Klaszterezés

Klaszterezés célja:

A klaszterezés kizárólag azokat az információkat használják fel az adatok csoportosításához amelyeket az adatokban talál és amelyek az adatpontok közötti kapcsolatokat írják le. Az a cél, hogy egy klaszter csoporton belüli objektumok egymáshoz hasonlóak legyenek vagy valamilyen kapcsolat álljon fel köztük. Annál jobb a klaszterezés minél nagyobb az egyes csoportokon belüli hasonlóság és minél nagyobb a különbség az egyes csoportok között

A klaszterezés különböző típusai

A klaszterek egy teljes gyűjteményét klaszterezésnek hívják. Az alábbi részben a különböző fajta klaszterezéseket mutatok be: hierarchikus azaz egymásba ágyazott, felosztó vagyis nem egymásba ágyazott, kizáró, átfedő, fuzzy és végül teljes vagy részleges.

hierarchikus (egymásba ágyazott)

Ha megengedjük, hogy az egyes klasztereknek lehessenek alklasztereik akkor hierarchikus klaszterezéshez jutunk, amelyet úgy is el lehet képzelni, mint egy fába rendezett egymásba ágyazott klaszterek halmazát. A fának a csúcsai a klaszterek, a levélcsúcsokat leszámítva minden csúcs a gyermekeinek az uniója. Ebből adódóan a fa gyökere az összes adatpontot tartalmazó klaszter.

felosztó (nem egymásba ágyazott)

az adathalmaz olyan, nem átfedő alcsoportokra bontását értjük, hogy mindegyik adatobjektum pontosan egy részhalmazba kerül. Egyenként véve mindegyik klasztercsoport egy felosztó klaszterezés

kizáró (exclusive):

Az egyes adatobjektumok csak egyetlen csoportba (klaszterbe) kerülhetnek.

átfedő (overlapping), **nem-kizáró (non-exclusive)**

Átfedő klaszterezésről beszélünk akkor, ha az adatobjektumok egyszerre több klaszter csoporthoz is tartozhatnak.

fuzzy

Minden objektum réasze az összes klaszternek valamilyen súlynak megfelelő mértékben, amely 0 és 1 közötti értékeket vehet fel, ahol a 0 azt jelenti, hogy egyáltalán nem tartozik bele az adott klaszterbe az egy pedig azt jelenti, hogy teljesen beletartozik, tehát fuzzy halmazokként tekinthetünk a klaszterekre.

teljes (complete) klaszterezés

mindegyik elemet hozzárendeli valamelyik klaszterhez vagyis nem megengedettek az olyan pontok amelyek egy klaszterhez sem tartoznak.

részleges

A részleges klaszterezés azt jelenti, hogy az adathalmaz egyes objektumai nem tartoznak jól meghatározott klaszterekbe. Az adathalmazbeli objektumok sokszor zajt, outlier értéket vagy érdektelen információt képviselnek.

Klaszterező algoritmusok

K-közép

A k-közép módszer egy centroidot: középpontot választ ki kezdő pontnak, amely egyszerű esetben a pontok egy csoportjának az átlaga és általában csak folytonos n-dimenziós térben elhelyezkedő pontokra alkalmazható

Az algoritmus leírása egyszerűen:

Első lépés a K darab kezdő pont kiválasztása, ahol a K paraméter értékét előre meg kell határozni a valószínűsíthető klaszterek számának megfelelően, tehát K a klaszterek mennyiségét jelenti. Ezt követően frissítjük minden klaszter középpontját a hozzájuk rendelt klaszthereknek megfelelően, majd a hozzárendelés és frissítés lépéseit folytatjuk mindaddig, amig egyetlen pont se vált klasztert vagy ezt megforditva egyetlen középpont sem változik.

Pontok legközelebbi középponthoz rendelése

Egy pont középponthoz rendeléséhez szükséges valamilyen távolsági mérték, amely meghatározza a középpontol mért távolság fogalmát. Az euklideszi távolságot gyakran használjuk euklideszi térben elhelyezkedő pontoknál, míg például dokumentumok esetében a koszinusz hasonlóság áll rendelkezésünkre. Léteznek más távolsági mértékek, ilyen például a Manhattan távolság amely alkalmazható olyan esetekben amikor az adatok euklideszi térben helyezkednek el

Kiugró értékek kérdése:

Négyzetes hiba kritérium esetén az outlier értékek negatívan befolyásolják a klaszterek megtalálását. A kiugró értékek miatt előfordulhat, hogy az eredményül kapott klaszterközéppontok és igy nem megfelelő klaszterezést kapunk, ennek megfelelően a klaszterezés jóságának mérésére használt SSE is nagyobb lesz. Ezért ilyen esetekben fontos a kiugró értékek előzetes eltávolítása. Előfordulhat azonban, hogy egyes klaszterezési eljárásoknál nem távolíthatjuk el az outlier adatokat mert azok is alapvető részét képzik az elemzésnek. Ilyenek például a pénzügyi elemzések, ahol éppen a kiugró értékek lehetnek a legérdekesebb adatok.

Fontos probléma a kiugró értékek azonosításának módja. Ha olyan megközelítéseket alkalmazunk, melyek a klaszterezés előtt távolítják el a kiugró értékeket, elkerüljük a nehezen klaszterezhető pontok klaszterezését. A kiugró értékek a feldolgozást követően is azonosíthatok Számon tarthatjuk például, hogy a pontok egyenként mennyivel járulnak hozzá az SSE értékéhez

Összevonó és felosztó hierarchikus klaszterezés

Összevonó (agglomerative)

Kiinduláskor minden pontot önálló klaszternek tekintünk és minden lépésben osszevonjuk a két legközelebbi klasztert.

Felosztó (divisive)

Kiinduláskor tekintsünk egy minden pontot magába foglaló klasztert és minden egyes lépésben osszuk ketté a klasztert addig, amig minden pont különálló klaszterré nem válik. Ebben az algoritmusban az a kérdés merül fel, hogy mi alapján válasszuk szét a klasztereket.

Az összevonó hierarchikus klaszterező módszerek messze a leggyakrabban használtak.

A hierarchikus klaszterezést gyakran ábrázolják **dendrogram** segítségével, amely reprezentálja a klaszter alklaszter kapcsolatokat, valamint az összevonások (összevonó nézőpont) vagy felosztások (felosztó nézőpont) sorrendjét is.

Alapvető összevonó hierarchikus klaszterező algoritmus

Sok összevonó hierarchikus klaszterező módszer egyetlen megközelítés változata: az egyedi pontokkal mint klaszterekkel indulva egymás után vonjuk össze a legközelebbi klasztereket egészen addig, amíg egyetlen klaszter nem marad

Előnyök és hátrányok

Általánosabban az ilyen típusú algoritmusokat akkor használjuk, ha hierarchiára van szükségünk. Ismertek továbbá olyan tanulmányok, amelyek azt állítják, hogy ezek az algoritmusok jobb minőségű klasztereket állítanak elő. Az összevonó hierarchikus klaszterező algoritmusok azonban processzor és tárigényesek. Az eljárás azon tulajdonsága, hogy az összevonások visszavonhatatlanok, problémát okozhat olyan zajos, magas-dimenziójú adatok esetén, mint például a dokumentum adatok viszont ez kezelhető bizonyos mértékig, ha először egy másik módszerrel, mint például a K-közép módszerrel klaszterezzük az adatokat.

Sűrűség alapú klaszterezés: DBSCAN

A sűrűség-alapú klaszterezés olyan nagy sűrűségű területeket keres, amelyeket alacsony sűrűségű területek választanak el egymástól. A DBSCAN egy egyszerű és hatékony sűrűség-alapú klaszterező algoritmus, amely számos olyan fogalmat szemléltet, amely fontos bármely sűrűség-alapú klaszterező megközelítés számára.

Pontok osztályozása középpont-alapú sűrűség alapján

A középpont-alapú sűrűségi megközelítés egy pontot aszerint osztályoz az elhelyezkedése alapján, hogy

(1) egy sűrű terület belsejébe (belső pont),

(2) egy sűrű terület szélére (határpont) vagy

(3) egy ritka területre (zajos vagy háttér pont) esik.

Belső (core) pontok:

Ezek a pontok a sűrűség-alapú klaszterek belső részében helyezkednek el. Egy pont belső pont, ha a távolságfüggvény és egy, a felhasználó által megadott Eps távolság paraméter által meghatározott környezetében található szomszédos pontok száma elér egy bizonyos MinPts küszöbértéket, mely szintén egy, a felhasználó által megadott paraméter

Határ (border) pontok:

Egy határpont nem belső pont, de egy belső pont környezetében van. Egy határpont akár több belső pont szomszédja is lehet.

Zajos (noise) pontok:

Minden olyan pont zajos pont, amely nem belső vagy határpont

Ha adott a belső, határ-, és zajos pontok fenti definíciója, akkor a DBSCAN algoritmus az alábbi kötetlen módon írható le. Bármely két belső pont ugyanabba a klaszterbe kerül, amennyiben elég közel egy Eps távolságon belül vannak egymáshoz. Hasonlóképpen, bármely határpontot,

amely elég közel van egy belső ponthoz, a belső ponthoz tartozó klaszterhez rendeljük A zajos pontokat figyelmen kívül

Idő- és tárbonyolultság

A DBSCAN algoritmus időbonyolultsága O(m× az Eps -sugarú környezetben található pontok megtalálásához szükséges idő), ahol m a pontok száma. Legrosszabb esetben ez O(m2). Alacsony dimenzióban azonban léteznek olyan adatszerkezetek, mint például a k-d fák, amelyek lehetővé teszik az összes pont kinyerését egy adott távolságon belül egy megadott ponthoz, és így az időbonyolultság akár is O(m*log(m))lehet. A DBSCAN tárigénye még magas dimenziójú adatoknál is O(m), mivel elég viszonylag kis számú adatot tárolni mindegyik pontra, nevezetesen a klasztercímkét és a pont besorolását mint belső, határ, vagy zajos pont.

Klaszterezés kiértékelése

Képesnek lenni annak megkülönböztetésére, hogy van-e az adatokban nem-véletlen szerkezet, csupán a klaszterellenőrzés egyik fontos szempontja.

A kiértékelés kérdései:

1. Egy adathalma **klaszterezhetőség**ének megállapítása, azaz, hogy az adatokban ténylegesen van-e nem-véletlen szerkezet.
2. A klaszterek megfelelő számának meghatározása.
3. Annak kiértékelése külső információkra való hivatkozás ***nélkül***, hogy egy klaszteranalízis eredménye milyen jól illeszkedik az adatokra.
4. Egy klaszteranalízis eredményének összehasonlítása külsőleg ismert eredményekkel, mint például külsőleg biztosított osztálycímkék.
5. Két klaszterhalmaz összehasonlítása annak meghatározására, hogy melyik a jobb.

A kiértékelés tipusai

Felügyelet nélküli

Egy klaszterezési szerkezet jóságát méri külső információk felhasználása nélkül. Ennek egy példája az SSE. A klaszter érvényesség felügyelet nélküli mértékeit gyakran két további osztályra bontják: a **klaszter kohézió** (tömörség, szorosság) mértékei, melyek azt határozzák meg, hogy milyen szorosan kapcsolódnak az objektumok egy klaszterhez, valamint a **klaszter elkülönülés** -izoláció mértéke, melyek azt határozzák meg, hogy egy klaszter mennyire kölönül el a többi klasztertől. A felügyelet nélküli mértékeket gyakran **belső indexek**nek hívják, mivel csak olyan információkat használnak fel, amelyek az adathalmazban fellelhetők.

Felügyelt

Azt méri, hogy a klaszterező algoritmus által előállított klaszterezési szerkezet mennyire illeszkedik valamilyen külső szerkezetre. A felügyelt indexek egy példája az entrópia, amely azt méri, hogy milyen jól illeszkednek a klasztercímkék külsőleg adott osztálycímkékre. A felügyelt mértékeket gyakran **külső indexek**nek nevezik, mert olyan információkat használnak fel, amelyek nem képzik az adathalmaz részét.

Relatív

Különböző klaszterezéseket vagy klasztereket hasonlít össze. Egy relatív klaszter kiértékelési mérték egy olyan felügyelt vagy felügyelet nélküli kiértékelési mérték, amely összehasonlítási célra szolgál. A relatív mértékek tehát valójában nem a klaszter kiértékelési mértékek egy külön

típusát képviselik, hanem inkább ilyen mértékek egy speciális felhasználást jelentik. Például két K-közép klaszterezést összehasonlíthatunk az SSE vagy az entrópia segítségével is.

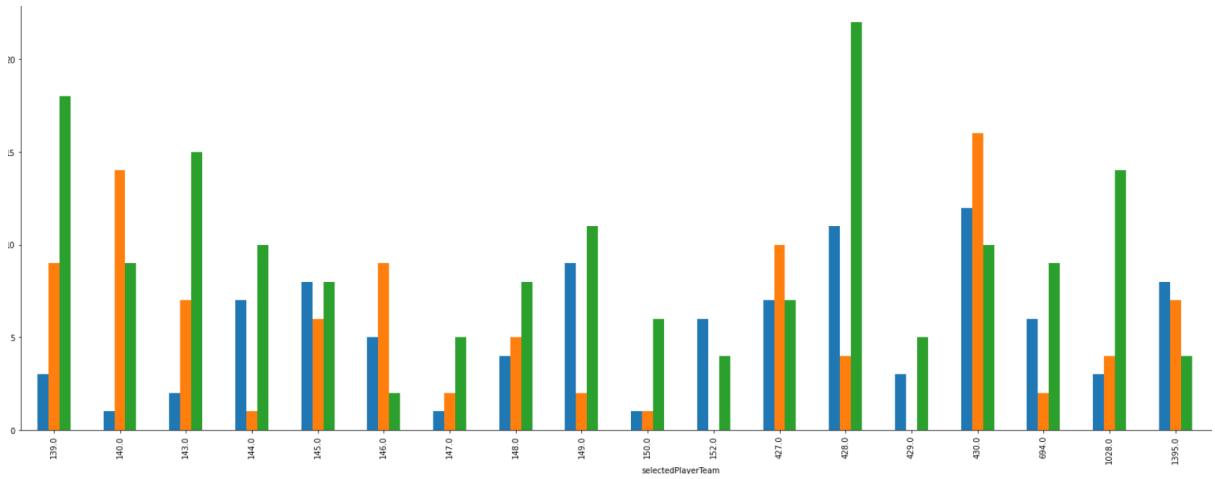
A témalaboratórium során a klaszterezési algoritmusokkal ismerkedtem meg részletesebben, ezeken belül is a k-Means algoritmussal. Megtanultam például, hogy hogyan kell megtalálni a megfelelő klaszter számot az elbow method segítségével és elmélyedtem a PCA- principal component analysis módszerében.

A félév során minden órára elkészítettem az aktuális adat halmaznak és az elvárásoknak megfelelő klaszter analíziseket.

A végső feladatom az volt, hogy a csapatokat a játékosaik teljesítményé alapján 3 klaszterbe soroljam. Ehhez a feladathoz a félév során elsajátított technológiákat használtam fel. A PCA segítségével 2 feature-re szűkítettem a rendelkezésre álló oszlopokat.

Majd az adatokat K-Means algoritmussal klaszteresztem és végül a pandas könyvtár groupby függvénye segítségével megszámoltam, hogy az egyes csapatokban, milyen klaszterekbe sorolt játékosok vannak és igy megkaptam a megfelelő elosztást közép felső és alsóház között.

Az alábbi diagram azt mutatja, hogy az egyes csapatokban hány játékos van besorolva az egyes klaszterekbe.



A hallgatótársaim előadásain keresztül a klaszteranalizis mellett megismerkedtem még más az elemzéshez szükséges lépésekkel:

1. Imputation : Az adatok tisztítása, null értékek szűrése.
2. feature selection: A fontosabb oszlopok megtartása és a többi elvetése
3. normalization
4. classification: Az adatok besorolása különböző előre megadott osztályokba
5. regression
6. clustering: Korábban részletesen taglalt.
7. evaluation: Különböző tanuló modellek értékelése

A tárgyon megszerzett tudás segítségével képessé váltam egy big data feladat megérésére és megoldására különös tekintettel a klaszterezési eljárásokra.

források: https://gyires.inf.unideb.hu/KMITT/a04/index.html