Notes: Pagerank

Due: Sunday, 6 Sep 2020 at midnight

1 Pre-lecture Work

Problem 1. Matt Cutts was formerly the head of Google's web spam team, and now runs the United States Digital Service (a recently created branch of the US government). Watch his video on "How Google Search Works."

`https://www.youtube.com/watch?v=KyCYyoGusqs`

Problem 2. Our primary text for this week is *Deeper Inside Pagerank* by Langville and Meyers. Read sections 1, 2, 3, 5.1, 6.1, 6.2, 9, 10.

`https://galton.uchicago.edu/~lekheng/meetings/mathofranking/ref/langville.pdf`

Based on the reading, define the following terms:

1. markov chain

2. stationary vector

3. stochastic matrix

4. irreducible matrix

5. primitive matrix

6. aperiodic markov chain

7. $\mathbf{P}$

8. $\bar{\mathbf{P}}$

9. $\bar{\bar{\mathbf{P}}}$

10. $\boldsymbol{\pi}$

11. dangling nodes

12. personalization vector

Problem 3. Answer the following questions.

1. Is the matrix $\mathbf{P}$ stochastic? irriducible? primitive?

2. Is the matrix $\bar{\mathbf{P}}$ stochastic? irriducible? primitive?

3. Is the matrix $\bar{\bar{\mathbf{P}}}$ stochastic? irriducible? primitive?

Problem 4. Either prove or give a counterexample to the following claims.

1. $\mathrm{rank}(\bar{\bar{\mathbf{P}}}) = 1$.

2. $\mathrm{rank}(\bar{\bar{\mathbf{P}}}) = n$.

3. Let $\mathbf{x}$ be an eigenvector of $\mathbf{P}$ with eigenvalue λ. Then $\frac{1}{2}\mathbf{x}$ is also an eigenvector of $\mathbf{P}$ with eigenvalue $\frac{1}{2}\lambda$.

4. The smallest eigenvalue of $\bar{\bar{\mathbf{P}}}$ is exactly 0.

5. The largest eigenvalue of $\bar{\bar{\mathbf{P}}}$ is exactly 1.

6. The largest eigenvalue of $\mathbf{P}$ is exactly 1.

7. The largest eigenvector of $\mathbf{P}$ is simple. Recall that a simple eigenvalue has multiplicity 1. That is, there is exactly 1 eigenvector with the same eigenvalue.

8. The largest eigenvector of $\bar{\mathbf{P}}$ is simple.

9. The largest eigenvector of $\bar{\bar{\mathbf{P}}}$ is simple.

10. The eigenvectors of $\bar{\bar{\mathbf{P}}}$ are orthogonal.

11. The eigenvectors of $\bar{\bar{\mathbf{P}}}\bar{\bar{\mathbf{P}}}^T$ are orthogonal.

12. The eigenvectors of $\bar{\bar{\mathbf{P}}}^T\bar{\bar{\mathbf{P}}}$ are orthogonal.

2 Lecture

Problem 5. The purpose of the pagerank vector $\boldsymbol{\pi}$ is to provide a ranking of of how important a node is. There are many alternative ways to provide such a ranking. One simple alternative is to rank nodes by their in-degree. For "typical" graphs, the in-degree ranking and the pagerank ranking will be similar, but there are graphs for which the two rankings can be arbitrarily different from each other.

Draw a graph such that the top ranked node according to pagerank is the bottom ranked node according to in-degree.

Problem 6. The beginning of Section 5 shows the following equivalent definitions for the pagerank vector $\boldsymbol{\pi}$:

$$\boldsymbol{\pi}^T \bar{\bar{\mathbf{P}}} = \boldsymbol{\pi}^T \qquad \text{and} \qquad \boldsymbol{\pi}^T (\mathbf{I} - \bar{\bar{\mathbf{P}}}) = \mathbf{0}^T. \tag{1}$$

It should be obvious why these definitions of $\boldsymbol{\pi}$ are equivalent. Less obvious (and not shown in the paper) is that the following definition is also equivalent. Prove this equivalence.

$$\boldsymbol{\pi} = \operatorname*{arg\,max}_{\mathbf{w} \in \mathbb{R}^d, \|\mathbf{w}\|_2 \leq 1} \|\mathbf{w}^T \bar{\bar{\mathbf{P}}}\|_2 \tag{2}$$

Problem 7. In this question you will calculate the runtime of the power method for computing pagerank. Assume that P is a sparse matrix and that $\boldsymbol{\pi}$ is dense.

1. Equation 5.1 shows the power method iteration for solving for $\boldsymbol{\pi}$. It is reproduced below

$$\mathbf{x}^{(k)T} = \alpha \mathbf{x}^{(k-1)T} \mathbf{P} + (\alpha \mathbf{x}^{(k-1)T} \mathbf{a} + (1-\alpha)) \mathbf{v}^T. \tag{3}$$

   What is the runtime of computing $\mathbf{x}^{(k)}$ from $\mathbf{x}^{(k-1)}$?

2. Given only $\mathbf{x}^{(0)}$, what is the runtime of computing $\mathbf{x}^{(K)}$ by iterating Equation (3) K times?

3. When computing pagerank, we typically do not know the final number of iterations K in advance. Instead, we continue our computation until the following condition is met:

$$\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 \leq \epsilon, \tag{4}$$

where ϵ is a "small" number that controls how accurate we want our solution to be. Compute a formula for K as a function of ϵ.

4. Substitute your answer for part 3 into your answer for part 2 to get a formula for the overall runtime in terms of the final desired accuracy ϵ.

5. Now assume that $\bar{\bar{\mathbf{P}}}$ is stored as a dense matrix. Repeat the calculations for the runtime of the power method in terms of the desired accuracy ϵ.

6. We say that an algorithm for computing the pagerank *converges* if in the limit as the number of iterations goes to infinity, the algorithm returns the correct pagerank vector. Have we shown that the power method converges? Or are there conditions in which is will *diverge* (i.e. not converge)?

7. Why does it never make sense to store $\boldsymbol{\pi}$ as a sparse vector?

8. Prove the following statement or provide a counterexample.

$$\|\mathbf{x}^{(k)}\|_2 < \|\mathbf{x}^{(k-1)}\|_2. \tag{5}$$

9. Based on the results of part 8 above, how should we adjust our implementation of the power method?

Problem 8. There are many alternative algorithms for computing pagerank vectors. In this problem, we will investigate an algorithm that I call the *exponentially accelerated power method*, although it does not have a commonly accepted name. This is a divide and conquer algorithm that can achieve the same accuracy ϵ as the power method with only a logarithmic number of iterations.

The estimated pagerank vector is given by

$$\mathbf{y}^{(K)} = \mathbf{x}^{(0)}\mathbf{Q}_K, \tag{6}$$

where

$$\mathbf{Q}_k = \begin{cases} \bar{\bar{\mathbf{P}}} & \text{if } k = 0 \\ \mathbf{Q}_{k-1}\mathbf{Q}_{k-1} & \text{otherwise} \end{cases}. \tag{7}$$

In the standard power method, the matrix $\bar{\bar{\mathbf{P}}}$ is not stored explicitly, but is calculated from the $\mathbf{P}$ matrix. In this problem, you can assume for simplicity that the $\bar{\bar{\mathbf{P}}}$ matrix is stored explicitly as a dense matrix, and that $\mathbf{Q}_k$ is also stored as a dense matrix.

1. Show that $\mathbf{y}^{(K)} = \mathbf{x}^{(2^K)}$. This equivalence is why the algorithm is "exponentially accelerated."

   HINT: Use induction to show that $Q_K = \bar{\bar{\mathbf{P}}}^{2^K}$. The result follows by combining this fact with (6) and Equation (5.1) in the paper.

2. What is the runtime of calculating $\mathbf{Q}_k$ given $\mathbf{Q}_{k-1}$?

3. What is the runtime of computing $\mathbf{y}^{(K)}$ in terms of K?

4. As with the standard power method, we do not know the total number of iterations of the exponential power method in advance. Instead, we iterate until

$$\|\mathbf{y}^{(K)} - \mathbf{y}^{(K-1)}\|_2 \leq \epsilon, \tag{8}$$

where ϵ is a predetermined small constant value. Bounding the number of iterations K required to satisfy (8) is quite a bit more technical than in the previous problem. You do not have to compute this solution yourself; you may assume in subsequent problems that

$$K = O\left(\log \frac{\log \epsilon}{\log \alpha}\right) \tag{9}$$

satisfies (8). Notice that this number of iterations is logrithmic compared to the number of iterations in the standard power method, and this is where the name exponentially accelerated comes from.

5. What is the runtime of computing $\mathbf{y}^{(K)}$ in terms of ϵ?

6. Under what conditions is the exponentially accelerated power method faster than the standard power method?

7. Under what conditions is it slower?

8. What bad thing would happen if $\mathbf{P}$ was stored as a sparse matrix and $\bar{\bar{\mathbf{P}}}$ was calculated from $\mathbf{P}$ as in the standard power method?