1. félévi beszámoló

Báskay János (baskayj@student.elte.hu)
Statisztikus Fizika, Biológiai Fizika és Kvantumrendszerek Fizikája PhD program
Témavezető: Pollner Péter

A dolgozat címe: Hálózat-statisztikák adattudományi alkalmazásai

Bevezetés

A 21. század információs forradalma megteremtette a lehetőséget, hogy a természeti és társadalomi jelenségek tanulmányozása során korábban elképzelhetetlennek tűnő adatmennyiséget gyűjthessünk össze. Hamar világossá vált, hogy ezeket manuálisan feldolgozni, majd előrejelzéseket alkotni lehetetlen feladat, erre a problémára nyújtott megoldást a különböző gépi tanulási módszerek megjelenése. Ezek segítségével az adatok közötti korrelációk könnyedén összegyúrhatók egy robosztus modellé, mellyel klasszifikációs vagy regressziós problémák oldhatók meg.

Ezzel párhuzamosan gyors fejlődésen ment keresztül a Hálózattudomány is, melynek keretein belül rendelkezésre áll számos eszköz a komplex rendszerek megértésére és modellezésére.

PhD tanulmányaim során szeretnék a Hálózattudomány eszközeinek felhasználásával és gépi tanulási módszerek interdiszciplináris alkalmazásán keresztül átfogó ismereteket elsajátítani az Adattudomány területén.

Aktuális félévben elvégzett kutatások ismertetése

Gépi tanulás kis adatszetten

Általában ahhoz, hogy sikeresen használjunk bármilyen gépi tanulás modellt, nagy (több ezer adatpontból álló) adatszettekre van szükség. Ezzel szemben pedálul orvosi alkalmazásnál jóval kevesebb adat áll rendelkezésre. Én az OKITI-ben összeállított gerincműtéteket tartalmazó adatokon dolgoztam, mely az orvosi szakmában nagynak számít (kb. 500 adatponttal) ám gépi tanulásra igen kicsi.

Első körben klasszifikációs feladatokat vizsgáltam: ’Milyen típusú műtétre volt szükség?’;’Volt-e több műtétje a betegnek?’ Mivel az adatszett túlnyomó részben kategorikus adatokból állt, ezeket érdemes volt bináris adattá átkódolni. (’one-hot-encoding’) Ennek a veszélye, hogy a döntési fa alapú modellek nem mindig kezelik jól az ilyen adatokat, másrészről pedig a rendelkezésre álló feature-ök (oszlopok) száma igy már egy nagyságrendbe esik az adatpontok (sorok) számával, ami szinte minden modellnek problémát okoz.

Amennyiben tudni szeretnék, hogy melyik modell jobb a másiknál, vagy éppenséggel melyik oszlopot érdemes megtartani, valamilyen validációs módszerre van szükség. A klasszikus a KFold validáció, melyben K részre osztjuk fel az adatokat, melyből K-1 részen tanítunk és 1 részen kiértékelünk. Mivel kevés adat van, igy a legextrémebb esetet is használhatjuk, amikor mindig csak egy adatpontot hagyunk meg a kiértékelésre. (’leave-one-out validáció’) A probléma az, hogy K növelésével nő a validációs pontok szórása, ugyanis csökken a teszt halmaz nagysága. Erre a out-of-bootstrap validáció jelnet megoldást, vagyis az adatszettel azonos hosszúságú véletlenszerű mintát készítünk az eredeti adatokból (megengedve az ismétlést), majd azon adatok, melyeket nem húztunk ki lesznek a teszt halmaz. Ennek a módszernek az előnye, hogy az ismétlésszám növelése nem befolyásolja a szórást. (Sőt, egyre pontosabb becslést ad rá!) Itt a szórás akkor változik, hogyha változik a bootstrap minta hossza. A 100 ismétléses out-of-bootstrap validáció egy megfelelő kompromisszum futásidő és pontosság között. (A konkrét pontozási rendszerre több lehetőség is létezik,

általában érdemes azokat használni, amelyek a *confusion mx.*-on alapszanak, pl. ROC görbe alatti területet.)

Ahhoz, hogy valóban el lehessen dönteni mely modellel érdemes dolgozni minden modellt optimalizálni kell. Ehhez rendelkezésre állnak paraméterek melyek a modell teljesítményét javíthatják (vagy ronthatják). A cél itt is az, hogy ezen paraméterek által kifeszített több dimenziós téren megtalálni az optimális beállítást, figyelemben tartva a futásidőt. Erre az egyszerű véletlenszerű keresésekhez képest egy ügyes kompromisszum a Bayes-statisztikán alapuló paraméter keresés. Ennek lényege, hogyha egy adott paraméter jobban teljesít a többinél, akkor a későbbi iterációkban nagyobb valószínűséggel húzunk majd a környékéről.

Ahogy már említettem, a bináris kódolású adatok egyik problémája, hogy jelentősen megnő a rendelkezésre álló feature-ök, száma, melyek nem feltétlenül mind hasznosak a modell számára. Ezen feature-ök összes lehetséges kombinációja egy hatalmas állapot teret feszit ki, melyen az optimumot keressük, mivel nem értékelhetünk ki minden pontot, kompromisszumot kell kötni a futásidő és a pontosság között. A klasszikus megoldás erre a problémára a greedy feature addíció/elimináció, amely az elején szintén lassú, ugyanis minden lepésben ki kell értékelni minden lehetséges eddig fel nem használt feature-t. Erre a problémára szolgál jó megoldással az összetett operátoros keresés, melynek lényege, hogy minden lépés elején sorba rakjuk az állapottéren különböző irányba mozgató operátorokat, majd a sorrend alapján egyesítjük őket. Először az első két legjobbat, majd az első hármat …stb. mindaddig, amig az összetett operátor jobban teljesít, mint az egyel kevesebb op.-ból álló összetett op.. Hogyha már nem lehet javulást elérni, akkor a megmaradt operátorok sorrendjét frissítjük és ezekből is elkezdünk összetett operátorokat végezni. Amennyiben az összetett operátorok nem javítanak a teljesítményen, akkor a módszer a greedy módszerbe megy át. Ezzel elérhető, hogy 5 lépés alatt megtaláljuk az optimális feature szettet, míg a greedy módszernek ez >20 lépésbe és több mint tízszer annyi időbe telik.

Az eddig említett módszerek segítségével egy egyszerű logisztikus regresszió AUROC pontja 100 ismétléses out-of-bootstrap validáción ***0.66****-ról* ***0.72****-re javult.*

Budapesti közösségi közlekedés hálózata

Továbbá az *Adatmodellek és adatbázisok a tudományban* nevű tárgy keretében létrehoztam egy SQL adatbázist a BKK online elérhető menetrendéből, melynek segítségével a Budapesti közösségi közlekedés hálózata felépíthető. Egy ilyen közlekedési hálózat páros gráfként viselkedik, ahol az egyik halmazt a megállók, a másikat pedig járatok adják. Érdekes megvizsgálni, hogy amennyiben leválasztjuk az egyik halmazt és azt, mint egyszerű hálózat vizsgáljuk milyen különbségek figyelhetők meg a megálló hálózat és a járat hálózat között.

A vizsgálat során a következő egyszerűsítések történtek: a hálózatokon minden él kétirányú, és két pont közötti kapcsolatnál nem vesszük figyelembe a temporális információkat (’Hány percet kell várni egy átszállásra?’), kivéve hogyha az adott járat éjszaka közlekedik.

Ezek után azt találjuk, hogy a járatok hálózatának fokszám eloszlása (p(k)) jól jellemezhető az átlagos fokszámmal, míg a megállók hálózatának p(k)-ja skálafüggetlen jelleget mutat, egy k_{min} cutoff-al, ami magyarázható azzal, hogy van egy minimális számú megálló, ami kitesz egy járatot. Klaszterzetettség terén mind a globális mind a lokális klaszterezettségi koeficiense nagyobb volt a járathálózatnak. A nagy fokszámú pontok asszortativitást tekintve neutrálisan viselkedtek mindkét hálózatban.

A jövőben érdemes megvizsgálni azt, hogyan változnak a hálózatok, hogyha a kapcsolatok létezése időben korlátozott (a menetrend alapján), illetve, hogy az ún. ’fuzzy

clustering' algoritmusok mennyiben képesek a megálló hálózatban megtalálni a nyilvánvaló csoportokat (, melyek az egy járathoz tartozó megállók).

A projectről részletesebb beszámoló található (képekkel) a https://github.com/baskayj/bkk_network címen.

Tanulmányi tevékenység az aktuális félévben

- Adatmodellek és adatbázisok a tudományban (FIZ/3/086)
- Klaszterezés hálózatokkal (FIZ/3/064E)
- Szövegbányászat és gépi tanulás R-ben (MTA poltextLAB szervezésében)

Oktatási tevékenyég az aktuális félévben

Az alapszakos Fizika hallgatók számára tartott *Modern fizika laboratóriumi gyakorlatok* című tárgy *Spektrofotometria* méréshez készített jegyzőkönyveket (16 db + revíziók) javítottam segítve ezzel Kovács Biankának a mérésvezetésben. Várhatóan a következő szemesztertől kezdve átveszem majd a mérést.