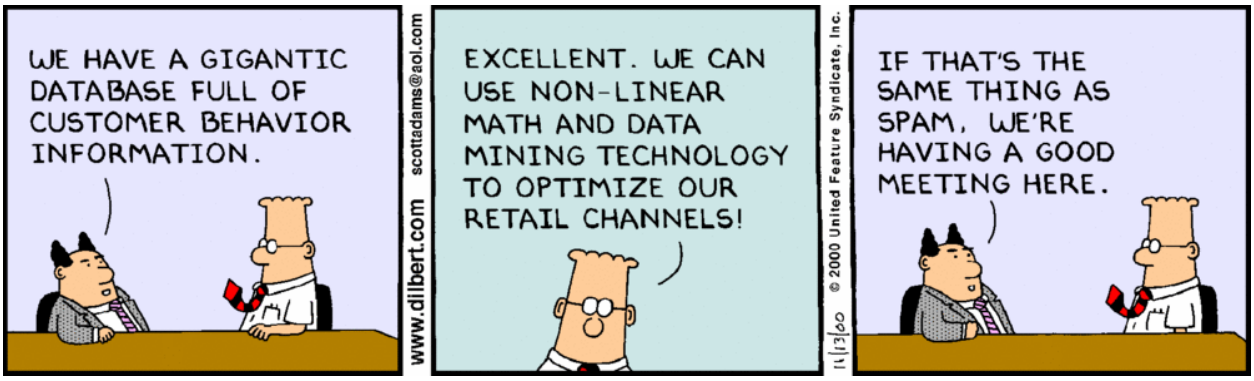
Notes: Pagerank I



1 Background

Pagerank is one of the most popular data mining techniques. It was made famous by the founders of Google (Larry Page and Sergey Brin), who used pagerank to improve the quality of results returned by search engines. Today, pagerank is also used in most fields that involve data. For example, it has been used to filter spam from social media, predict the behavior of drugs without clinical trials, detect intrusions into computer networks, and discover bugs in software. Search engines have significantly since Google first introduced pagerank over 20 years ago, but pagerank remains an integral part of how search engines work.

This sequence of notes covers some basic algorithms for computing the pagerank. I emphasize basic because there are literally hundreds of different algorithms that have been developed, and researchers are still developing new algorithms. These new, more advanced algorithms use concepts like distributed computing or more advanced math to improve their runtime. All of these algorithms return the same basic result— the top eigenvalue of a special graph called the *web graph*.

We will see how various design choices lead to a trade-off in computational accuracy versus speed. We will also explore an algorithm design pattern called *divide and conquer* for making algorithms faster. This will prepare you for the next portion of this course based off of the *Learning from Data* textbook, which will combine the concepts with statistics.

References 1. (recommended) 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", which discusses the importance of pagerank at a very high level.

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

References 2. Our primary text for this week is *Deeper Inside Pagerank* by Langville and Meyers. It is available on the github repo or at

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

You will be responsible for sections 1, 2, 3, 5.1, 6.1, 6.2.

2 Definitions

Note 1. You will have a closed-note quiz on Problem 1 below. For the quiz, I will print out these two sheets of paper exactly as they are in the notes, and you will have to fill out the definitions.

Problem 1. Reproduce the definitions from the reading of the following terms below.

1. irreducible matrix

2. primitive matrix

3. aperiodic markov chain

4. $\mathbf{P}$

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

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

7. $\boldsymbol{\pi}$

8. $\mathbf{v}$

Problem 2. The reading uses the following terms, but does not explicitly define them. Use wikipedia to find definitions for these terms.

You will not be tested on the exact, formal definitions of these terms. You just need to be able to use and understand them in context.

It is common for data mining papers to not define common terms. When I'm reading papers, I'm constantly having to look up definitions on wikipedia for these terms in order to understand the paper. One of the challenges of wikipedia is that there's a lot of information about each of these terms that may or may not be relevant. The main challenge of this problem is figuring out how to define these terms so that they help you understand the rest of this paper. I therefore recommend not looking up these terms in wikipedia until you encounter them in the reading; then provide definitions below that help you understand the reading.

1. markov chain

2. stationary vector

3. stochastic matrix

4. spectral radius

5. subdominant eigenvalue

3 "Trivial" Problems

The trivial problems in this section are designed to help you practice using the definitions above.

Note 2. Mathematicians define a problem to be *trivial* if the solution follows directly from the definitions without needing any major insights. Trivial problems can still take a long time to solve, however, because understanding the definitions is hard. Muggles[1] think that "trivial" problems should be "easy" to solve, but that's only the case if you have a really strong understanding of the involved definitions. It's okay if these trivial problems do not feel easy.

Problem 3. Give an example of:

1. a stochastic matrix

2. a non-stochastic matrix

[1] In the *Harry Potter* books, *muggles* are people who cannot use magic. I call non-mathematicians/non-computer scientists muggles because they see the sorts of things we do in this class as "magic".

3. an irreducible matrix

4. a reducible matrix

5. a primitive matrix

6. a non-primitive matrix

Note 3. Whenever I'm reading a data mining text, and I encounter a matrix, I always ask myself what sorts of properties the matrix might have. The purpose of the questions below is to help get you into a similar habit.

Problem 4. Answer the following questions.

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

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

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

Problem 5. Either prove or give a counterexample to the following claims.

HINT: Any claim which is true will have a "trivial" proof. When looking for counterexamples, use the identity and zero matrices as building-blocks.

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

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

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.