データ解析特論 第 10 回
ランキングとその評価

201720690 小松 弘人

2018/01/07

1　課題

manaba から共著関係ネットワークのデータをダウンロードし、そのネットワークからその分野における権威を見つける。このデータには、apscoauthor.csv と citation_count.txt の 2 ファイルが含まれている。apscoauthor.csv は、物理学の分野の論文の共著関係ネットワークをエッジリストで表現したものである。また、citation_count.txt は、論文の被引用数であり、このデータをこの分野における権威度を表す正解データとして用いる。

今回の課題では、共著関係ネットワークにおけるノードの次数および PageRank によって著者をランキングする。次に、このランキングを以下の方法を用いて評価を行う。

1. 被引用数に基づくランキングと共著ネットワークに基づくランキングの順位相関係数を求める
2. 被引用数上位 1% の著者をこの分野の権威であると考え、これらの著者を正解データとみなす。その場合の、Precision-Recall 曲線を plot する
3. 被引用数をその著者の適合度とみなし、nDCG@100 を計算する

2　ランキング

著者のランキングは、各ノードに対して次数中心性 (degree)、固有ベクトル中心性 (evcent)、PageRank (page.rank) を計算し、sort 関数および rank 関数を用いて行う。

3　結果

3.1　順位相関係数

順位相関係数には、スピアマンの順位相関係数 (式 1) とケンドールの順位相関係数 (式 2) が存在する。

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}}, \bar{x} = \bar{y} = \frac{1+N}{2} \tag{1}$$

$$\tau = \frac{2P}{\frac{1}{2}N(N-1)} - 1 \tag{2}$$

x_i, y_i はデータ i の順位、P は 2 つのランキングで順序関係が一致しているデータ組の数、N はデータ数を示す。

それぞれのランキングについて順位相関係数を計算した結果を表 1 に示す。

表 1: 順位相関係数

順位相関係数	次数中心性	固有ベクトル中心性	PageRank
スピアマンの順位相関係数	0.5819306	0.2700521	0.5887235
ケンドールの順位相関係数	0.4520609	0.1905424	0.4348009

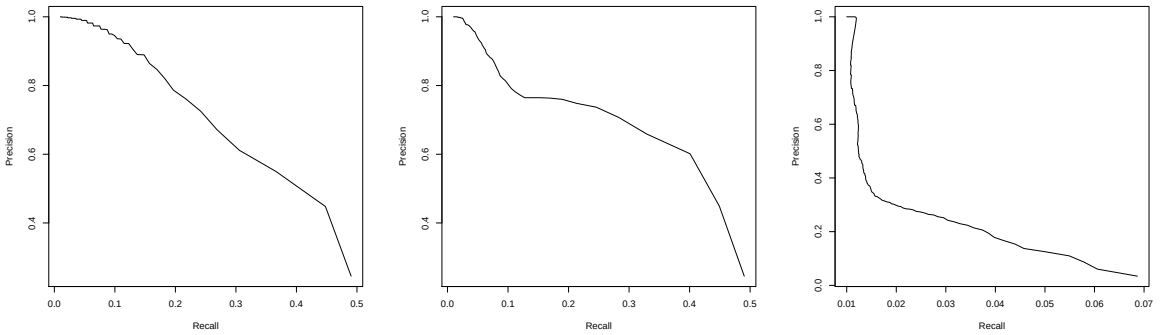
3.2 Precision-Recall 曲線

被引用数 (citation_count.txt) の上位 1% の著者を正解データとみなし、Precision (式 3)、Recall (式 4) を計算し、Precision-Recall 曲線を描画した。

$$P = \frac{|A \cap B|}{|B|} \tag{3}$$

$$R = \frac{|A \cap B|}{|A|} \tag{4}$$

Precision-Recall 曲線を図 1 に示す。



(a) 次数中心性　(b) 固有ベクトル中心性　(c) PageRank

図 1: Precision-Recall 曲線

3.3 nDCG@100

それぞれのランキングについて nDCG (式 5) を計算した。適合度は citation_count.txt の値を用いる。結果を表 2 に示す。

$$nDCG@k = \frac{\sum_{r=1}^{k} \frac{g(r)}{\log(r+1)}}{\sum_{r=1}^{k} \frac{g^{\star}(r)}{\log r+1}} \tag{5}$$

表 2: nDCG@100

次数中心性	固有ベクトル中心性	PageRank
0.2757818	0.2574463	0.2529339

4 考察

それぞれの評価指標における各ランキングの評価をまとめると、スピアマンの順位相関係数では PageRank、次数中心性、固有ベクトル中心性、ケンドールの順位相関係数では次数中心性、PageRank、

固有ベクトル中心性の順に評価が高かった。また、Precision-Recall 曲線を見ると、原点周りの面積が大きいのは次数中心性、固有ベクトル中心性、PageRank の順だった。nDCG@100 では、次数中心性、固有ベクトル中心性、PageRank の順であった。これらの評価指標を統合して考えると、次数中心性を基にランキングした結果が今回の問題においては最も正解に近いランキングを行えていたことが分かった。

PageRank と固有ベクトル中心性については、順位相関係数においては PageRank の方が大きく、Precision-Recall 曲線および nDCG@100 は固有ベクトル中心性の方が良いことから、ランキング全体の一致度は PageRank でランキングを行った場合の方が高いが、ランキング上位のデータの適合度については固有ベクトル中心性の方が良いという結果になった。

5 スクリプト

```
library(igraph)

# データの読み込み
g = read.graph('data/apscoauthor.csv')
answer = read.table('data/citation_count.txt')

# ランキング
## 次数中心性の計算
d = degree(g)
## 固有ベクトル中心性の計算
e = evcent(g)$vector
## の計算PageRank
p = page.rank(g)$vector

## ランキングの作成
ans = sort(answer$V2, decreasing = TRUE, method = "s", index.return = TRUE)
ans_rnk = rank(-answer$V2)
res_d = sort(d, decreasing = TRUE, method = "s", index.return = TRUE)
res_d_rnk = rank(-d)
res_e = sort(e, decreasing = TRUE, method = "s", index.return = TRUE)
res_e_rnk = rank(-e)
res_p = sort(p, decreasing = TRUE, method = "s", index.return = TRUE)
res_p_rnk = rank(-p)

## 順位相関係数
### スピアマン
rho_d = (cor.test(ans_rnk, res_d_rnk, method="s"))$estimate; rho_d
rho_e = (cor.test(ans_rnk, res_e_rnk, method="s"))$estimate; rho_e
rho_p = (cor.test(ans_rnk, res_p_rnk, method="s"))$estimate; rho_p

### ケンドール
tau_d = (cor.test(ans_rnk, res_d_rnk, method="k"))$estimate; tau_d
tau_e = (cor.test(ans_rnk, res_e_rnk, method="k"))$estimate; tau_e
tau_p = (cor.test(ans_rnk, res_p_rnk, method="k"))$estimate; tau_p

## precision--recall curve
A = ans$ix[1:(length(ans$ix)*0.01)]
seg = 200

P_d = c(); R_d = c()
for (i in 0:seg) {
    q = 100*i/seg
    B = res_d$ix[1:(length(ans$ix)*q/100)]
```

```
    AandB = intersect(A, B)
    P_d[i] = length(AandB) / length(B)
    R_d[i] = length(AandB) / length(A)
}
plot(P_d, R_d, type="l", xlab="Recall", ylab="Precision")

P_e = c(); R_e = c()
for (i in 0:seg) {
    q = 100*i/seg
    B = res_e$ix[1:(length(ans$ix)*q/100)]
    AandB = intersect(A, B)
    P_e[i] = length(AandB) / length(B)
    R_e[i] = length(AandB) / length(A)
}
plot(P_e, R_e, type="l", xlab="Recall", ylab="Precision")

P_p = c(); R_p = c()
for (i in 0:seg) {
    q = 100*i/seg
    B = res_p$ix[1:(length(ans$ix)*q/100)]
    AandB = intersect(A, B)
    P_p[i] = length(AandB) / length(B)
    R_p[i] = length(AandB) / length(A)
}
plot(P_p, R_p, type="l", xlab="Recall", ylab="Precision")

## nDCG@100
k = 100

### 理想的なランキングにおける値
isum = 0
for(r in 1:k) {
    isum = isum + ans$x[r] / log(r+1)
}

dsum = 0
for(r in 1:k) {
    dsum = dsum + answer$V2[res_d$ix[r]] / log(r+1)
}

esum = 0
for(r in 1:k) {
    esum = esum + answer$V2[res_e$ix[r]] / log(r+1)
}

psum = 0
for(r in 1:k) {
    psum = psum + answer$V2[res_p$ix[r]] / log(r+1)
}

(dsum/isum)
(esum/isum)
(psum/isum)
```