# Notes: Pagerank II

## 1 Conceptual

**Problem 1.** 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 2.** Either prove or give a counterexample to the following claims.

1. The node with the largest out-degree can never have the highest pagerank.

2. Two nodes in a graph cannot have the same pagerank value.

3. Assume that all nodes in the graph have an in-degree of 2 and an out-degree of 2. Then all nodes will have the same pagerank.

## 2 Runtimes

**Problem 3.** 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

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

   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 (1) $K$ 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:

$$\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 \leq \epsilon, \tag{2}$$

where $\epsilon$ is a "small" number that controls how accurate we want our solution to be. The expression $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2$ is often called the *residual*.

Compute a formula for the number of iterations $K$ required to achieve a residual less than $\epsilon$.

HINT: See the discussion on page 346.

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 residual $\epsilon$.

**Problem 4.** Repeat Problem 3 assuming $P$ is stored as a dense matrix instead of a sparse matrix.

**Problem 5.** How do the results from problems 3 and 4 above compare to the LAPACK runtimes for computing the top-eigenvalue of a matrix?

# 3 Implementation Details

**Problem 6.** Why does it never make sense to store $\boldsymbol{\pi}$ as a sparse vector?

**Problem 7.** Why is the following inequality "almost always" true:

$$\|\mathbf{x}^{(k)}\|_2 < \|\mathbf{x}^{(k-1)}\|_2. \tag{3}$$

Based on Inequality (3) above, how should we adjust our implementation of the power method to ensure numerical stability?