

ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ
Мэдээлэл холбооны технологийн сургууль



ЛАБ 11
ТАЙЛАН

Шалгасан багш: Б.ТУЯАЦЭЦЭГ /F.SW03/

Гүйцэтгэсэн: B221910040 Н.Жавхлантөгс

Улаанбаатар 2025

Лабораторын ажил 11: Link Prediction

1.LinkPred санг суулгах ([to top](#))

Эхний алхам бол, linkpred санг суулгах бөгөөд амжилттай суусан тохиолдолд уг сангын функцүүдтэй ажиллах боломжтой болно. Санг суулгахдаа pip ашиглаж болно.

```
!pip install linkpred
```

2.Таамаглах үйл явцын дараалал

linkpred нь дараах 3-н үе шаттайгаар Холбоосыг таамаглах үйл явцыг биелүүлдэг.

- 1.Сүлжээгээ байгуулах - Network Loading
- 2.Таамаглагчаа сонгох, аппликейшн - Predictor selection and application
- 3.Үр дүн үнэлэх - Results evaluation

2.А Файлаас уншиж сүлжээг байгуулах

Эхлээд, linkpred нь файлаас графыг уншихыг шаарддаг. Бид энэ удаа Game of Thrones Season 6 (Хаадын тоглоом цуврал)-ийн ирмэгүүдийг мэдээллийг ашиглана.

```
%matplotlib inline
import matplotlib.pyplot as plt
import linkpred
import networkx as nx
def read_net_w(filename):
    g = nx.Graph()
    with open(filename) as f:
        f.readline()
        for l in f:
            l = l.split(",")
            g.add_edge(l[0], l[1], weight=int(l[2]))
    return g
```

Кодын тайлбар:

1. Сангууд

- linkpred – холбоос таамаглах алгоритмуудыг агуулсан сан.
- networkx – граф бүтээж, удирдах сан.
- %matplotlib inline – зурагнуудыг notebook дээр шууд харуулах.

2. CSV файлаас граф үүсгэх функц

- CSV файлд орой (node)-уудын хоорондын холболт, тэдгээрийн жин (жингийн утга) хадгалагдсан.
- add_edge – орой хоорондын ирмэгийг нэмнэ.

3. Game of Thrones Season 6 өгөгдлөөр граф үүсгэх

```
g = read_net_w(f'/content/got-data/got-s6-edges.csv')
```

2.B Удирдлагагүй сургалтын таамаглагч тооцоолох

linkpred нь 4 бүлэг удирдлагагүй сургалтын таамаглагчид өргөн хэрэглэгдэж байна. Үүнд: Neighborhood:

- AdamicAdar, AssociationStrength, CommonNeighbours, Cosine, DegreeProduct, Jaccard, MaxOverlap, MinOverlap, NMeasure, Pearson, ResourceAllocation

Paths:

- GraphDistance, Katz

Ranking:

- SimRank, RootedPageRank

Miscellanea:

- Community, Copy, Random

Энэ лаборатороор бид цөөн хэдэн аргыг орууллаа. Энгийн ойлгомжтой байлгахын тулд бид сонгогдсон таамаглагч бүрт топ-5 нь үр дүнг харууллаа.

2.B.1 Neighborhood

Эхний хэсэг бүлэг таамаглагч бол зангилаа бүрийн хувьд холбогдох боломжтой хос оройг уг хоёр оройн хөрш оройнуудтайгаа хэр хүчтэй сул холбогдсон байдлаас үүдэн холбоосыг үүсэх эсэх магадлалаар олдог. Энэ төрлийн таамаглагчид нь дараах асуултанд хариулт олохыг хичээдэг. How many friend we have to share in order to become friends?

Common Neighbors

Илүү олон дундын найзуудтай бол бид найз болох магадлал өндөр байна.

```
cn = linkpred.predictors.CommonNeighbours(g, excluded=g.edges())
cn_results = cn.predict()

top = cn_results.top(5)
for edge, score in top.items():
    print(edge, score)
```

Кодын тайлбар:

- CommonNeighbours: Хоёр орой нийт хэдэн хөрштэйг тооцдог.
- excluded=g.edges() – аль хэдийн холбогдсон оройнуудыг оруулахгүй.
- top(5) – хамгийн өндөр оноотой 5 таамаглалыг гаргана.

Кодын үр дүн:

```
TOMMEN - MACE 8.0
NED - LADY_CRANE 8.0
NED - CLARENZO 8.0
MARGAERY - IZEMBARO 8.0
```

Jaccard

Топологи бүтэц болон атрибутуудын ижил төсөөтэй байдал нь найз болох эсэхийг тодорхойлно.

```
jc = linkpred.predictors.Jaccard(g, excluded=g.edges())
jc_results = jc.predict()

top = jc_results.top(5)
for edge, score in top.items():
    print(edge, score)
```

Кодын тайлбар:

- linkpred.predictors.Jaccard: Энэ нь Jaccard Similarity арга. Хоёр орой хамтран хөрштэй бол тэр хэмжээгээр таамаглалаа өндөр гэж үздэг.
- excluded=g.edges(): Хоёр орой аль хэдийн холбогдсон байвал түүнийг хасна (шинэ холбоос таамаглах учраас).
- jc.predict(): Таамаглал хийнэ.
- top(5): Онооны хувьд хамгийн их магадлалтай 5 холбоосыг гаргаж ирнэ.
- for edge, score in top.items(): Тухайн таамагласан холбоосууд болон оноог хэвлэнэ.

Кодын үр дүн:

```
RED_PRIEST - KINVARA 1.0
LITTLE_SAM - DICKON 1.0
TOMMEN - MACE 0.7272727272727273
JOFFREY - CAMELLO 0.7272727272727273
OLD_NAN - BENJEN 0.7142857142857143
```

Adamic Adar

Бидний дундын найзын тоо нь биднийг найз болох эсэхийг тодорхойлоход ашиглагдана.

```
aa = linkpred.predictors.AdamicAdar(g, excluded=g.edges())
aa_results = aa.predict()

top = aa_results.top(5)
for edge, score in top.items():
    print(edge, score)
```

Кодын тайлбар:

- AdamicAdar: Хамтарсан хөрш их байх тусам, гэхдээ ховор хөрштэй бол илүү өндөр оноо өгдөг.
- Ялгаа нь Jaccard бол харьцаа, харин Adamic-Adar бол логарифмын жинтэй нийлбэр ашигладаг.

Кодын үр дүн:

```
TOMMEN - MACE 3.1875572810307893
NED - LADY_CRANE 3.126371165725784
```

```
NED - CLARENZO 3.126371165725784
SANSА - ROBB 3.0966443151332466
JOFFREY - CAMELLO 2.8944937119329133
```

2.B.2 Paths

Хоёрдогч төрлийн таамаглагчид нь хос оройн хоорондын зайнаас хамаарч тэд ирээдүйд холбогдох эсэхийг магадлаар боддог. Энэ төрлийн таамаглагчид нь дараах асуултад хариулт олохыг хичээдэг: How distant are we?

Katz

Katz computes the weighted sum over all the paths between two nodes.

```
kz = linkpred.predictors.Katz(g, excluded=g.edges())
kz_results = kz.predict()

top = kz_results.top(5)
for edge, score in top.items():
    print(edge, score)
```

Тайлбар:

- Katz: Граф дахь бүх боломжит замыг харгалзан үздэг. Хол ойрын замыг авч үздэг тул илүү нарийвчлалтай.

Кодын үр дүн:

```
LADY_CRANE - JAQEN 0.005571576860773001
SANSА - ALLISER THORNE 0.004234378077202
TYRION - DAARIO 0.004145905054793999
CERSEI - BRIENNE 0.0038862606857720003
MELISANDRE - LYANNA MORMONT 0.0035937193855959996
```

Graph Distance

Графын зай буюу Distance нь хоёр оройн хоорондын богино замын уртаар тодорхойлогдоно.

```
gd = linkpred.predictors.GraphDistance(g, excluded=g.edges())
gd_results = gd.predict()

top = gd_results.top(5)
for edge, score in top.items():
    print(edge, score)
```

Тайлбар:

- GraphDistance: Зөвхөн хамгийн богино замаар дүгнэлт хийдэг.

Кодын үр дүн:

```
LADY_CRANE - JAQEN 36.136054421768705
TYRION - DAARIO 30.29268292682927
```

```
CERSEI - BRIENNE 28.44827586206896
MELISANDRE - LITTLEFINGER 23.427491305005294
CERSEI - BRONN 23.294117647058822
```

2.B.3 Ranking

Гуравдагч бүлэг таамаглагчид нь граф дахь хос оройнуудын байрлалаас хамаарч холбогдох магадлалыг тооцоолно. Дараах асуултанд хариулж буй байдлаас хамаарч тодорхойлогдоно:
*How similar are we?

SimRank

Хоёр зангилаа холбоосоор холбогдохын тул тэдгээрийн хөрш оройнууд ижил төсөөтэй байх хэрэгтэй.

```
simrank = linkpred.predictors.SimRank(g, excluded=g.edges())
simrank_results = simrank.predict(c=0.5)

top = simrank_results.top(5)
for edge, score in top.items():
    print(edge, score)
```

Тайлбар:

- SimRank: “If two nodes are linked to similar nodes, then they are similar.” гэх ойлголт дээр тулгуурладаг.

Кодын үр дүн:

```
MAESTER_CITADEL - MAESTER_AEMON 0.3920701611834253
RAY - GATINS 0.3886464843365491
GATINS - BERIC 0.3675791855431439
THOROS - GATINS 0.3636591552182033
MAESTER_AEMON - GILLY 0.319889623157789
```

2.B.4 Miscellanea

Энэ нь холбоосыг таамаглахдаа санамсаргүй байдлаар сонгож авдаг. linkpred groups here approaches that are commonly used as baselines.

Random

Санамсаргүй байдлаар таах.

```
rnd = linkpred.predictors.Random(g, excluded=g.edges())
rnd_results = rnd.predict()

top = rnd_results.top(5)
for edge, score in top.items():
    print(edge, score)
```

Тайлбар:

- Random: Санамсаргүй үнэлгээ өгөх – харьцуулахад хэрэглэнэ.

Кодын үр дүн:

```
MAESTER_CITADEL - MAESTER_AEMON 0.3920701611834253
RAY - GATINS 0.3886464843365491
GATINS - BERIC 0.3675791855431439
THOROS - GATINS 0.3636591552182033
MAESTER_AEMON - GILLY 0.319889623157789
```

2.С Ялгаатай таамаглагчидын харьцуулах болон үнэлэх

Холбоосуудыг үүсэх эсэхыг таамаглахдаа бид сургалтын болон тестийн өгөгдлүүдтэй ажиллаж шалгана. Өгөгдлөө сургалтын болон тестийн болгон хуваах шаардлагатай.

```
import random
import itertools
from linkpred.evaluation import Pair

# Building the test network
test = read_net_w(f'/content/got-data/got-s5-edges.csv')

# Exclude test network from learning phase
training = g.copy()

# Node set
nodes = list(g.nodes())
nodes.extend(list(test.nodes()))

# Compute the test set and the universe set
test = [Pair(i) for i in test.edges()]
universe = set([Pair(i) for i in itertools.product(nodes, nodes) if i[0]!=i[1]])
```

Кодын тайлбар:

- test: Season 5 граф – шинэ холбоос үнэхээр гарсан эсэхийг шалгах туршилтын өгөгдөл.
- training = g.copy(): Season 6 графаар сургалт хийнэ.
- nodes = list(...): Бүх оройнуудыг жагсаалт болгон авна.
- itertools.product(nodes, nodes) → бүх боломжит оройны хосуудыг гаргана.
- Pair(i): Эдгээр хосуудыг Pair форматаар хадгална (үнэлгээ хийхэд ашиглагддаг тусгай бүтэц).

Дараагаар нь бид сургалтын өгөгдөл дээрээ таамаглагчаа ажиллуулна.

```
cn = linkpred.predictors.CommonNeighbours(training,
excluded=training.edges())
```

```

cn_results = cn.predict()

aa = linkpred.predictors.AdamicAdar(training, excluded=training.edges())
aa_results = aa.predict()

jc = linkpred.predictors.Jaccard(training, excluded=training.edges())
jc_results = jc.predict()

```

Тэгээд бид тестийн өгөгдлөө ашиглан таамаглагч хэр үнэн таамаглаж буй үнэлнэ.

```

cn_evaluation = linkpred.evaluation.EvaluationSheet(cn_results, test,
universe)
aa_evaluation = linkpred.evaluation.EvaluationSheet(aa_results, test,
universe)
jc_evaluation = linkpred.evaluation.EvaluationSheet(jc_results, test,
universe)

```

Тайлбар:

- EvaluationSheet(...): Таамаглалын үр дүнг үнэлнэ.
- Үндсэн 3 зүйл:
 - Recall (TPR) – Таамагласан холбоосууд дотроос жинхэнэ шинэ гарсан холбоосууд хэд байсан бэ.
 - Fallout (FPR) – Алдаатайгаар таамагласан хуурамч холбоосуудын харьцаа.
 - AUC (Area under ROC) – Нарийвчлалын хэмжүүр.

Үр дүнгүүдээ тайлбарлахдаа бид ROC -ийн муруй ашиглан дүрслэнэ

```

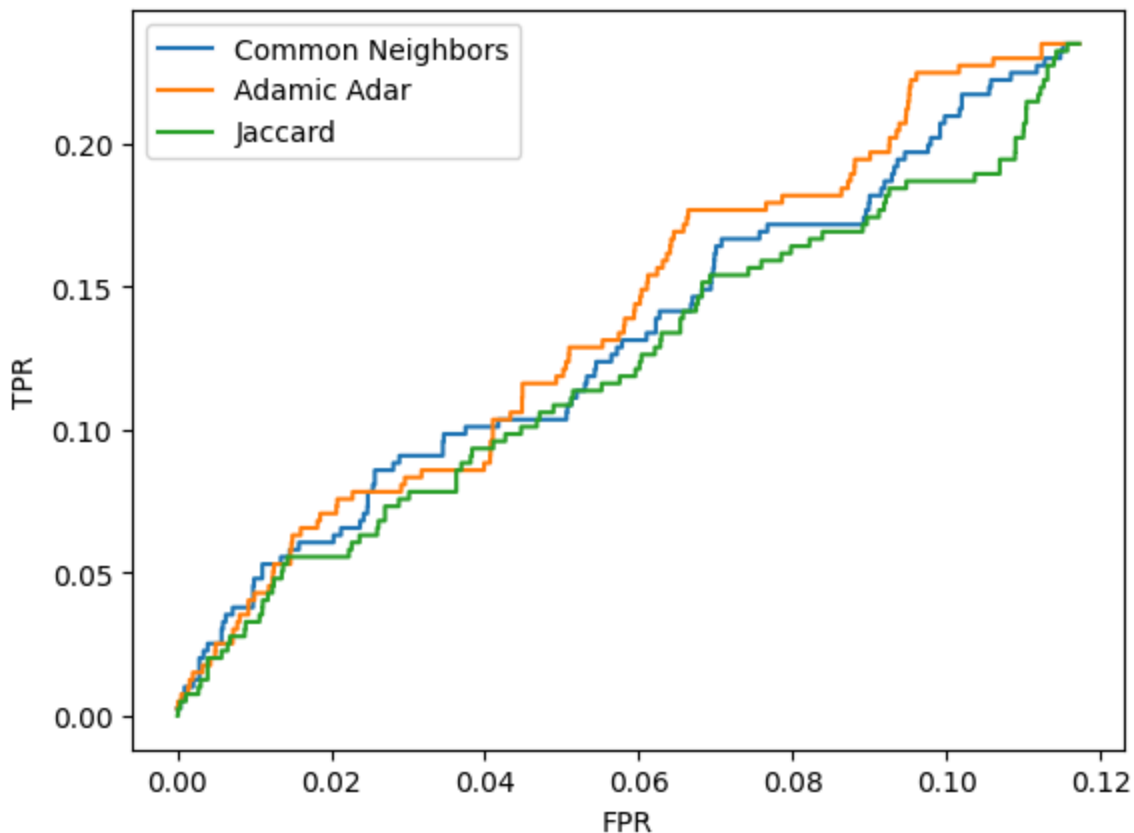
plt.plot(cn_evaluation.fallout(), cn_evaluation.recall(), label="Common
Neighbors")
plt.plot(aa_evaluation.fallout(), aa_evaluation.recall(), label="Adamic
Adar")
plt.plot(jc_evaluation.fallout(), jc_evaluation.recall(), label="Jaccard")
plt.ylabel("TPR")
plt.xlabel("FPR")
plt.legend()
plt.show()

```

Тайлбар:

- ROC Curve: Холбоос таамаглалтын нарийвчлалын харагдах хэлбэр.
- График дээр TPR vs FPR харьцуулах бөгөөд зүүн дээд булан руу ойр байх тусам илүү сайн.
- plt.legend() – Аль граф аль аргых болохыг харуулна.

Кодын үр дүн:



Үр дүнгээ нэгтгэн тайлбарлах, дүгнэхэд хамгийн тохиромжтой, энгийн арга нь ROC муруй болон үүндээр AUC бодож үзэх юм.

```
from sklearn.metrics import auc

print("Area Under Roc Curve (AUROC)")
print(f"Common Neigh.: \t {auc(cn_evaluation.fallout(),
cn_evaluation.recall())}")
print(f"Adamic Adar: \t {auc(aa_evaluation.fallout(),
aa_evaluation.recall())}")
print(f"Jaccard: \t {auc(jc_evaluation.fallout(),
jc_evaluation.recall())}")
```

Тайлбар:

- auc(...) – ROC curve доорх талбайг тооцно.
- AUROC 0.5-аас их бол таамаглал нь random-с илүү сайн.
- Их байх тусам сайн таамаглаж байна гэсэн үг.

Кодын үр дүн:

```
Area Under Roc Curve (AUROC)
Common Neigh.:      0.015376333109730609
Adamic Adar:        0.016201171969940018
Jaccard:             0.01430571353574241
```

Даалгавар:

1. Jaccard болон Adamic Adar хоёр яагаад ялгаатай үр дүн үзүүлж байна вэ?

Тайлбарлана уу.

- а. Jaccard болон Adamic-Adar нь хоёулаа link prediction-д хэрэглэгддэг төвлөрсөн бус (local) аргачлалууд боловч тэдний тооцоолох арга өөр өөр учраас үр дүн нь ялгаатай байдаг.

Шинж	Jaccard	Adamic-Adar
Үндсэн санаа	Хоёр орой (node)-н нийт хөршүүд дотроос хамтарсан хөршийн харьцаа	Хамтарсан хөршийн тоо, гэхдээ ховор хөршүүдэд илүү жин өгдөг
Онцлог	Хамтарсан хөршийн тоо болон хэмжээний харьцаанд суурилдаг	Хамтарсан хөрш нь ховор бол илүү чухал гэж үздэг
Ялгаа	Хэрвээ хоёр орой олон нийтлэг хөрштэй бол оноо өндөр, гэхдээ бүх хөрш адил жинтэй	Хэрвээ нийтлэг хөрш нь бага холболттой бол илүү өндөр жин өгдөг — энэ нь “чанартай” хөрш гэж үздэг

2. linkpred санг ашиглан decision tree алгоритмаар холбоосыг илрүүлж болох уу.

- а. Үгүй. linkpred сан дотор Decision Tree алгоритм бүрэлдэхүүнд байхгүй.

linkpred юу ашигладаг вэ?

- Common Neighbors
- Jaccard
- Adamic-Adar
- Resource Allocation
- Preferential Attachment
- Katz
- Rooted PageRank гэх мэт

Хэрвээ Decision Tree ашиглахыг хүсвэл

1. Өөрөө dataset бэлдэж:

- Орой хос бүрийг нэг мөр болгож
- Граф дотор байгаа эсэхийг “target” болгож
- Хөршүүдийн тоо, зэрэг (degree), Jaccard score, Adamic-Adar score гэх мэт онцлог (features)-ийг бэлдэж болно.

2. sklearn.tree.DecisionTreeClassifier ашиглан supervised learning хийж болно.

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```