističku raspodjelu i pretpostavku da vrijednosti zadanog skupa podataka podliježu toj raspodjeli. Kao stršeće vrijednosti uzimaju se one čija je vjerojatnost za zadanu raspodjelu najmanja.

Metode temeljene na udaljenosti evaluiraju točke prema udaljenosti do njima susjednih točaka. Ove su metode prikladne za skupove podataka s manjim brojem atributa, jer razlika udaljenosti postaje manja u podacima velikih dimenzija. Jednostavan primjer funkcije koja se koristi u ovom pristupu je euklidska udaljenost.

Metode temeljene na dubini traže stršeće vrijednosti na rubovima prostora definiranog vrijednostima atributa, neovisno o raspodjeli.

Metode temeljene na odstupanju u skupu pronalaze podatke koji po nekim karakteristikama ne odgovaraju skupu. Drugim riječima, pronalaze vrijednosti čijim bi se uklanjanjem smanjila varijanca skupa podataka.

Algoritmi temeljeni na gustoći uspoređuju gustoću oko točke s gustoćama susjednih točaka na lokalnoj razini. Kao mjera za određivanje stršećih vrijednosti koristi se relativna gustoća točke u odnosu na susjedne. Razni specifični pristupi koriste različite funkcije za gustoću.

Većina jednostavnih pristupa nisu prikladni na podatke s mnogo atributa. Raspored točaka u ovom slučaju postaje rjeđi, a razlika udaljenosti među točkama se smanjuje povećanjem dimenzionalnosti. Za ovakve slučajeve moguće je tražiti stršeće vrijednosti u odnosu na neku projekciju, tj. podskup atributa.

Metode za otkrivanje stršećih vrijednosti mogu kao rezultat dati binarnu podjelu skupa na normalne i stršeće vrijednosti. Alternativno, izlaz može biti niz sortiranih kontinuiranih vrijednosti dobivenih kao rezultat izračuna za svaki pojedini primjerak. Vrijednosti opisuju u kojoj je mjeri svaki pojedini primjerak skupa podataka anomalija. Rezultate dobivene različitim mehanizmima nije moguće međusobno uspoređivati.

S obzirom na referentni skup kod promatranja pojedinog objekta, pristupi otkrivanja anomalija dijele se na lokalne i globalne. Kod globalnih pristupa, referentni skup sadrži sve podatke, uključujući sve stršeće vrijednosti čija prisutnost može utjecati na rezultat. Pretpostavlja se postojanje samo jednog mehanizma za generiranje normalnih vrijednosti. Drugi je pristup lokalni, kod kojeg se za promatranu točku određuje referentni skup kao podskup cijelog skupa podataka. Glavni je problem kod ovog pristupa izbor lokalnog referentnog skupa.

U nastavku su opisani principi nekih od najčešće korištenih algoritama za detekciju anomalija i rezultat primjene njihove implementacije u sklopu platforme ELKI.

2.1. Gaussov model

2.1.1. Opis Gaussovog modela

Otkrivanje stršećih vrijednosti korištenjem Gaussove razdiobe je jednostavan statistički model koji koristi pretpostavku da su svi primjerci skupa podataka generirani normalnom razdiobom. Normalne se pojave nalaze u dijelu razdiobe s visokom vjerojatnošću, a stršeće su one vrijednosti za koje je mala vjerojatnost da su generirane tom razdiobom (npr. imaju odstupanje minimalno tri puta veće od standardne devijacije).

U skupu podataka nalazi se *m* primjeraka s vrijednostima *n* atributa. Računa se srednja vrijednost svakog od atributa izrazom:

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j{}^i \quad . \tag{1}$$

Varijanca svakog od svojstava računa se izrazom:

$$\sigma_j{}^2 = \frac{1}{m} \sum_{i=1}^{m} \left(x_j{}^i - \mu_j \right)^2 \quad . \tag{2}$$

Konačno, pripadna vjerojatnost svakog primjerka određuje se formulom (3) normalne razdiobe:

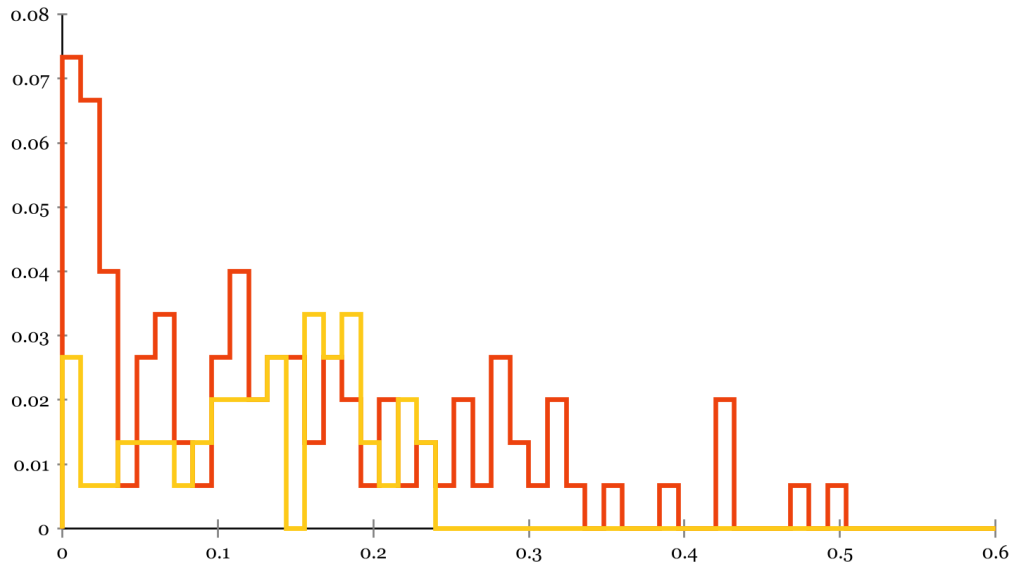
$$p(x) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\cdot\sigma_j} exp\left(-\frac{(x_j-\mu_j)^2}{2{\sigma_j}^2}\right) \quad . \tag{3}$$

Dobivena vjerojatnost koristi se u usporedbi u kojoj su mjeri primjerci podataka stršeći. Moguće je zadati parametar ε koji predstavlja prag za odjeljivanje normalnih od stršećih vrijednosti. Za primjerke za koje vrijedi $p(x) < \varepsilon$ tada vrijedi da su anomalije.

2.1.2. Primjer korištenja Gaussovog modela

U ELKI-ju je algoritam koji koristi jednu Gaussovu razdiobu za otkrivanje anomalija modeliran razredom `de.lmu.ifi.dbs.elki.algorithm.outlier.GaussianModel`.

Histogram izlaznih vrijednosti sa slike 2.1 dobiven je za ulazni skup podataka iz datoteke iris.csv (preuzete s [2]) uz zastavicu *invert* postavljenu na „false“. Vrijednosti odgovaraju vjerojatnostima dodijeljenim pojedinim točkama, a kako velik dio točaka ne odgovara predloženom Gaussovom modelu, većina izlaznih vrijednosti je bliže donjem rubu intervala s vrijednostima karakterističnim za anomalije.



Slika 2.1 Histogram za rezultate izvođenja algoritma GaussianModel nad datotekom *iris.csv*

Nakon sortiranja izlaznih rezultata izdvojeno je pet objekata s najmanjim vjerojatnostima u odnosu na zadanu Gaussovu funkciju. Ti su primjerci najvjerojatnije anomalije:

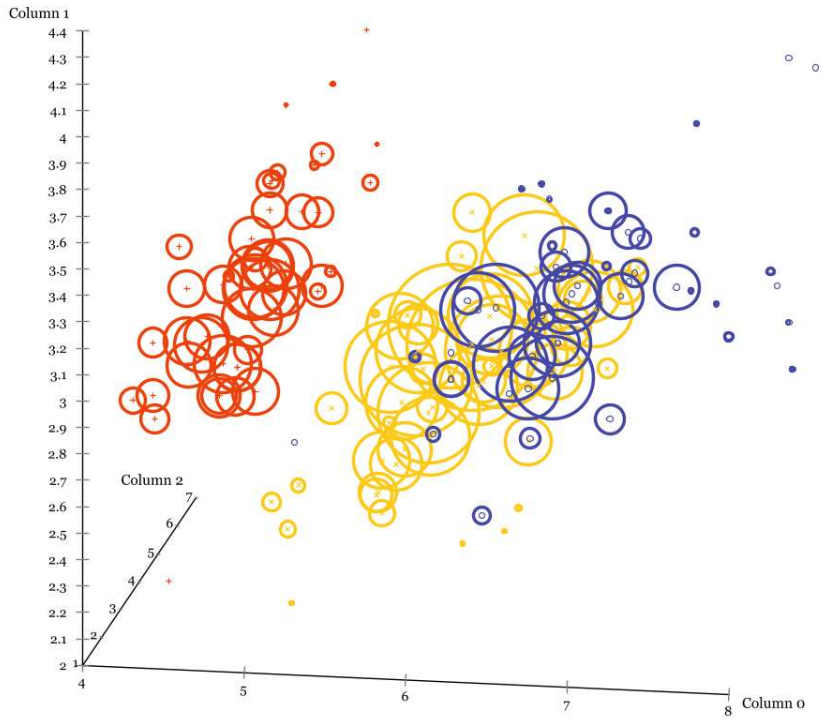
```
ID=8554    7.9    3.8    6.4    2.0    Iris-virginica    gaussian-model-
outlier=7.892077253537483E-4

ID=8557    6.1    2.6    5.6    1.4    Iris-virginica    gaussian-model-
outlier=8.964882394692772E-4

ID=8540    7.7    3.8    6.7    2.2    Iris-virginica    gaussian-model-
outlier=9.086119510224646E-4

ID=8564    6.9    3.1    5.1    2.3    Iris-virginica    gaussian-model-
outlier=0.0011081920529280947

ID=8464    4.5    2.3    1.3    0.3    Iris-setosa    gaussian-model-
outlier=0.0019005796140281316
```



Slika 2.2 Dijagram raspršenosti za rezultate izvođenja algoritma GaussianModel nad datotekom *iris.csv*

Na dijagramu raspršenosti na slici 2.2 uočljivo je kako su izlazni rezultati veći za točke u području veće gustoće. Takvo tumačenje rezultata nije uobičajeno u odnosu na ostale metode detekcije anomalija kod kojih veće vrijednosti rezultata upućuju na anomalije, stoga

ELKI omogućuje invertiranje rezultata tako da anomalijama odgovaraju najveće vrijednosti izlaza postavljanjem zastavice invert na „*true*“.

Postavljanjem zastavice poredak najvećih anomalija je sljedeći:

```
ID=8404    7.9    3.8    6.4    2.0    Iris-virginica    gaussian-model-
outlier=0.9984251653854941

ID=8407    6.1    2.6    5.6    1.4    Iris-virginica    gaussian-model-
outlier=0.9982110911162446

ID=8390    7.7    3.8    6.7    2.2    Iris-virginica    gaussian-model-
outlier=0.9981868987015016

ID=8414    6.9    3.1    5.1    2.3    Iris-virginica    gaussian-model-
outlier=0.99778864404338

ID=8314    4.5    2.3    1.3    0.3    Iris-setosa    gaussian-model-
outlier=0.9962074641851052
```

Primjerci su ostali isti, ali su vrijednosti rezultata za njih sada približno 1.

2.2. Algoritam *k*-najbližih susjeda

2.2.1. Opis algoritma *k*-najbližih susjeda

Algoritam *k*-najbližih susjeda (engl. *k-nearest neighbors*) ili kraće algoritam *k*-NN je metoda otkrivanja stršećih vrijednosti temeljena na udaljenosti. Algoritam pronalazi *k* najbližih susjeda nekog objekta u skupu podataka s obzirom na definiranu funkciju udaljenosti, pri čemu je *k* parametar koji se zadaje algoritmu. Uz temeljnu pretpostavku da se slične pojave u podacima nalaze u međusobnoj blizini, udaljenost do *k*-tog susjeda može se koristiti kao mjera za određivanje stršećih vrijednosti (engl. *outlier score*) [4]. Postoje različite varijante algoritma koje koriste najveću udaljenost u definiranom *k*-susjedstvu točke, tj. udaljenost do *k*-tog susjeda, srednju vrijednost ili medijan svih udaljenosti *k*-NN. Implementacija algoritma u ELKI-ju odgovara prvoj varijanti. Veća udaljenost upućuje na manju lokalnu gustoću točaka te mogućnost da promatrana točka predstavlja stršeću vrijednost. Uobičajeno je korištenje euklidske funkcije udaljenosti, pri čemu se primjerci podataka predstavljaju kao višedimenzionalni vektori. Udaljenost do *k*-tog najbližeg susjeda točke *p* (tzv. *k*-udaljenost) onda možemo računati izrazom:

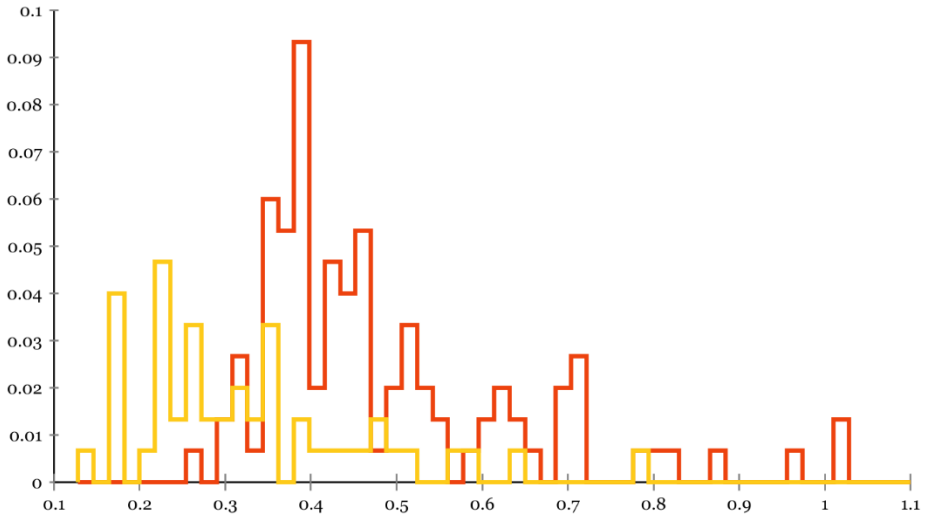
$$D^k(p) = \sqrt{(x - x_k)^2 + (y - y_k)^2} \quad . \tag{4}$$

Parametar *k* zadaje se algoritmu i može imati značajnog utjecaja na ishod izvođenja. Točke s najvećom *k*-udaljenošću najvjerojatnije su stršeće vrijednosti.

2.2.2. Primjena algoritma *k*-najbližih susjeda

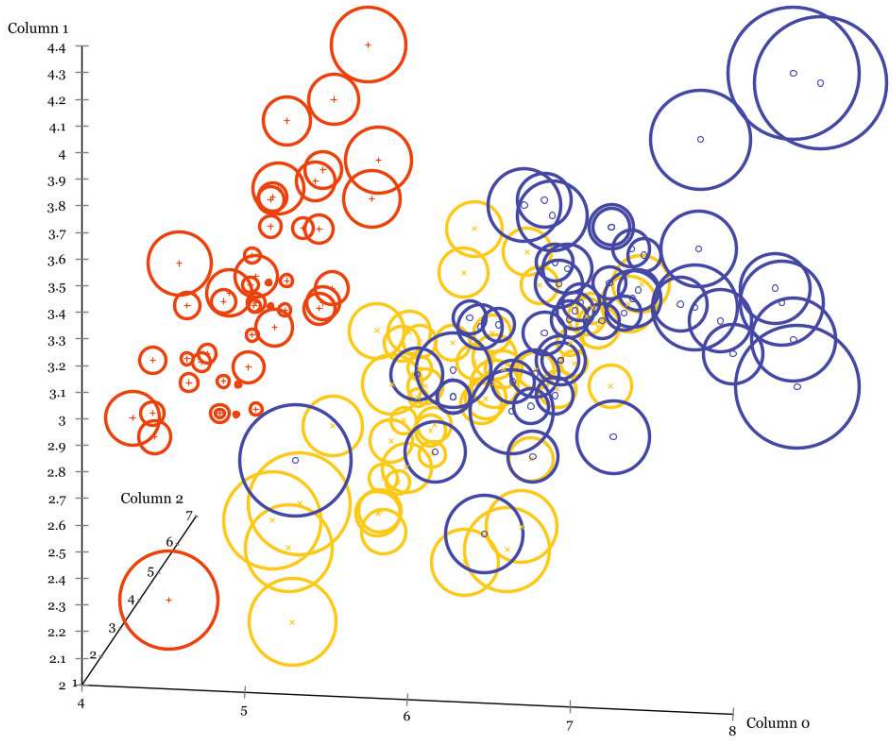
U ELKI-ju je ovaj algoritam modeliran razredom `de.lmu.ifi.dbs.elki.algorithm.outlier.distance.KNNOutlier`. Kod konstrukcije objekta zadaje se obavezno parametar *k*.

Detekcija stršećih vrijednosti u sljedećim primjerima provedena je na podacima iz datoteke *iris.csv* s 4 atributa i 150 primjeraka. Kao funkcija udaljenosti korištena je euklidska udaljenost. Slika 2.3 prikazuje histogram rezultata za proveden algoritam kojem je kao *k* zadan broj 5. Anomalijama očekivano pripadaju veće vrijednosti izlaznih rezultata, pa je crvena krivulja pomaknuta u odnosu na krivulju normalnih vrijednosti u desno.



Slika 2.3 Histogram rezultata izvođenjem algoritma kNN za ulaznu datoteku *iris.csv*

Na dijagramu raspršenosti na slici 2.4 s vrijedostima prva tri stupca jasno je uočljivo kako su točke udaljenije od mjesta najveće koncentracije modelirane većim krugovima, odnosno da im je izvođenjem algoritma pripisan veći izlazni rezultat.



Slika 2.4 Dijagram raspršenosti dobiven izvođenjem algoritma kNN za ulaznu datoteku *iris.csv*

Nakon sortiranja podataka prema dobivenom rezultatu može se izdvojiti pet primjeraka s najvećim rezultatom:

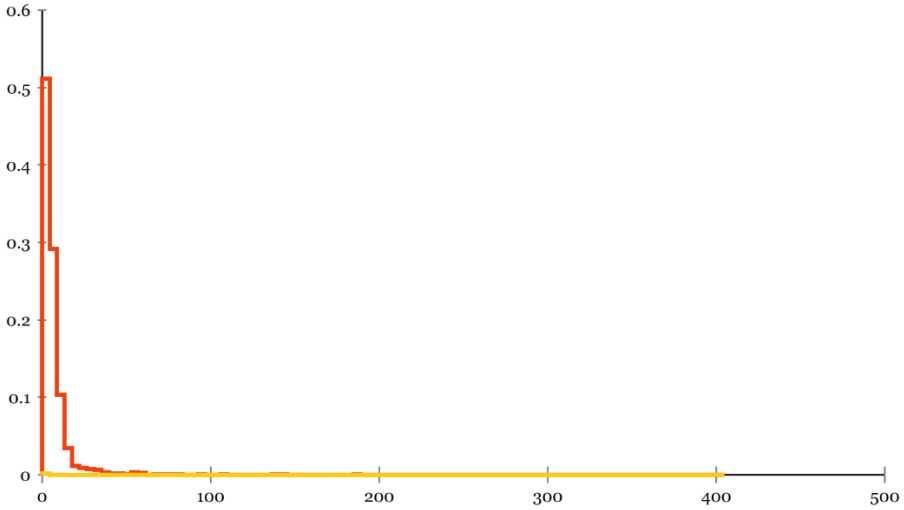
```
ID=432 7.9 3.8 6.4 2.0 Iris-virginica knn-outlier=0.8831760866327846
ID=418 7.7 3.8 6.7 2.2 Iris-virginica knn-outlier=0.818535277187245
ID=407 4.9 2.5 4.5 1.7 Iris-virginica knn-outlier=0.7615773105863909
ID=342 4.5 2.3 1.3 0.3 Iris-setosa    knn-outlier=0.7141428428542852
ID=410 7.2 3.6 6.1 2.5 Iris-virginica knn-outlier=0.6708203932499366
```

U nastavku je opisano 5 primjeraka s najvišim dobivenim rezultatom pri izvođenju algoritma uz ulazni parametar jednak 20. Stršeće vrijednosti izdvojene na ovaj način djelomično se razlikuju od primjeraka izdvojenih uz ulazni parametar 2, što ilustrira osjetljivost algoritma na odabir vrijednosti *k*.

```
ID=732 7.9 3.8 6.4 2.0 Iris-virginica knn-outlier=1.7944358444926367
```

```
ID=719 7.7 2.6 6.9 2.3 Iris-virginica knn-outlier=1.7832554500127007
ID=718 7.7 3.8 6.7 2.2 Iris-virginica knn-outlier=1.7029386365926404
ID=723 7.7 2.8 6.7 2.0 Iris-virginica knn-outlier=1.5811388300841898
ID=699 5.1 2.5 3.0 1.1 Iris-versicolor knn-outlier=1.4352700094407327
```

Slika 2.5 dobivena je vizualizacijom rezultata izvođenja za ulaznu datoteku *ad.data* preuzetu s [2] s 1558 numeričkih atributa uz vrijednost parametra k=15. Histogram daje dobar uvid u činjenicu da algoritam kNN nije prikladan za primjenu na podacima velikih dimenzija. Rezultati za određivanje stršećih vrijednosi su za gotovo sve primjerke mali u odnosu na mogući raspon vrijednosti i podjednaki iznosom, pa nisu posebno informativni.



Slika 2.5 Histogram dobiven algoritmom kNN za datoteku *ad.data*

2.3. Algoritam LOF

2.3.1. Opis algoritma LOF

Lokalni faktor stršećih vrijednosti (engl. *local outlier factor*, LOF) je vrijednost koja opisuje u kojoj je mjeri promatrana točka izolirana u odnosu na svoje okruženje. Promatrano lokalno područje određeno je s k-najbližih susjeda, gdje je k zadani ulazni parametar. Algoritam LOF spada u skupinu metoda temeljenih na gustoći. Uspoređivanjem lokalne

gustoće objekta s lokalnom gustoćom susjednih identificiraju se regije slične gustoće i točke sa znatno manjom gustoćom u odnosu na svoje susjede koje predstavljaju kandidate za stršeće vrijednosti [5]. Prednost algoritma u odnosu na algoritam *k*-NN je u mogućnosti otkrivanja lokalnih stršećih vrijednosti.

Na dobivene rezultate uvelike utječe izbor parametra *k*. Uz preveliku vrijednost *k* moguće je da će lokalne stršeće vrijednosti ostati neprimijećene. Kod malih vrijednosti *k* algoritam je više lokalno usmjeren, ali i skloniji pogreškama u skupovima podataka s mnogo šuma.

Udaljenost objekta do njegovog *k*-tog susjeda je *k*-udaljenost (engl. *k-distance*). Dohvatljiva udaljenost (engl. *reachability distance*) definira se kao maksimum udaljenosti dviju točaka i *k*-udaljenosti druge točke:

$$reach - dist_k(A,B) = max\{k - distance(B), d(A,B)\}. \quad (5)$$

Dohvaljiva udaljenost objekta A od objekta B je udaljenost tih dvaju objekata, ali ne manja od *k*-udaljenosti objekta B. Taj se rezultat dalje koristi za računanje tzv. lokalne dohvatljive udaljenosti (engl. *local reachability density*). Najprije se izračuna dohvatljiva udaljenost prema objektu A od svih njegovih susjeda. Lokalna dohvatljiva udaljenost objekta A je inverzna vrijednost prosjeka tih *k* udaljenosti dana izrazom:

$$ldr_k(A) = \frac{|N_k(A)|}{\sum_{B \in N_k(A)} reach\text{-}dist_k(A,B)}. \quad (6)$$

Konačno, vrijednost LOF nekog objekta u skupu podataka dobiva se kao omjer prosjeka lokalne dohvatljivosti *k*-susjednih objekata i lokalne dohvatljive udaljenosti samog promatranog objekta A prema izrazu:

$$\mathrm{LOF}_k(\mathrm{A}) = \frac{\sum_{\mathrm{B} \in \mathrm{N}_k(\mathrm{A})} \mathrm{lrd}_k(\mathrm{B})}{|\mathrm{N}_k(\mathrm{A})| \cdot \mathrm{lrd}_k(\mathrm{A})}. \quad (7)$$

Ako vrijednost LOF iznosi približno 1, objekt je usporediv sa susjedima i nije stršeća vrijednost. Vrijednost manja od 1 upućuje da objekt ima gustoću veću nego susjedni objekti, dok je vrijednost veća od 1 tipična za područja manje gustoće i stršeće vrijednosti. Zahvaljujući lokalnom pristupu, algoritam LOF može pronaći stršeće vrijednosti u odnosu na pojedini podskup ulaznog skupa.

2.3.2. Primjena algoritma LOF

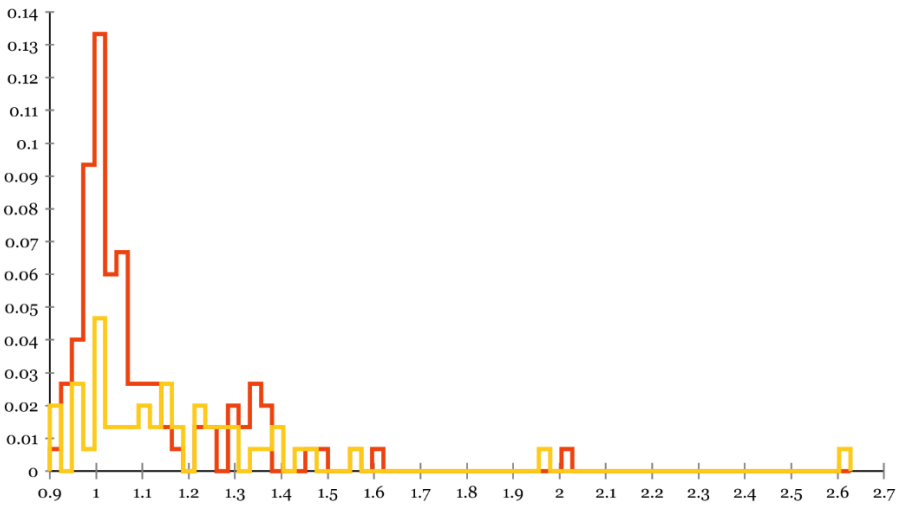
Algoritam LOF je u ELKI-ju modeliran razredom `de.lmu.ifi.dbs.elki.algorithm.outlier.lof.LOF`. Pri konstrukciji je obavezno zadati parametar *k* za određivanje *k*-susjedstva svakog objekta. Slika predstavlja histogram izlaznih vrijednosti za ulaznu datoteku *iris.csv*.

Kao pet vrijednosti s najvećim odstupanjem može se izdvojiti:

```
ID=642 4.5 2.3 1.3 0.3 Iris-setosa lof-outlier=2.6159986694868995
ID=707 4.9 2.5 4.5 1.7 Iris-virginica lof-outlier=2.014742368442045
ID=623 4.6 3.6 1.0 0.2 Iris-setosa lof-outlier=1.9565944789706058
ID=710 7.2 3.6 6.1 2.5 Iris-virginica lof-outlier=1.6025838047637768
ID=625 4.8 3.4 1.9 0.2 Iris-setosa lof-outlier=1.5628870071703473
```

Izdvojeni primjerci su različiti, a rezultati neusporedivi s primjercima izdvojenim primjenom drugih pristupa.

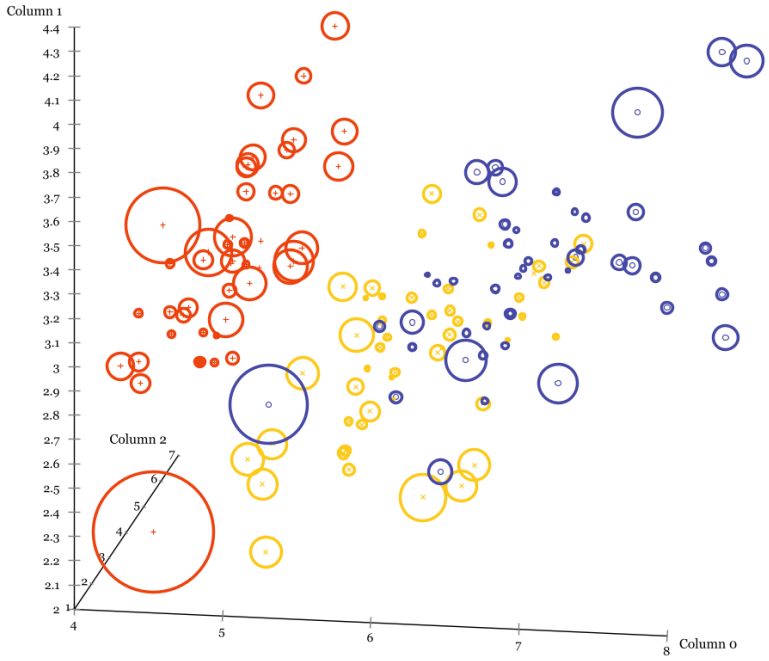
Na histogramu na slici 2.6 je uočljivo kako je rezultat za većinu primjeraka (koji pripadaju gušćim područjima) približno 1, a rezultati za stršeće vrijednosti su očekivano nešto veći.



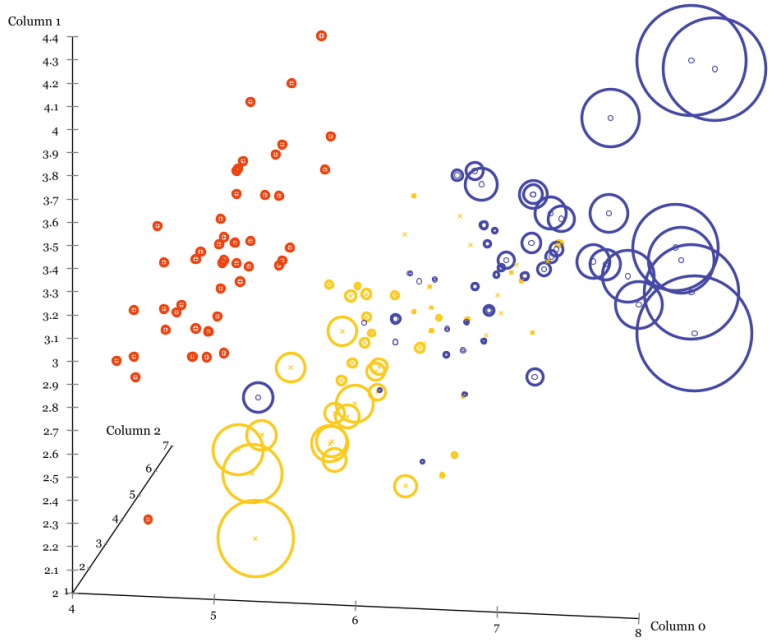
Slika 2.6 Histogram dobiven algoritmom LOF za ulaznu datoteku *iris.csv*

Usporedbom grafa raspršenosti na slici 2.7 uz zadan parametar *k*=5 i grafa na slici 2.8 koji je vizualizacija rezultata s parametrom *k*=50 za isti ulazni skup podataka uočljivo je

kako se znatnim povećanjem vrijednosti *k* narušava sposobnost algoritma za otkrivanjem anomalija na lokalnoj razini. Graf na slici 2.8 ukazuje na to da je algoritam za veliki *k* manje osjetljiv za razlikovanje vrijednosti unutar područja veće gustoće.



Slika 2.7 Dijagram raspršenosti uz parametar *k*=5



Slika 2.8 Dijagram raspršenosti uz parametar k=50

2.4. Algoritam LoOP

2.4.1. Opis algoritma LoOP

Algoritam lokalnih vjerojatnosti stršećih vrijednosti (engl. *local outlier probabilities*), ili kraće LoOP, izveden je iz LOF algoritma kao alternativna metoda manje osjetljiva na izbor parametra k. Algoritam kombinira vrijednosti temeljene na gustoći (poput LOF) s probabilističkim pristupom [6].

Izlazne vrijednosti algoritma pripadaju intervalu [0, 1] i mogu se izravno interpretirati kao vjerojatnost da je objekt stršeća vrijednost. Vrijednost će biti bliža 0 za točke unutar područja veće gustoće, dok vrijednosti bliže 1 karakteriziraju stršeće vrijednosti, tj. područja manje gustoće.

Ako je *D* skup podataka s *n* objekata, vjerojatnosna udaljenost (engl. *probabilistic distance*) *pdist*(*o*, *S*) objekta *o* iz skupa *D* u odnosu na kontekstni skup $S \subseteq D$ je vrijednost sa svojstvom da je udaljenost nekog objekta *s* iz kontekstnog skupa *S* do objekta *o* manja nego vjerojatnosna udaljenost *pdist* s vjerojatnošću od najmanje φ. To je svojstvo opisano izrazom:

$$\forall s \in S\colon P[d(o,s) \leq pdist(o,S)] \geq \varphi \ . \tag{8}$$

Za lokalnu procjenu gustoće *S* može se umjesto φ koristiti izraz (9), gdje je *erf* Gaussova funkcija pogreške opisana izrazom (10):

$$\lambda = \sqrt{2} \cdot erf^{-1}(\varphi) \quad , \tag{9}$$

$$erf(\varphi) = \frac{2}{\sqrt{\pi}} \int_0^{\varphi} e^{-t^2}\, dt \ . \tag{10}$$

Ako se uz to pretpostavi da se *o* nalazi u središtu skupa *S*, može se računati standardna udaljenost objekta u skupu *S* u odnosu na *o* prema izrazu:

$$\sigma(o,S) = \sqrt{\frac{\sum_{s \in S} d(o,s)^2}{|S|}} \ . \tag{11}$$

Skup *S* određen je s *k*-najbližih susjeda oko *o* i pretpostavlja se da otprilike podliježe normalnoj raspodjeli. Definira se probabilistička funkcija skupa objekta *o* u odnosu na *S* izrazom:

$$pdist(\lambda, o, S) = \lambda \cdot \sigma(o,S) \ . \tag{12}$$

Vjerojatnosni lokalni faktor stršećih vrijednosti (engl. *probabilistic local outlier factor*) definira se kako je opisano u izrazu (13) i predstavlja omjer procjene gustoće oko *o* i očekivane vrijednosti procjena gustoća oko svih objekata u skupu *S(o)*:

$$\mathrm{PLOF}_{\lambda,\mathrm{S}}(o) = \frac{pdist(\lambda,o,S(o))}{E_{s \in S(o)}[pdist(\lambda,s,S(s))]} - 1 \ . \tag{13}$$

Računa se vrijednost koja predstavlja standardnu devijaciju PLOF vrijednosti izrazom:

$$\mathrm{nPLOF} = \lambda \cdot \sqrt{E[PLOF^2]} \ . \tag{14}$$

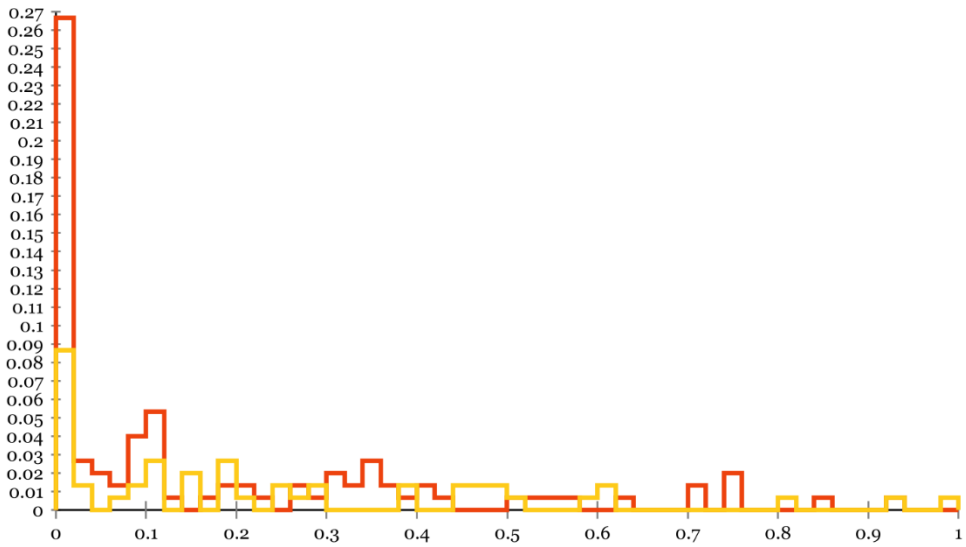
Konačno, primjenom funkcije pogreške dobiva se lokalna vjerojatnost stršećih vrijednosti (engl. *local outlier probability*), opisana izrazom (15), kao veličina za određivanje u kojoj je mjeri jedna vrijednost u skupu podataka stršeća:

$$\mathrm{LoOPs(o)} = \max\left\{0, \mathrm{erf}\left(\frac{\mathrm{PLOF}_{\lambda,\mathrm{S}}(\mathrm{o})}{\mathrm{nPLOF} \cdot \sqrt{2}}\right)\right\} \ . \tag{15}$$

Vrijednosti dobivene ovim pristupom konzistentne su unutar jednog skupa podataka, ali i u različitim skupovima, stoga omogućuju uspoređivanje svih rezultata stršećih vrijednosti u kontekstu skupa podataka, ali i među različitim skupovima [6].

2.4.2. Primjena algoritma LoOP

Primjena je ponovno ilustrirana primjerom na podacima iz datoteke *iris.csv*. Na slici 2.9 prikazan je histogram izlaznih vrijednosti dobiven za izvođenje algoritma uz vrijednost parametra k=5. Zbog opisanog mehanizma generiranja rezultata, specifičnost je algoritma u odnosu na druge lokalne pristupe ta da su izlazne vrijednosti za pojedine primjerke normalizirane, tj. ograničene na interval [0, 1].



Slika 2.9 Histogram dobiven alhoritmom LoOP za ulaznu datoteku *iris.csv*

Izdvajanjem najviših izlaznih vrijednosti dobivaju se primjerci:

```
ID=2092 4.5 2.3 1.3 0.3 Iris-setosa loop-outlier=0.9920364657211175
ID=2157 4.9 2.5 4.5 1.7 Iris-virginica loop-outlier=0.9286949626375781
ID=2073 4.6 3.6 1.0 0.2 Iris-setosa loop-outlier=0.9239630350581469
ID=2113 6.0 2.2 4.0 1.0 Iris-versicolor loop-outlier=0.8538210802571655
ID=2075 4.8 3.4 1.9 0.2 Iris-setosa loop-outlier=0.8049015671793139
```

Dobivene stršeće vrijednosti u potpunosti se razlikuju od ranije izdvojenih drugim pristupima.

2.5. Algoritam OPTICS-OF

2.5.1. Opis algoritma OPTICS-OF

Algoritam redoslijeda točaka za identificiranje strukture grupa (engl. *ordering points to identify the clustering structure*), ili kraće OPTICS, je algoritam za pronalažnje grupa temeljenih na gustoći u skupovima podataka. OF u nazivu označava faktor stršećih vrijednosti (engl. *outlier factor*). Kod primjene za računanje faktora stršećih vrijednosti, ovaj algoritam spada u lokalne pristupe i daje rezultate koji su usporedivi s najbližim susjednim točkama, a ne nužno i sa cijelim skupom [8].

U općenitom slučaju algoritam prima dva parametra: ε koji označava radijus oko točaka i *MinPts*, koji predstavlja minimalan broj točaka potreban za formiranje grupe. Definira se tzv. jezgrena udaljenost točke p koja predstavlja jedan primjerak podataka izrazom (16) u kojem je $|N_\varepsilon(p)|$ susjedstvo točke p određeno radijusom ε:

$$core - dist_{\varepsilon,MinPts}(p) = \begin{cases} nedefinirano, & za\ |N_\varepsilon(p)| < MinPts \\ MinPts - ta\ najmanja\ udaljenost\ u\ N_\varepsilon(p), & inače \end{cases}. \quad (16)$$

Dohvatljiva udaljenost druge točke o iz točke p definira se formulom:

$$reach - dist_{\varepsilon,MinPts}(o,p) = \begin{cases} nedefinirano, & za\ |N_\varepsilon(p)| < MinPts \\ \max\left(core - dist_{\varepsilon,MinPts}(p), dist(p,o)\right), & inače \end{cases}. \quad (17)$$

Opisane vijednosti nisu definirane ako u skupu nije dostupna grupa dovoljne gustoće. Parametar ε služi za odbacivanje nedovoljno gustih grupa koje nisu od interesa.

Kod primjene algoritma za detekciju stršećih vrijednosti uvodi se veličina lokalne gustoće dohvatljivosti (engl. *local reachibility density*) opisana izrazom (18) kao inverzna vrijednost prosječne dohvatljive udaljenosti iz *MinPts* najbližih susjednih točaka do p:

$$ldr_{MinPts}(p) = 1 \Big/ \frac{\sum_{o \in N_{MinPts}(p)} reach - dist_{\infty,MinPts}(p,o)}{|N_{MinPts}(p)|}. \quad (18)$$

Konačno, faktor koji određuje stršeće vrijednosti računa se izrazom:

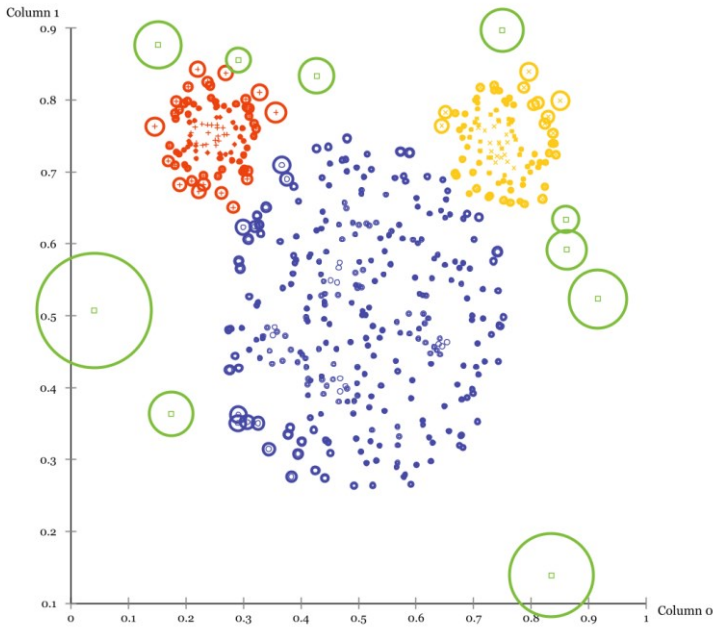
$$OF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \quad . \tag{19}$$

Faktor računa prosjek omjera *ldr* vrijednosti *MinPts* najbližih susjeda i *p*. On iznosi 1 kada su te vrijednosti jednake, a to je karakteristika točaka unutar grupa jednolike gustoće. Više vrijednosti faktora upućuju na anomalije.

2.5.2. Primjena algoritma OPTICS-OF

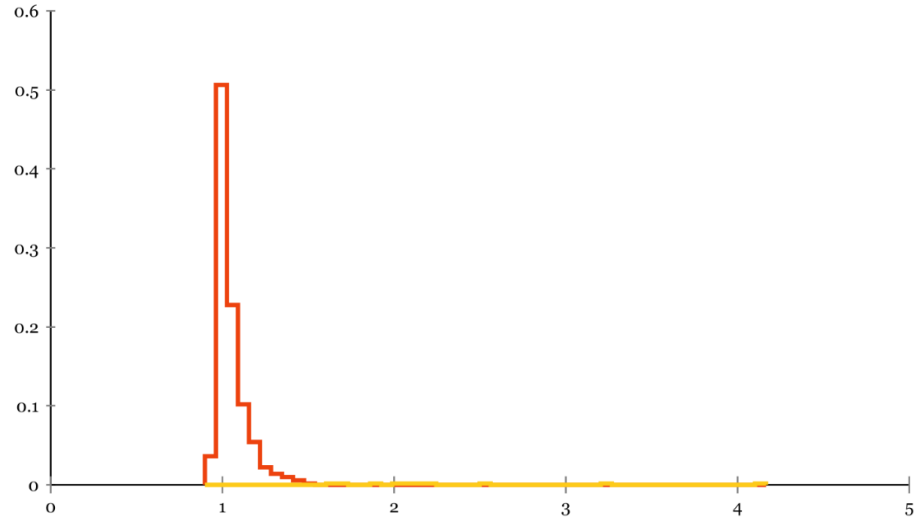
Razred `de.lmu.ifi.dbs.elki.algorithm.outlier.OPTICSOF` sadrži implementaciju algoritma OPTICS za otkrivanje anomalija. Kao obvezan parametar zadaje se broj točaka *MinPts* potreban za formiranje grupe.

Ponašanje algoritma je najbolje predočeno na slici 2.10 dijagramom raspršenosti za dvodimenzionalni skup podataka *mouse.csv* preuzet s [1], koji sadrži tri grupirana područja i točke koje predstavljaju šum. Krugovi koji predstavljaju stršeće vrijednosti najvećeg su polumjera, a osim njih, u manjoj se mjeri ističu i krugovi oko točaka smještenih na rubovima grupa. Ta je pojava vidljiva i kod ispisa sortiranih rezultata: prve su vrijednosti točke označene kao šum (engl. *noise*), zatim slijede točke različitih oznaka s rubova grupa.



Slika 2.10 Dijagram raspršenosti dobiven algoritmom OPTICS-OF za ulaznu datoteku *mouse.csv*

Na prikazu histograma na slici 2.11 uočljivo je kako većina izlaznih rezultata iznosi približno 1, što je karkteristika točaka unutar grupa. Za stršeće vrijednosti rezultati teže prema nešto većim vrijednostima.



Slika 2.11 Histogram dobiven algoritmom OPTICS-OF za ulaznu datoteku *mouse.csv*

2.6. Algoritam *DB-outlier*

2.6.1. Opis algoritma *DB-outlier*

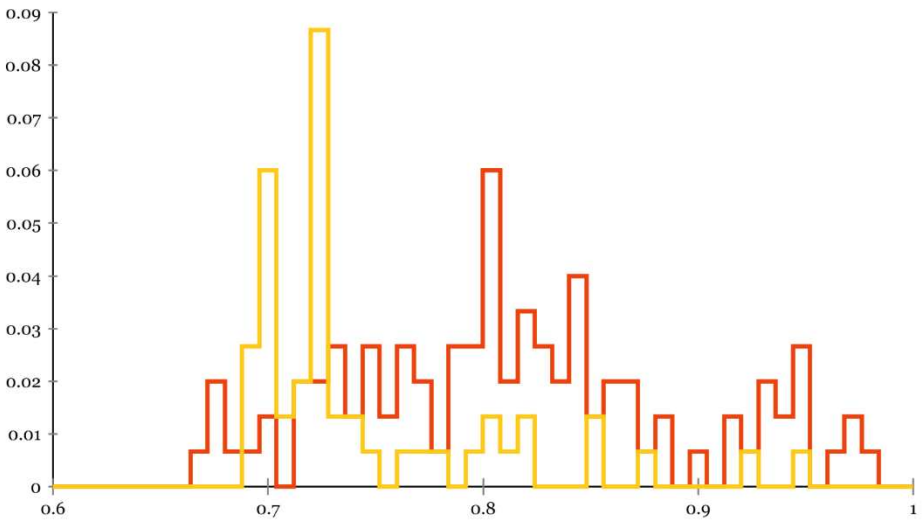
Stršeća vrijednost na temelju udaljensoti (engl. *DB-outlier*) definirana je kao objekt o iz skupa podataka T od kojeg je barem p udio objekata iz T udaljen za više od D. Parametar p određuje udio objekata koji se moraju nalaziti izvan D-susjedstva objekta o. D i p zadaju se kao parametri algoritma. Sa M se označava najveći dozvoljeni broj objekata u susjedstvu točke o. Algoritam se provodi računanjem udaljenosti između ispitivanog objekta i svih ostalih. Ako se pronađe manje od M točaka za koje vrijedi da je udaljenost manja od D, tj. ako D-susjedstvo sadrži manje od M objekata, ispitivani objekt se klasificira kao stršeća vrijednost. Ovaj je algoritam jedan od jednostavnijih i često korištenih za detekciju stršećih vrijednosti.

2.6.2. Primjena algoritm *DB-outlier*

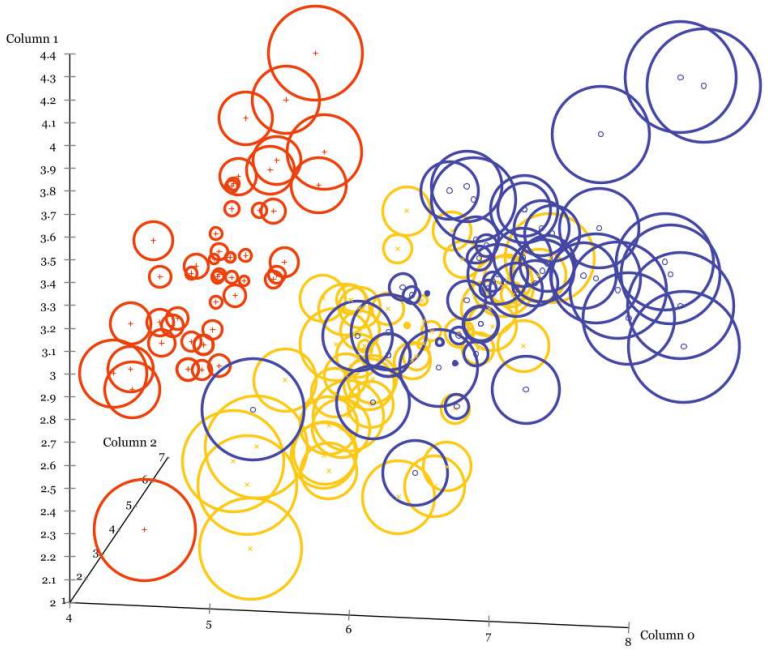
ELKI sadrži dvije implementacije ovog pristupa. Klasa `de.lmu.ifi.dbs.elki.algorithm.outlier.distance.DBOutlierScore` je

programsko ostvarenje algoritma koji kao izlaz daje kontinuirane vrijednosti za sve podatke. Kod konstrukcije potrebno je specificirati samo parametar *D*, tj. udaljenost kojom se definira susjedstvo objekata.

Histogram izlaznih vrijednosti na slici 2.12 te dijagram na slici 2.13 odnose se na ulaznu datoteku *iris.csv* uz parametar *D*=0,9.



Slika 2.12 Histogram dobiven algoritmom DB-outlier za ulaznu datoteku *iris.csv*



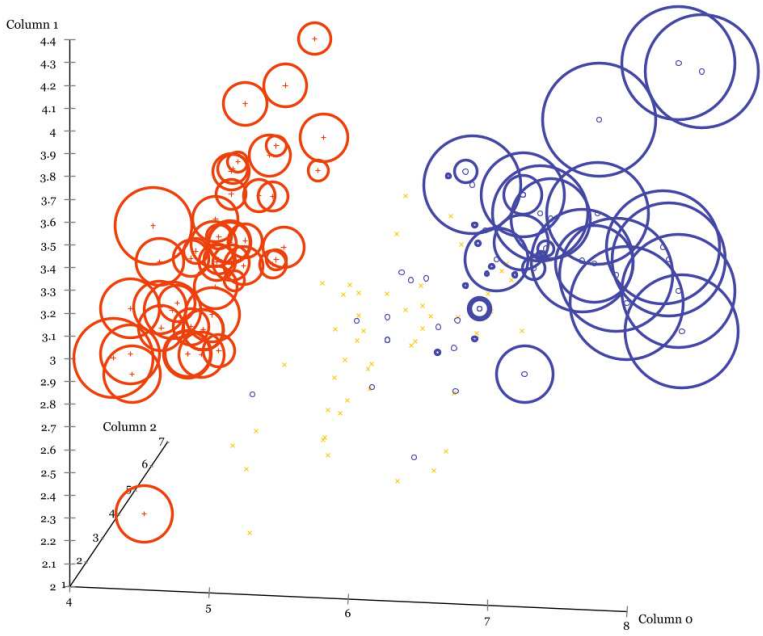
Slika 2.13 Dijagram raspršenosti dobiven algoritmom DB-outlier za ulaznu datoteku *iris.csv*

Prvih pet primjeraka nakon sortiranja prema izlaznoj vrijednosti je:

```
ID=11740 7.9 3.8 6.4 2.0 Iris-virginica db-outlier=0.98
ID=11726 7.7 3.8 6.7 2.2 Iris-virginica db-outlier=0.9733333333333334
ID=11727 7.7 2.6 6.9 2.3 Iris-virginica db-outlier=0.9733333333333334
ID=11731 7.7 2.8 6.7 2.0 Iris-virginica db-outlier=0.96
ID=11715 4.9 2.5 4.5 1.7 Iris-virginica db-outlier=0.9466666666666667
```

Ovaj poredak sličan je poretku dobivenom primjenom algoritma *k*-NN.

Algoritam je osjetljiv na odabir parametara *D*. Parametar *D* mora biti u skladu s rasponom vrijednosti atributa, a *p* određuje osjetljivost algoritma. Primjerice, zadavanjem parametra *D*=5,0 za primjerke unutar područja veće gustoće dobiju se znatno manje vrijednosti u odnosu na izolirane objekte. To je ilustrirano veličinom pripadnih krugova na slici 2.14.

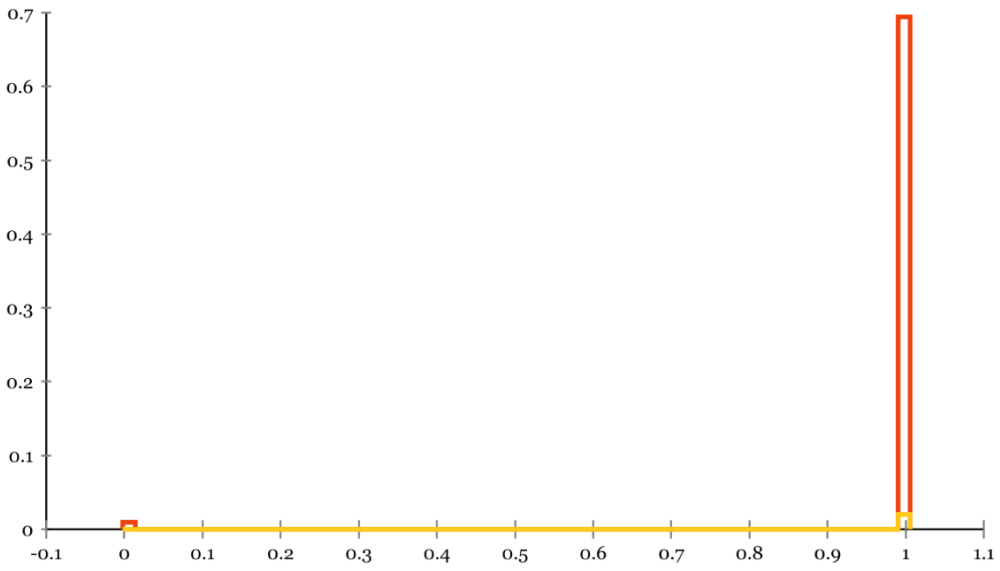


Slika 2.14 Dijagram raspršenosti za ulazni parametar *D*=5.0

Implementacija verzije algoritma koji dijeli primjerke na normalne i anomalije je `de.lmu.ifi.dbs.elki.algorithm.outlier.distance.DBOutlierDetection`.

Kod konfiguracije je osim parametra *D* potrebno zadati *p* kao broj iz intervala [0, 1] koji određuje koliki se udio podataka mora nalaziti izvan *D*-susjedstva promatranog objekta.

Na slici 2.15 je histogram rezultata dobivenih za ulaznu datoteku *mouse.csv* uz *D*=0.11 i *p*=0.8.



Slika 2.15 Histogram rezultata algoritma koji daje binarne rezultate

Uz ispravan odabir parametara, gotovo svim stršećim vrijednostima pridružena je vrijednost 1.

2.7. Algoritam ABOD

2.7.1. Opis algoritma ABOD

Algoritam za otkrivanje anomalija temeljen na kutevima (engl. *angle based outlier detection*) prikladan je za višedimenzionalne podatke kada uspoređivanje udaljenosti među primjercima u skupu više nema smisla. Svim se primjercima dodjeljuje vrijednost faktora stršećih vrijednosti temeljenih na kutu (engl. *angle-based outlier factor*), kraće ABOF.

Svi se primjerci podataka mogu prikazati kao višedimenzionalni vektori, pri čemu dimenzija odgovara broju atributa. Ovaj pristup, osim udaljenosti točaka, uzima u obzir i smjer odgovarajućih radij-vektora. Vektor razlike $\vec{B} - \vec{A}$ dviju točaka koje predstavljaju primjerke označen je kao $\vec{AB}$. Vrijednost ABOF definira se izrazom (20) kao varijanca

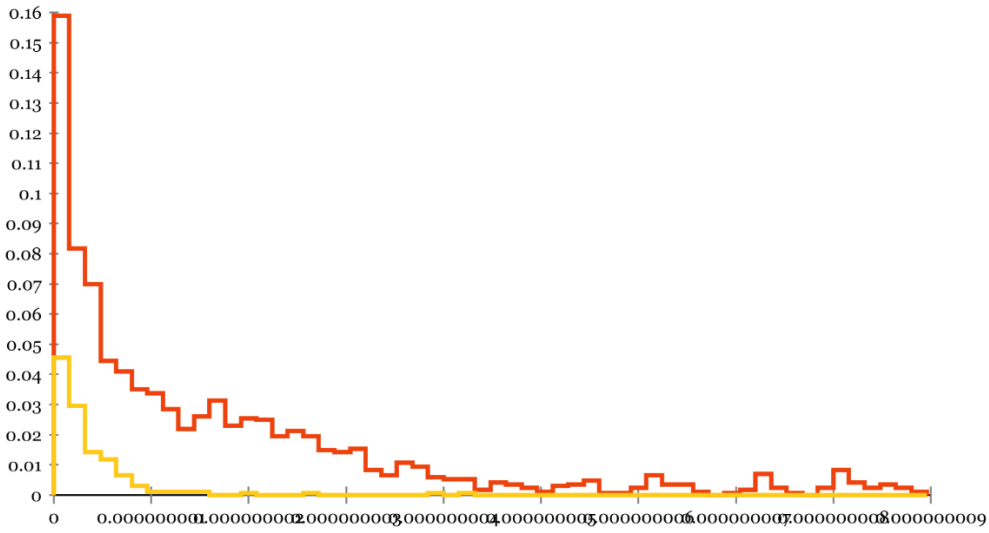
kutova između vektora razlike promatrane točke $\vec{A}$ sa svim kombinacijama parova preostalih točaka $\vec{B}, \vec{C}$ iz skupa podataka:

$$ABOF\left(\vec{A}\right) = VAR_{\vec{B},\vec{C}\in D}\left(\frac{\vec{AB}\cdot\vec{AC}}{\left|\vec{AB}\right|\cdot\left|\vec{AC}\right|}\right) \quad . \tag{20}$$

ABOF je mjera raznolikosti kutova između promatranih vektora. Ako je spektar kutova za promatranu točku širok, a varijanca veća, to znači da je točka okružena točkama raznih smjerova, tj. da se nalazi unutar grupe. Ako su vektori prema drugim točkama u samo određenim smjerovima, to označava da je točka smještena van područja koja predstavljaju grupe. Međusobno slični mali kutovi upućuju na stršeće vrijednosti [7]. Prednost je također što ovaj algoritam ne zahtijeva specifikaciju nikakvih ulaznih parametara koji bi mogli utjecati na kvalitetu rezultata. Glavni je nedostatak manja brzina izvođenja u odnosu na ostale pristupe.

2.7.2. Primjena algortima ABOD

Primjena algoritma ABOD ilustrirana je na skupu podataka iz datoteke *qsar_androgen_receptor.csv* preuzete s [2] s 1558 atributa i 3279 redaka. Za analizu podataka tih dimenzija jednostavniji pristupi (npr. temeljeni na udaljenosti) ne bi dali valjane rezultate. Iz histograma na slici je vidljivo da su izlazne vrijednosti dobivene ovim algoritmom za stršeće podatke veće nego za normalne. Dobiveni su rezultati razmjerno mali, jer se varijanca koja se koristi kao izlazna vrijednost smanjuje s povećanjem broja atributa. Kod skupova podataka manjih dimenzija, rezultati su veći.



Slika 2.16 Histogram dobiven algoritmom ABOD za ulaznu datoteku *qsar_androgen_receptor.csv*

Zaključak

Postoje brojni pristupi otkrivanju stršećih vrijednosti u podacima s različitim svojstvima i domenom primjene. Platforma ELKI nudi implementaciju raznih algoritama za otkrivanje anomalija te je stoga prikladna za analizu ponašanja i rezultata dobivenih različitim pristupima.

Pristupi se mogu podijeliti na nekoliko načina. Promatrajući skup podataka referentan pri određivanju mjere u kojoj je neka točka anomalija razlikuju se globalni i lokalni pristupi. S obzirom na izlaz koji algoritam proizvodi, metode se dijele na one s binarnim izlazom (koje svakom primjerku pripisuju oznaku normalne ili stršeće vrijednosti) te metode koje daju kontinuirani izlaz, tj. uz svaki primjerak vežu vrijednost izračunatu primjenom algoritma prema kojoj se slobodno može interpretirati u kojoj su mjeri vrijednosti stršeće.

Jedan su od mogućih pristupa otkrivanju stršećih vrijednosti statistički testovi. Jednostavan primjer takve metode je korištenje Gaussovog modela koji stršeće vrijednosti pronalazi kao točke za koje je mala vjerojatnost za određenu Gaussovu razdiobu.

Algoritmi temeljeni na udaljenosti među točkama česti su u primjeni zbog jednostavnosti. Algoritam *k*-NN i DB-outlier globalno su orijentirani, dok je algoritam LOF osmišljen kao pristup sposoban za otkrivanje stršećih vrijednosti i na lokalnoj razini. Ovi su pristupi prikladni za detekciju stršećih vrijednosti u podacima manjih dimenzija, jer se povećanjem broja atributa razlika između udaljenosti smanjuje te udaljenost kao mjera za određivanje stršećih vrijednosti postaje znatno manje korisna. Izlazi navedenih algoritama djelomično ovise o korisnički zadanim ulaznim parametrima.

Algoritam LoOP izveden je iz algoritma LOF i kombinira lokalni pristup otkrivanju anomalija temeljen na udaljenosti s probabilističkim pristupom, a manje je osjetljiv na izbor ulaznog parametra.

Algoritam OPTICS korišten za grupiranje podataka moguće je koristiti i za otkrivanje anomalija, pri čemu se onda on onda naziva OPTICS-OF. Specifičnost je ovog pristupa razlikovanje točaka koje su unutar od onih koje su na rubovima grupa.

Posebnost algoritma ABOD je što pri računanja faktora koji određuje stršeće vrijednosti u obzir osim udaljenosti među točkama uzima i kutove između vektora

definiranih točkama. To ovaj pristup čini posebno prikladnim za analizu podataka s velikim brojem atributa. Prednost je algoritma također i neovisnost o ulaznim parametrima.

Literatura

[1] Poveznica: https://elki-project.github.io/; pristupljeno 3. lipnja 2020.

[2] UCI Machine Learning Repository. Poveznica: https://archive.ics.uci.edu/ml/datasets.php; pristupljeno 2. lipnja 2020.

[3] Kriegel, H.P., Kröger, P., Zimek, A. *Outlier Detection Techniques.* The Thirteenth Pacific-Asia Conference on Knowledge Discovery and Data Mining; Poveznica: https://www.dbs.ifi.lmu.de/Publikationen/Papers/tutorial_slides.pdf; pristupljeno 25. svibnja 2020.

[4] Yang, P., Huang B., *KNN Based Outlier Detection Algorithm in Large Dataset*, 2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote Sensing; Poveznica: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5070231; pristupljeno 25. svibnja 2020

[5] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J. *LOF: Identifying Density-Based Local Outliers.* Proc. ACM SIGMOD 2000 Int. Conf. On Management of Data, Dallas, (2000). Poveznica: https://www.dbs.ifi.lmu.de/Publikationen/Papers/LOF.pdf; pristupljeno 25. svibnja 2020.

[6] Kriegel, H.P., Kröger, P., Schubert, E., Zimek, A. *LoOP: Local Outlier Probabilities*. Institut für Informatik, Ludwig-Maximilians Universität München. Poveznica: https://www.dbs.ifi.lmu.de/Publikationen/Papers/LoOP1649.pdf; pristupljeno 26. svibnja 2020.

[7] Kriegel, H.P., Schubert, M., Zimek, A. *Angle-Based Outlier Detection in High-dimensional Data.* Ludwig-Maximilians-Universität München. Poveznica: https://www.dbs.ifi.lmu.de/Publikationen/Papers/KDD2008.pdf; pristupljeno 27. svibnja 2020.

[8] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J. *OPTICS-OF: Identifying Local Outlier.* Institut für Informatik, Ludwig-Maximilians Universität München. Poveznica: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.6586&rep=rep1&type=pdf; pristupljeno: 27. svibnja 2020.

[9] Maven Repository: de.lmu.ifi.dbs.elki; Poveznica: https://mvnrepository.com/artifact/de.lmu.ifi.dbs.elki, pristupljeno 5. lipnja 2020.

Sažetak

Stršećim vrijednostima u podacima smatraju se opažanja koja po nekim svojstvima odstupaju od većine ostalih u skupu. Otkrivanje stršećih vrijednosti važan je postupak u dubinskoj analizi podataka korišten pri otkrivanju zloupotrebe platnih kartica, u medicinskoj dijagnostici, u sigurnosnim sustavima i brojnim drugim područjima. ELKI kao platforma otvorenog koda pruža implementaciju niza algoritama za grupiranje i detekciju anomalija. U ovom završnom radu dan je pregled najpopularnijih metoda za otkrivanje stršećih vrijednosti implementiranih unutar platforme ELKI temeljenih na udaljenosti, dubini, devijaciji, gustoći, itd., pri čemu su opisani principi rada, prednosti i nedostatci, specifičnosti i domene primjene pojedinih algoritama.

Summary

Outliers are defined as data points that differ significantly from other points in the data set. Outlier detection is an important procedure in data mining used in credit fraud detection, healthcare, etc. ELKI is an open source software containing algorithms for clustering and outlier detection. In this Bachelor thesis, an overview of some of the most popular approaches to outlier detection implemented in ELKI is provided, which also gives insight into each method's principles, advantages and downsides, as well as it's particulars and field of use.

Skraćenice

ELKI	*Environment for DeveLoping KDD-Applications Supported by Index-Structures*	okruženje za razvoj KDD aplikacija podržanih strukturama indeksa
KDD	*Knowledge Discovery in Databases*	otkrivanje znanja u bazama podataka
GUI	*Graphical User Interface*	grafičko korisničko sučelje
API	*Application Programming Interface*	aplikacijsko programsko sučelje
k-NN	*k-nearest neighbors*	*k*-najbližih susjeda
LOF	*Local Outlier Factor*	faktor lokalnih stršećih vrijednosti
LoOP	*Local Outlier Probabilities*	vjerojatnosti lokalnih stršećih vrijednosti
ABOD	*Angle Based Outlier Detection*	otkrivanje stršećih vrijednosti temeljeno na kutu
OPTICS	*Ordering Points To Identify the Clustering Structure*	poredak točaka za identifikaciju strukture grupiranja