Лабораториска вежба 5
Детекција на заедници

Податочно множество – Cora

Податочното множество Cora се состои од 2 708 научни публикации класифицирани во една од седум класи. Мрежата се состои од 5 429 врски. Секој јазол претставува научен труд, а рабовите помеѓу јазлите претставуваат цитирана врска помеѓу двата документи. Секоја публикација во множеството е опишана преку вектор со 0/1 вредности кој покажува отсутност/присутност на соодветниот збор од речникот во насловот на публикацијата. Речникот се состои од 1 433 уникатни зборови. Податочното множество може да се преземе од следниот линк. Датотеката README обезбедува повеќе детали.

Како вистините заедници во дадената мрежа потребно е да се користат класите на секој од јазлите публикации. Со тоа се формираат седум заедници кои е потребно да се детектираат.

Задачи

Задача 1 – Детекција на заедници со Girvan-Newman алгоритам (3 поени)

Во оваа задача треба да се имплементира детекција на заедници со алгоритмот Girvan-Newman. Вчитајте го графот од датотеката со име „cora.cites“, која што претставува запис на листата на врски од графот. Искористете го алгоритмот Girvan-Newman за предвидување на заедници во графот. Во графот има 7 заедници кои одговараат на дадените класи за секој од јазлите. Овие класи треба да се искористат како вистински заедници во графот. Со предвидените и постоечките заедници да се пресмета:

- Adjusted Mutual Information (AMI)
  - **`sklearn.metrics.adjusted_mutual_info_score`**
- Normalized Mutual Information
  - **`sklearn.metrics.normalized_mutual_info_score`**

Задача 2 – Детекција на заедници со спектрално кластерирање (3 поени)

Да се извршат дадените барања од задача 1 преку детекција на заедници со спектрално кластерирање. За овој алгоритам може да се искористи

имплементацијата достапна од пакетот sklearn: **`sklearn.sklearn.cluster.SpectralClustering`**. Какви се добиените резултати?

Задача 3 – Детекција на заедници со длабоко учење и кластерирање (4 поени)

Да се извршат дадените барања од задача 1 со нов алгоритам за детекција на заедници. Најпрво потребно е да се креираат мрежни вгнездувања за јазлите во графот со употреба на граф автоенкодер. Овој автоенкодер е составен од енкодер кој ги мапира јазлите во латентен простор, по што следува декодер кој се обидува да ја реконструира матрицата на соседство од латентниот простор креиран со енкодерот. Дадена е имплементација longae, и за тренирање на моделот потребно е да ја извршите скриптата longae/train_reconstruction.py со следната команда:

```
python longae/train_reconstruction.py data/com-dblp.ungraph.txt 1
```

каде што **`data/com-dblp.ungraph.txt`** е локалната патека до графот, а **`1`** претставува CPU единица за тренирање на моделот (за GPU може да се користи 0).

Со извршување на оваа скрипта се креираат тежините на моделот за енкодирање на јазлите. Искористете ги овие тежини за да ги креирате и сочувате дадените мрежни вгнездувања на јазлите (разгледајте го веќе дефинираниот код во скриптата **`train_reconstruction.py`**, каде што потребна ви е променливата encoder и вчитување на тежините во оваа променлива, по што може да се направат предвидувањата).

На вака искреираните вектори на јазлите применете алгоритам за кластерирање по ваша желба (предлог: K-means или некој друг алгоритам имплементиран во библиотеката sklearn). Со предвидените кластери се дефинираат заедниците кои може да се детектираат во графот. Со овие предвидувања извршете ги барањата за евалуација како во претходните задачи и направете споредба на новиот метод со претходните методи (време, перформанс, погодност, и сл.).