
# مبانی ریاضی علوم‌داده

# گزارش پروژه عملی سوم

**پوریا صامتی**

**۴۰۴۰۹۷۱۴**

زمستان ۱۴۰۴

## فهرست مطالب

# گام صفر-آماده‌سازی و شناخت داده

در این پروژه دارای سه مجموعه‌داده از جنس گراف هستیم. این سه مجموعه‌داده عبارتند از:

- **زیرگراف A**
- **زیرگراف B**
- **گراف کامل WikiVote**

در ادامه نسخه‌های مختلف الگوریتم PageRank با توجه به این ۳ مجموعه‌داده اعمال خواهند شد. ما قصد داریم تا این سه مجموعه را به عنوان گراف ورودی الگوریتم در نظر بگیریم. بدین جهت لازم است که در ابتدا با توزیع رئوس این سه گراف آشنا شویم و بدانیم با چه حجمی از رئوس و یال‌ها مواجه هستیم.

## خلاصه آمار گراف WikiVote

در جدول زیر تعداد رئوس و یال‌های گراف WikiVote را مشاهده می‌کنید:

```
Number of nodes: 8298
Number of edges: 103689
```

حال توزیع درجه ورودی و خروجی گراف WikiVote را مشاهده می‌کنید:

| | max | mean | median | zero_percent |
|---|---|---|---|---|
| In-degree | 457 | 12.495662 | 0.0 | 71.306339 |
| Out-degree | 893 | 12.495662 | 1.0 | 26.367799 |

حال تعداد گره‌هایی را مشاهده می‌کنید که درجه خروجی آنها ۰ است که به آن‌ها Dangling Node یا Deadend گفته می‌شود.

```
Dangling nodes: 2188
Dangling percent: 26.367799469751745 %
```

در مرحله بعد تعداد Self Loop بررسی می‌شود. یعنی آیا مبدا و مقصد گرهی یکسان است یا خیر. در گراف A این مورد مشاهده نمی‌شود.

```
Number of self-loops: 0
```

با توجه به ذات داده‌های **WikiVot**، رأی دادن به خود **معنای واقعی ندارد.** به همین دلیل Self-Loop توزیع PageRankرا به‌صورت مصنوعی منحرف می‌کند و همچنین تفسیر گرافی ندارند. به همین دلیل قبل از تنظیم آن به عنوان ورودی الگوریتم، آنها را **حذف** می‌کنیم.

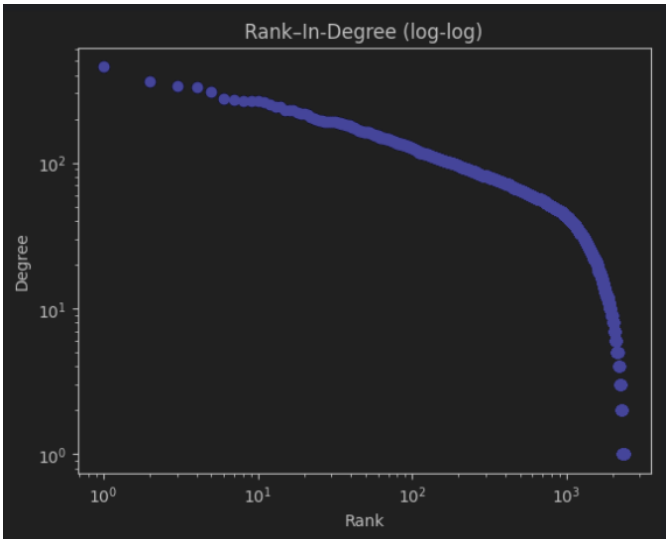
در جدول زیر **۲۰ گره برتر در تعداد درجه ورودی** مشاهده می‌شوند:

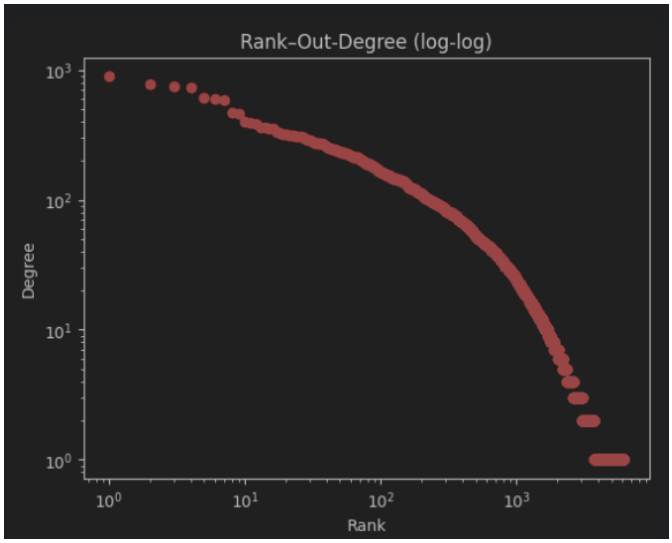
| | Node | Degree |
|---|---|---|
| 0 | 4037 | 457 |
| 1 | 15 | 361 |
| 2 | 2398 | 340 |
| 3 | 2625 | 331 |
| 4 | 1297 | 309 |
| 5 | 2565 | 274 |
| 6 | 762 | 272 |
| 7 | 2328 | 266 |
| 8 | 5254 | 265 |
| 9 | 3352 | 264 |
| 10 | 4191 | 259 |
| 11 | 2066 | 254 |
| 12 | 1549 | 245 |
| 13 | 3089 | 244 |
| 14 | 2535 | 232 |
| 15 | 737 | 231 |
| 16 | 4335 | 228 |
| 17 | 3456 | 223 |
| 18 | 5412 | 219 |
| 19 | 3334 | 217 |

در جدول زیر ۲۰ گره برتر در تعداد **درجه خروجی** مشاهده می‌شوند:

| | Node | Degree |
|---|---|---|
| 0 | 2565 | 893 |
| 1 | 766 | 773 |
| 2 | 11 | 743 |
| 3 | 457 | 732 |
| 4 | 2688 | 618 |
| 5 | 1166 | 599 |
| 6 | 1549 | 587 |
| 7 | 1151 | 472 |
| 8 | 1374 | 462 |
| 9 | 1133 | 399 |
| 10 | 5524 | 389 |
| 11 | 5802 | 385 |
| 12 | 3642 | 362 |
| 13 | 4967 | 361 |
| 14 | 2972 | 357 |
| 15 | 1608 | 355 |
| 16 | 173 | 334 |
| 17 | 2485 | 324 |
| 18 | 311 | 321 |
| 19 | 3453 | 319 |

نمودار Log-Log برای این درجه ورودی و درجه خروجی به شرح زیر است:



# گام اول-PageRank نسخه Dense روی دو زیر گراف کوچک از WikiVote

## مقدمه

حال وقت آن است که از دو زیرگرافی که در مرحله قبل یاد شد استفاده کنیم. در ابتدا داده‌های هر دو گراف را تحلیل می‌کنیم. هدف این بخش آن است که توزیع رئوس و یال‌های این دو گراف را بدست آوریم. در ابتدا با اطلاعات کلی هر گراف آشنا می‌شویم. این اطلاعات در جدول زیر موجود هستند:

| | Subgraph | Nodes | Edges | Self_Loops | Deadends | Deadend_percent |
|---|---|---|---|---|---|---|
| 0 | A | 500 | 12691 | 0 | 128 | 25.6 |
| 1 | B | 500 | 14951 | 0 | 0 | 0.0 |

توزیع رئوس و یال‌های زیرگراف A به شکل جدول زیر است:

| | max | mean | median | zero_percent |
|---|---|---|---|---|
| In-degree | 95 | 25.382 | 22.0 | 0.0 |
| Out-degree | 499 | 25.382 | 7.0 | 25.6 |

توزیع رئوس و یال‌های زیرگراف B به شکل جدول زیر است:

| | max | mean | median | zero_percent |
|---|---|---|---|---|
| In-degree | 169 | 29.902 | 27.5 | 36.0 |
| Out-degree | 178 | 29.902 | 23.0 | 0.0 |

سپس با استفاده از داده‌های استخراج شده وارد مرحله بعدی می‌شویم. در این مرحله ماتریس همسایگی هر زیرگراف را بر اساس یال‌های زیرگراف‌های A و B می‌سازیم. پس از آن نوبت به ساخت Transition Matrix برای زیرگراف‌های A و B می‌شود. حال داده‌های لازم برای الگوریتم PageRank را در اختیار داریم.

## اجرای PageRank – نسخه پایه‌ای

طبق توضیحات بیان شده در بخش ۳-۱ الگوریتم را پیاده‌سازی می‌کنیم. توجه شود که در این نسخه Deadend از بین نرفته‌اند و داده‌ها را بدون پردازش خاصی به الگوریتم فوق ارسال کرده‌ایم. طبق توضیحات سوال این الگوریتم را بر هر دو مجموعه A و B اعمال کرده‌ایم. نتایج اجرای الگوریتم به شرح زیر است:

| | subgraph | iterations | pagerank_value_sum | last_residual |
|---|---|---|---|---|
| 0 | A | 49 | 0.435101 | 7.511420e-10 |
| 1 | B | 41 | 1.000000 | 6.293527e-10 |

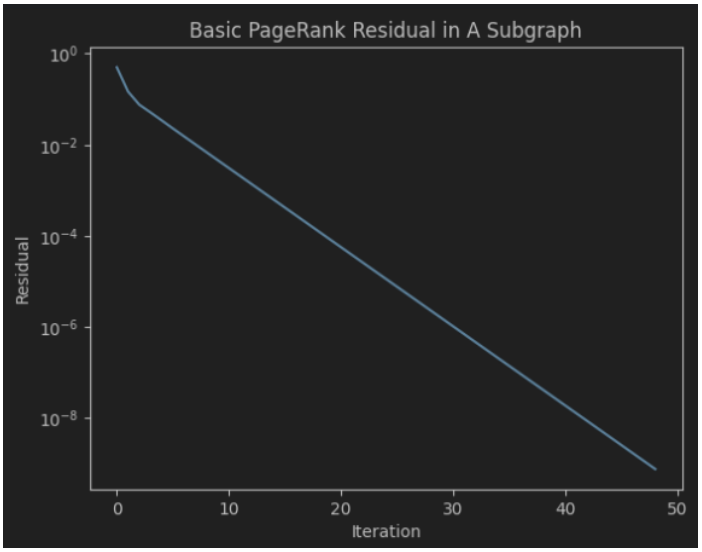
حال می‌خواهیم بررسی دقیق‌تری بر نتایج اعمال الگوریتم بر هر زیرگراف داشته باشیم. در ابتدا زیرگراف A را بررسی می‌کنیم. این الگوریتم به ۱۰ گره زیر بیشترین امتیاز را در زیرگراف A داده است. این گره‌ها به شرح زیر هستند:

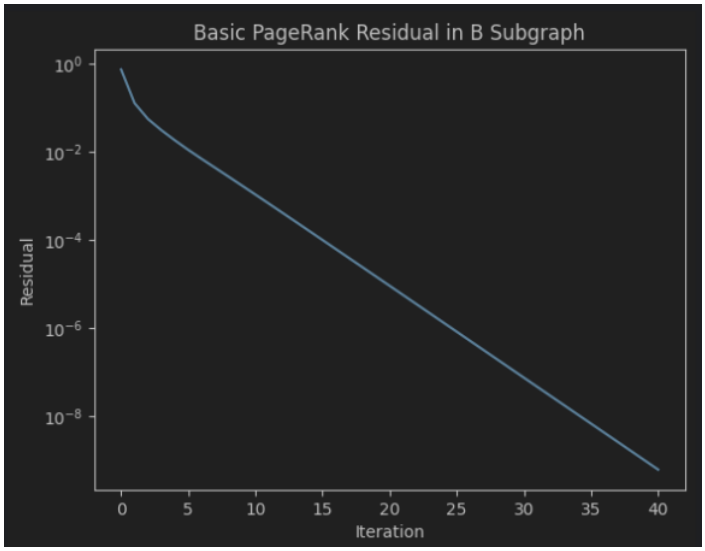
| | Rank | Node | Score |
|---|---|---|---|
| 0 | 1 | 168 | 0.003736 |
| 1 | 2 | 88 | 0.003162 |
| 2 | 3 | 132 | 0.003141 |
| 3 | 4 | 322 | 0.003017 |
| 4 | 5 | 338 | 0.002998 |
| 5 | 6 | 121 | 0.002748 |
| 6 | 7 | 448 | 0.002704 |
| 7 | 8 | 468 | 0.002619 |
| 8 | 9 | 426 | 0.002438 |
| 9 | 10 | 498 | 0.002422 |

همچنین برای زیرگراف B داریم که:

| | Rank | Node | Score |
|---|---|---|---|
| 0 | 1 | 250 | 0.012321 |
| 1 | 2 | 265 | 0.011402 |
| 2 | 3 | 181 | 0.010663 |
| 3 | 4 | 318 | 0.010038 |
| 4 | 5 | 448 | 0.009008 |
| 5 | 6 | 336 | 0.008834 |
| 6 | 7 | 157 | 0.008833 |
| 7 | 8 | 242 | 0.008756 |
| 8 | 9 | 333 | 0.008525 |
| 9 | 10 | 92 | 0.008515 |

نمودار همگرایی خطا بر حسب تکرار برای این دو زیرگراف به شرح زیر است:




همانطور که مشاهده کردیم، مشکل اساسی در زیرگراف A رخ داده است. مجموع امتیازات page Rank در این زیرگراف برابر 0.43 شده است. ما میدانیم که بعد از همگرایی الگوریتم، این مجموع امتیاز باید برابر 1 باشد که در اینجا بدین شکل نیست.

## اجرای Page Rank – نسخه رفع نشتی Deadend

در این نسخه سعی بر آن کردیم که مانع از مشکل بوجود آمده در نسخه قبلی بشویم. در این نسخه در هر تکرار مجموع امتیازات را مجبور می‌کنیم که برابر ۱ باشد. نتایج اجرای الگوریتم به شرح جدول زیر است:

| | subgraph | iterations | pagerank_value_sum | last_residual |
|---|---|---|---|---|
| 0 | A | 18 | 1.0 | 4.176660e-10 |
| 1 | B | 41 | 1.0 | 6.293527e-10 |

تغییرات موجود در این نسخه از الگورتیم **مشکل مجموع امتیاز را حل** کرد. اما دو اتفاق مثبت دیگر نیز رخ داده است:

- **تعداد تکرار الگوریتم**: طبق این مشاهده استفاده از نسخه رفع نشتی باعث شده است که الگوریتم در هر دو زیرگراف با تعداد تکرار کمتری نسبت به الگوریتم حالت پایه همگرا شود.
- **میزان خطا**: خطا در آخرین تکرار نیز با استفاده نسخه رفع‌نشتی کاهش داشته است. مخصوصا در زیرگراف A.

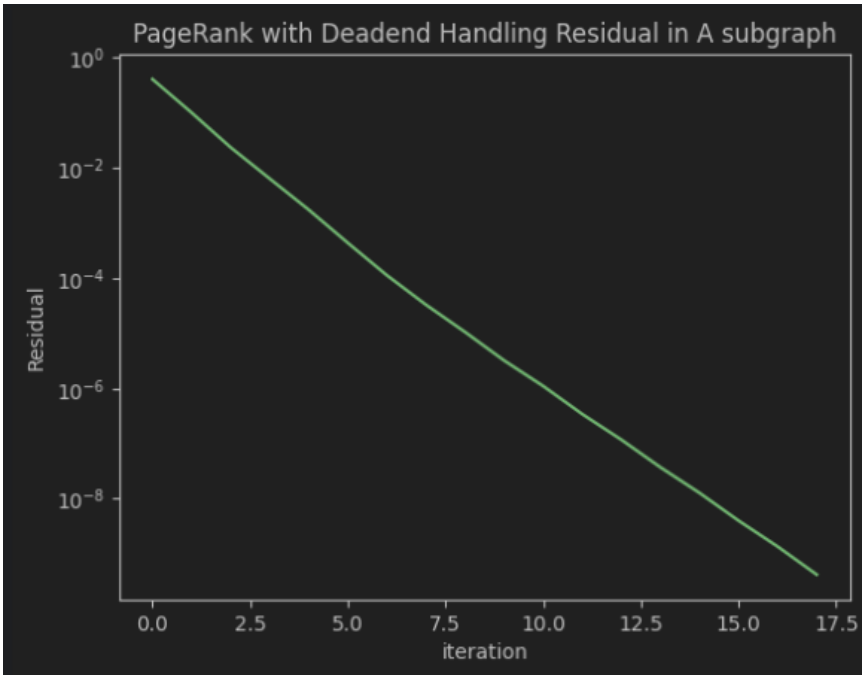
این الگوریتم به ۱۰ گره زیر بیشترین امتیاز را در زیرگراف A داده است:

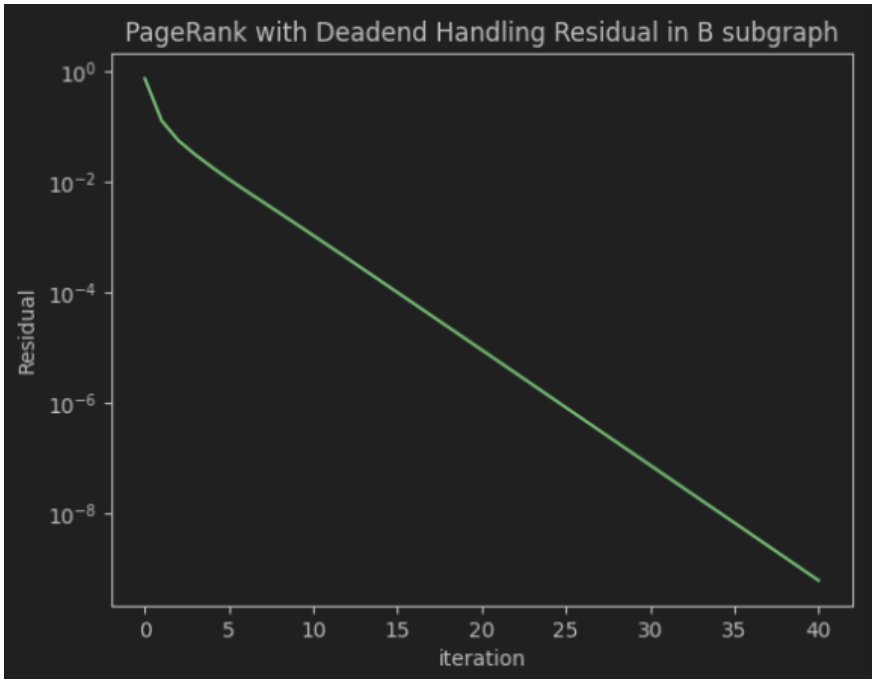
| | Rank | Node | Score |
|---|---|---|---|
| 0 | 1 | 168 | 0.003736 |
| 1 | 2 | 88 | 0.003162 |
| 2 | 3 | 132 | 0.003141 |
| 3 | 4 | 322 | 0.003017 |
| 4 | 5 | 338 | 0.002998 |
| 5 | 6 | 121 | 0.002748 |
| 6 | 7 | 448 | 0.002704 |
| 7 | 8 | 468 | 0.002619 |
| 8 | 9 | 426 | 0.002438 |
| 9 | 10 | 498 | 0.002422 |

این الگوریتم به ۱۰ گره زیر بیشترین امتیاز را در زیرگراف B داده است:

| | Rank | Node | Score |
|---|---|---|---|
| 0 | 1 | 250 | 0.012321 |
| 1 | 2 | 265 | 0.011402 |
| 2 | 3 | 181 | 0.010663 |
| 3 | 4 | 318 | 0.010038 |
| 4 | 5 | 448 | 0.009008 |
| 5 | 6 | 336 | 0.008834 |
| 6 | 7 | 157 | 0.008833 |
| 7 | 8 | 242 | 0.008756 |
| 8 | 9 | 333 | 0.008525 |
| 9 | 10 | 92 | 0.008515 |

نمودار همگرایی خطا بر حسب تکرار برای این دو زیرگراف در الگوریتم PageRank بهبود یافته به شرح زیر است:



## سوالات تحلیلی

### ۱-چرا در زیرگراف A در نسخه Basic مقدار $\sum_i pi$ کمتر از ۱ شده است؟

همانطور که مشاهده کردیم در زیرگراف A ما با راس‌هایی مواجه بودیم که Deadend بودند. یعنی از این رئوس یالی خارج نمیشد و درجه خروجی آن‌ها 0 بود. به همین دلیل در حین اجرای الگوریتم Page Rank اصلی، در حین اجرای الگوریتم نشت رخ می‌دهد. این نشت باعث می‌شود که مجموع امتیازها برابر 1 نشود. اما در **نسخه Fix** الگوریتم ما بصورت اجباری در حین اجرای الگوریتم **با استفاده از نرمال‌سازی** بردار امتیازات این نشت را جبران می‌کنیم و مجموع امتیازات را همواره ۱حفظ می‌کنیم..

### ۲-چرا در زیرگراف B در نسخه Basic و نسخه اصلاح شده تقریبا یکسان هستند؟

همانطور که مشاهده کردیم در زیرگراف B ما راس Deadend نداریم. به همین دلیل هیچ نشتی در حین اجرای الگوریتم رخ نمیدهد یا نشت بسیار بسیار جزئی می‌باشد. به همین دلیل تفاوت چندانی بین نسخه پایه و نسخه fix برای این زیرگراف مشاهده نمی‌شود.

### ۳-اگر $\epsilon$ را از $10^{-9}$ به $10^{-8}$ تغییر دهیم، Top-10 چقدر عوض می‌شود؟ چرا؟

در نتیجه Top-10 تغییر محسوسی دیده نمی‌شود. اما در میزان Residual و تعداد تکرار تا همگرایی تغییرات محسوس هستند. جدول زیر حاصل از اعمال PageRank با $\epsilon = 10^{-8}$ است:

| | subgraph | iterations | pagerank_value_sum | last_residual |
|---|---|---|---|---|
| 0 | A | 16 | 1.0 | 3.977144e-09 |
| 1 | B | 36 | 1.0 | 6.906224e-09 |

# گام دوم-PageRank نسخه Sparse

## مقدمه

طبق خواسته سوال، ابتدا ماتریس CSR از روی لیست یال‌ها ساخته می‌شود. سپس ماتریس انتقال را با استفاده از **نرمال‌سازی سطری** نرمال می‌کنیم. حال باید تعداد گره‌های Deadend را طبق خواسته سوال اعلام کنیم. این تعداد به شرح زیر است:

```
Deadends for A_csr 128
Deadends for B_csr 0
```

حال مقدمات لازم برای اعمال الگوریتم Page Rank نسخه Sparse موجود است. در ادامه به اجرای این الگوریتم و نتایج آن می‌پردازیم.

## اجرای الگوریتم PageRank نسخه Sparse

این الگوریتم برای هر دو زیرگراف A و B ابتدا اعمال می‌شود. شرایط دو زیرگراف به شرح زیر است:

| | Dataset | Nodes | Edges | NNZ | Build_Time |
|---|---|---|---|---|---|
| 0 | A | 500 | 12691 | 12691 | 0.00000 |
| 1 | B | 500 | 14951 | 14951 | 0.00099 |

حال الگوریتم مذکور را اعمال می‌کنیم. نتایج اجرا به شرح زیر هستند:

| | dataset | iterations | last_residual | sum_pagerank_values | converge_time | total_time |
|---|---|---|---|---|---|---|
| 0 | A | 49 | 7.511420e-10 | 0.435101 | 0.003003 | 0.003003 |
| 1 | B | 41 | 6.293526e-10 | 1.000000 | 0.002000 | 0.004011 |

نکته قابل ملاحظه این الگوریتم آن است که با سرعت بسیار بالایی این محاسبات را انجام می‌دهد و برای مواقعی که گراف بسیار بزرگ است بسیار مورد استفاده خواهد بود. زیرا می‌تواند با تکیه بر خلوت نگه داشتن ماتریس، محاسبات را در زمان کوتاه‌تری انجام دهد.

## اجرای الگوریتم PageRank نسخه Sparse بر روی گراف کامل

حال برای بررسی زمان محاسبه، این الگوریتم را بر روی گراف کامل WikiVote اعمال می‌کنیم تا ببینیم که زمان محاسبات با این گراف سنگین چقدر خواهد بود. در ابتدا مجموعه‌داده ورودی را بررسی میکنیم. این بررسی به جدول زیر منتج شده است:

| | Dataset | Nodes | Edges | NNZ | Build_Time |
|---|---|---|---|---|---|
| 0 | WikiVote | 7115 | 103689 | 12691 | 0.001012 |

حال این الگوریتم را اعمال می‌کنیم. نتایج اجرا به شرح زیر است:

| | dataset | iterations | last_residual | sum_pagerank_values | converge_time | total_time |
|---|---|---|---|---|---|---|
| 0 | WikiVote | 42 | 8.708277e-10 | 0.170035 | 0.003004 | 0.005018 |

زمان اجرا همچنان بسیار ناچیز است با اینکه این گراف نسبت به دو گراف قبلی دارای پیچیدگی بیشتر در تعداد گره‌ها و یال‌ها می‌باشد.

## سوالات تحلیلی

**۱-با استفاده از عددهای واقعی دیتاست WikiVote (تعداد نودها و تعداد یال‌ها) توضیح دهید چرا ساخت و نگهداری ماتریس Dense برای گراف کامل عملی نیست، اما نمایش Sparse (CSR) عملی است؟ جواب را با Big-O و هم با اعداد واقعی WikiVote توضیح دهید.**

در پیاده‌سازی Sparse PageRank، ماتریس انتقال $P$ به‌صورت **CSR** ذخیره می‌شود. در این ساختار، تنها عناصر غیرصفر (یال‌های موجود در گراف) نگه‌داری می‌شوند. پس پیچیدگی زمانی هر ضرب ماتریس-بردار برابر $O(nnz)$ است. که در آن nnz تعداد عناصر غیرصفر ماتریس (تقریباً برابر با تعداد یال‌ها) است. در مقابل، در نسخه‌ی *Dense* این پیچیدگی از مرتبه $O(n^2)$ است که برای گراف‌های واقعی با هزاران نود عملاً غیرقابل استفاده است.

پس از آنجا که گراف WikiVote به‌شدت تنک است($nnz \ll n^2$)، استفاده از CSR باعث کاهش چشمگیر زمان اجرای هر تکرار می‌شود و امکان اجرای PageRank روی گراف کامل را فراهم می‌کند.

**۲-اگر $\sum_i pi$ در Full Graph کمتر از ۱ شد، این نشانه چیست و چه چیزی در گراف باعث آن می‌شود؟**

کمتر شدن مجموع احتمال‌ها از ۱ نشان می‌دهد که در گراف نودهای Deadend وجود دارند و در PageRank این Deadend ها هنوز اصلاح نشده‌اند. این نودها هیچ یال خروجی ندارند و جرمی که به آن‌ها می‌رسد، در تکرار بعدی **منتقل نمی‌شود** در نتیجه بخشی از احتمال کل در هر تکرار خارج می‌شود که به آن **probability mass leakage** می‌گویند.

# گام سوم– Personalized PageRank + رفع مشکل dummy/Deadend

حال می‌خواهیم بطور عمیق و موثر هم الگوریتم PageRank را ارتقا دهیم و هم مشکل گره‌های Deadend را حل کنیم.

## بخش A–رفع مشکل Deadend (روی Full graph)

برای رفع مشکل گره‌های Deadend در الگوریتم PageRank دو راه وجود دارد:

- **استفاده ازQ-fix**: بازگرداندن نشت به بردار teleport شخصی‌سازی‌شده q.
- **استفاده از U-fix**: پخش یکنواخت نشت بین همه نودها.

حال می‌خواهیم نتیجه اعمال این دو روش را بر روی داده گراف کامل WikiVote بررسی کنیم. جدول زیر نتیجه اعمال Q-fix و U-Fix می‌باشد:

| | Methode | dataset | iterations | last_residual | sum_pagerank_values | converge_time | total_time |
|---|---|---|---|---|---|---|---|
| 0 | Q-Fix | Full Graph | 24 | 5.194719e-10 | 1.0 | 0.003008 | 0.005022 |
| 1 | U-Fix | Full Graph | 40 | 9.639756e-10 | 1.0 | 0.004004 | 0.006018 |

در جدول زیر ۲۰ گره برتر با استفاده از Q-fix در الگوریتم مشاهده می‌شوند:

| | Rank | Node | Score |
|---|---|---|---|
| 0 | 1 | 0 | 0.282787 |
| 1 | 2 | 171 | 0.046654 |
| 2 | 3 | 30 | 0.045736 |
| 3 | 4 | 13 | 0.043448 |
| 4 | 5 | 53 | 0.042683 |
| 5 | 6 | 118 | 0.042290 |
| 6 | 7 | 3 | 0.042020 |
| 7 | 8 | 88 | 0.009921 |
| 8 | 9 | 144 | 0.007171 |
| 9 | 10 | 420 | 0.006239 |
| 10 | 11 | 206 | 0.006087 |
| 11 | 12 | 168 | 0.005161 |
| 12 | 13 | 23 | 0.005086 |
| 13 | 14 | 29 | 0.004659 |
| 14 | 15 | 121 | 0.004046 |
| 15 | 16 | 149 | 0.004031 |
| 16 | 17 | 285 | 0.003570 |
| 17 | 18 | 173 | 0.003481 |
| 18 | 19 | 132 | 0.003455 |
| 19 | 20 | 5 | 0.003433 |

در جدول زیر ۲۰ گره برتر با استفاده از U-fix در الگوریتم مشاهده می‌شوند:

| | Rank | Node | Score |
|---|---|---|---|
| 0 | 1 | 168 | 0.001544 |
| 1 | 2 | 88 | 0.001307 |
| 2 | 3 | 132 | 0.001298 |
| 3 | 4 | 322 | 0.001247 |
| 4 | 5 | 338 | 0.001239 |
| 5 | 6 | 121 | 0.001136 |
| 6 | 7 | 448 | 0.001118 |
| 7 | 8 | 468 | 0.001082 |
| 8 | 9 | 426 | 0.001007 |
| 9 | 10 | 498 | 0.001001 |
| 10 | 11 | 35 | 0.000949 |
| 11 | 12 | 176 | 0.000943 |
| 12 | 13 | 149 | 0.000933 |
| 13 | 14 | 13 | 0.000911 |
| 14 | 15 | 12 | 0.000904 |
| 15 | 16 | 285 | 0.000893 |
| 16 | 17 | 53 | 0.000871 |
| 17 | 18 | 359 | 0.000851 |
| 18 | 19 | 278 | 0.000816 |
| 19 | 20 | 331 | 0.000812 |

این دو راه حل، باعث شدند تا گره‌های متفاوتی به عنوان پاسخ ثبت شود و هر کدام دارای تفاوت‌هایی در بیان گره‌های با امتیاز برتر هستند. با این وجود اشتراک پاسخ آنها و درصد این اشتراک را نیز بررسی کردیم. نتایج به شرح زیر است:

```
Overlap Percentage:  40.0
Overlapping Nodes:  [ 13  53  88 121 132 149 168 285]
```

## سوال تحلیلی

**۱-چرا با رفع Deadend میزان $\sum_i pi$ باید نزدیک ۱ شود؟**

با رفع Deadend دیگر نشت احتمال در هر تکرار رخ نمی‌دهد. به همین علت هیچ احتمالی از دست نرفته و جمع احتمال ۱ باقی خواهد ماند. این جرم احتمالی:

- در **q-fix** به بردار teleport شخصی‌سازی‌شده q بازگردانده می‌شود.
- در **u-fix** به‌صورت یکنواخت بین همه نودها پخش می‌شود.

## بخش B-Personalized PageRank (کاربردی بر روی FullGraph)

نتیجه اعمال این روش به شرح جدول زیر است:

| | Methode | dataset | iterations | last_residual | sum_pagerank_values | converge_time |
|---|---|---|---|---|---|---|
| 0 | personalized pagerank | Full Graph | 1 | 0.0 | 1.0 | 0.0 |

همچنین طبق این روش گره‌های زیر به عنوان گره‌های برتر شناخته شده اند:

| | Rank | Node | Score |
|---|---|---|---|
| 0 | 1 | 4037 | 0.000124 |
| 1 | 2 | 0 | 0.000302 |
| 2 | 3 | 4749 | 0.000124 |
| 3 | 4 | 4748 | 0.000124 |
| 4 | 5 | 4747 | 0.000124 |
| 5 | 6 | 4746 | 0.000124 |
| 6 | 7 | 4745 | 0.000124 |
| 7 | 8 | 4744 | 0.000124 |
| 8 | 9 | 4743 | 0.000124 |
| 9 | 10 | 4742 | 0.000124 |
| 10 | 11 | 4741 | 0.000124 |
| 11 | 12 | 4740 | 0.000124 |
| 12 | 13 | 4750 | 0.000124 |
| 13 | 14 | 4739 | 0.000124 |
| 14 | 15 | 4737 | 0.000124 |
| 15 | 16 | 4736 | 0.000124 |
| 16 | 17 | 4735 | 0.000124 |
| 17 | 18 | 4734 | 0.000124 |
| 18 | 19 | 4733 | 0.000124 |
| 19 | 20 | 4732 | 0.000124 |

## سوالات تحلیلی

### ۱-PPR در WikiVote چه نوع «اطرافیان مرتبط» را برجسته می‌کند؟

Personalized PageRank در گراف WikiVote ارتباطاتی را برجسته می‌کند که از نظر ساختاری و مسیری به seed انتخاب‌شده نزدیک‌تر هستند .برخلاف PageRank معمولی که اهمیت کلی و جهانی کاربران را نشان می‌دهد، این الگوریتم بر نودهایی تمرکز دارد که یا مستقیماً به seed رأی داده‌اند، یا در زنجیره‌های کوتاه رأی‌دهی به آن متصل‌اند. بنابراین این روش بیشتر روابط محلی، اجتماعی و جامعه‌محور پیرامون seed را برجسته می‌کند، نه لزوماً کاربران با رأی‌های زیاد در کل شبکه.

### ۲-چرا ممکن است بعضی از نودها عملا به seed ربطی نگیرند؟ (حس Disconnected)

در WikiVote ممکن است برخی نودها از نظر ساختاری به seed **هیچ مسیر جهت‌داری** نداشته باشند یا تنها از طریق مسیرهای بسیار طولانی و ضعیف متصل شوند. از آنجا که این روش با احتمال restart بالا (مثلاً ۰.۱۵) دائماً به seed بازمی‌گردد، جرم احتمالی فرصت کافی برای رسیدن به این نودهای دور یا قطع‌شده را پیدا نمی‌کند. در نتیجه، این نودها عملاً PageRank نزدیک به صفر می‌گیرند و رفتار آن‌ها مشابه نودهای disconnected ظاهر می‌شود.

## سوالات مفهومی

**۱-در WikiVote که یال A→B (یعنی A به B رأی داده)، PageRankبرای B دقیقاً چه تعبیر اجتماعی دارد؟**

در این گراف، PageRank بالای یک گره مثلB نشان‌دهنده‌ی آن است که کاربران مهم و تأثیرگذار زیادی به B رأی داده‌اند .چون رأی‌دهنده‌ها خودشان وزن دارند، رأی یک کاربر معتبر ارزش بیشتری دارد. پس PageRank در WikiVote را می‌توان به‌عنوان معیاری برای مقبولیت یا اعتبار تعبیر کرد، نه صرفاً تعداد رأی‌ها.

**۲-چگونه ممکن است کسی با رأی‌های کمتر PageRank بالاتری بگیرد؟**

PageRank به کیفیت رأی‌دهندگان حساس است، نه فقط به تعداد آن‌ها. اگر کاربری رأی‌های کمی دریافت کند اما این رأی‌ها از طرف کاربرانی با PageRank بالا باشد، آنگاه آن فرد هم اهمیت بالایی دارد. یعنی اهمیت او می‌تواند از فردی با رأی‌های زیاد ولی از کاربران کم‌اهیمت بیشتر شود. بنابراین روش PageRank نظر افراد بانفوذ ارزش بیشتری نسبت به رأی کاربران حاشیه‌ای دارد.

**۳-چرا اگر دو کاربر به هم رأی بدهند (دور بسته)، PageRank هر دو بالا می‌رود؟**

وقتی دو یا چند کاربر یک حلقه تشکیل دهند، آنگاه جرم احتمالی بین آن‌ها به‌صورت مداوم بازتوزیع می‌شود و از سیستم خارج نمی‌شود. این پدیده باعث **تقویت متقابل** می‌شود. اگر این حلقه به سایر بخش‌های گراف نیز متصل باشد، اثر آن تشدید می‌شود. به همین دلیل PageRank به ساختارهای چرخه‌ای حساس است و آن‌ها را به‌عنوان نواحی پایدار و مهم شناسایی می‌کند.

**۴-نقش restart = 0.15 چیست و چرا این مقدار رایج است؟**

ما می‌دانیم که وب‌گراف یک گراف همبند نیست. به همین دلیل اصلا اعمال PageRank بر آن هیچ توضیحی ندارد. به همین جهت از فرآیند Restart استفاده می‌شود. این فرآیند رفتاری ایجاد میکند که گویا گراف همبند است. مخصوصا با توجه به گره‌های Deadend این موضوع اهمیتی دوچندان پیدا می‌کند. همچنین جلوگیری از گیر افتادن در deadendها و تضمین همگرایی نیز از کاربردهای این فرآیند هستند. مقدار 0.15 به‌صورت تجربی نشان داده که تعادل مناسبی بین این عوامل ایجاد می‌کند و به همین دلیل به مقدار استاندارد تبدیل شده است.

**۵-اگر فقط یک معیار برای «مهم بودن» انتخاب کنیم، کدام مناسب‌تر است؟ in-degree، out-degree، PPR و PageRank؟**

PageRank و PPR در WikiVote صرفاً ابزارهای عددی نیستند، بلکه مدل‌هایی برای تحلیل و ساختار قدرت در یک شبکه‌ی رأی‌دهی واقعی هستند. تفاوت نسخه‌ها دقیقاً به تفاوت بین نگاه «جهانی» و «محلی» به اعتبار کاربران برمی‌گردد.