Vrije Universiteit Brussel

Faculty of Science
Master in Mathematics - specialization: Education

Similarity on Combinatorial Structures

Graduation thesis submitted in partial fulfillment of the requirements for
the degree of Master of Science in Mathematics - specialization: Education

Filip Moons

Promotor: Prof. Dr. P. Cara
Stagebegeleider: Promotor

Stagementor:

June 2015

Contents

Chapter 1

Preliminaries and notations

The Perron-Frobenius theorem states that a real square matrix with nonnegative entries has a unique largest real eigenvalue with an eigenvector that has only positive entries. The theorem was proved by Oskar Perron (1880-1975) in 1907 for strictly positive entries and extended by Ferdinand Georg Frobenius (1849-1917) to irreducible matrices with nonnegative entries.

1.1 Some families of matrices

In this section, we first introduce different kinds of matrices. Note that all matrices in this master thesis have real entries, unless otherwise stated. We start with permutation matrices and their uses. With permutation matrices, we can introduce irreducible matrices. Also nonnegative and primitive square matrices are presented. After defining those, we look at the Perron-Frobenius theorem.

1.1.1 Permutation matrices

Definition 1.1.1. *Given a permutation π of n elements:*

$$\pi : \{1, \dots, n\} \to \{1, \dots, n\},$$

with:

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & n \\ \pi(1) & \pi(2) & \cdots & \pi(n) \end{pmatrix}$$

*the associated **permutation matrix** P_π is the $n \times n$-matrix obtained by permuting the rows of the identity matrix I_n according to π. So:*

$$P_\pi = \begin{bmatrix} \mathbf{e}_{\pi(1)} \\ \mathbf{e}_{\pi(2)} \\ \vdots \\ \mathbf{e}_{\pi(n)} \end{bmatrix}.$$

where $\mathbf{e}_j$ is the j-th row of I_n.

Example 1.1.2. The permutation matrix P_π corresponding to the permutation

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 3 \end{pmatrix}$$

is:

$$P_\pi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Note that $p_{ij} = 1$ if and only if $\pi(i) = j$.

Property 1.1.3. *A permutation matrix P satisfies:*

$$PP^T = I_n,$$

where P^T is the transpose and I_n is the identity matrix.

Proof. By direct computation, we get:

$$(PP^T)_{ij} = \sum_{k=1}^{n} P_{ik} P_{kj}^T = \sum_{k=1}^{n} P_{ik} P_{jk}$$

Assume $i \neq j$. Then for each k, $P_{ik}P_{jk} = 0$ since there is only one nonzero entry in the k-th row and $i \neq j$, P_{ik} and P_{jk} can't be both the nonzero entry. So, $(PP^T)_{ij} = 0$ when $i \neq j$.

When $i = j$, then there exists a $k' \in \{1, \ldots, n\}$ with $P_{ik'}P_{jk'} = 1$, since there is only one nonzero entry in the k-th row, this k' is unique, which results in $\sum_{k=1}^{n} P_{ik}P_{jk} = (PP^T)_{ij} = 1$.

In other words,

$$(PP^T)_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases},$$

this is exactly the formula for the entries of the identity matrix. □

Corollary 1.1.4. *The transpose of a permutation matrix P is its inverse:*

$$P^T = P^{-1}.$$

This can also more easily be concluded by the fact that a permutation matrix is clearly an orthogonal matrix (a real $n \times n$-matrix with orthonormal entries).

1.1.2 Nonnegative and primitive matrices

Definition 1.1.5. *Let A and B be two real $n \times r$-matrices. Then, $A \geq B$ (respectively $A > B$) if $a_{ij} \geq b_{ij}$ (respectively $a_{ij} > b_{ij}$) for all $1 \leq i \leq n, 1 \leq j \leq r$.*

Definition 1.1.6. *A real $n \times r$-matrix A is **nonnegative** if $A \geq 0$, with 0 the n-null matrix.*

Definition 1.1.7. *A real $n \times r$-matrix A is **positive** if $A > 0$, with 0 the n-null matrix.*

Since row vectors are $1 \times n$-matrices, we shall use the terms nonnegative and positive vector throughout.

Notation 1.1.8. *Let B be an arbitrary complex $n \times r$-matrix, then $|B|$ denotes the matrix with entries $|b_{ij}|$. This is not to be confused with the determinant of a square matrix B, which we denote by* $\det(B)$.

Definition 1.1.9. *A nonnegative square matrix A is called **primitive** if there is a $k \in \mathbb{N}_0$ such that all entries of A^k are positive.*

1.1.3 Irreducible nonnegative matrices

In developing the Perron-Frobenius theory, we shall first establish a series of theorems and lemmas on nonnegative irredicuble square matrices.

Definition 1.1.10. *A square matrix A is called **reducible** if there is a permutation matrix P such that*

$$PAP^T = \begin{pmatrix} B & 0 \\ C & D \end{pmatrix}$$

*where B, and D are square matrices, each of size at least one and 0 is a zero matrix. A square matrix A is called **irreducible** if it is not reducible.*

It follows immediately that a 1×1-matrix is always irreducible by definition. We now show a useful property to identify a reducible matrix.

Property 1.1.11. *Let A be an $n \times n$-matrix with $n \geq 2$. Consider a nonempty, proper subset S of $\{1, \ldots, n\}$ with $a_{ij} = 0$ for $i \in S$, $j \notin S$. Then A is reducible.*

Proof. Let $S = \{i_1, i_2, \ldots, i_k\}$, where we assume, without loss of generality, that $i_1 < i_2 < \cdots < i_{k-1} < i_k$. Let S^c be the complement of S, consisting of the ordered set of elements $j_1 < j_3 < \cdots < j_{n-k}$. Consider the permutation σ of $\{1, 2, \ldots, n\}$ given by

$$\sigma = \begin{pmatrix} 1 & 2 & \ldots & k & k+1 & k+2 & \ldots & n \\ i_1 & i_2 & \ldots & i_k & j_1 & j_2 & \cdots & j_{n-k} \end{pmatrix}$$

σ can be represented by the permutation matrix $P_\sigma = (p_{ij})$, where $p_{rs} = 1$ if $\sigma(r) = s$. We prove that

$$PAP^T = \begin{pmatrix} B & 0 \\ C & D \end{pmatrix}$$

where B and D are square matrices and 0 is a $k \times (n-k)$ zero matrix. Consider row c and column d, where $1 \leq c \leq k$ and $k+1 \leq d \leq n$:

$$(PAP^T)_{cd} = \sum_i \sum_j p_{ci} a_{ij} p_{dj}. \tag{1.1}$$

It is enough to show that each term in the summation is zero. Suppose $p_{ci} = p_{dj} = 1$. Thus $\sigma(c) = i$ and $\sigma(d) = j$. Since $1 \leq c \leq k$, then $i \in \{i_1, i_2, \ldots, i_k\}$; similarly, since $k+1 \leq d \leq n$, we have $j \in \{j_1, j_2, \ldots, j_{n-k}\}$. By assumption, for such a pair i, j, we have $a_{ij} = 0$. That completes the proof. □

We know prove some equivalent definitions for a nonnegative, irreducible square matrix.

Theorem 1.1.12. *Let $A \geq 0$ be a nonnegative $n \times n$-matrix. Then the following conditions are equivalent:*

(1) A is irreducible.

(2) $(I + A)^{n-1} > 0$

(3) For any pair (i, j), with $1 \leq i, j \leq n$, there is a positive integer $k = k(i, j) \leq n$ such that $(A^k)_{ij} > 0$.

Proof. (1) $\Rightarrow$ (2): Let $\mathbf{x} \geq 0, \mathbf{x} \neq \mathbf{0}$ be an arbitrary vector in $\mathbb{R}^n$. If a coordinate of $\mathbf{x}$ is positive, the same coordinate is positive in $\mathbf{x} + A\mathbf{x} = (I + A)\mathbf{x}$ as well. We claim that $(I + A)\mathbf{x}$ has fewer zero coordinates than $\mathbf{x}$ as long as $\mathbf{x}$ has a zero coordinate. If this claim is not true, then the number of zero coordinates must be at least equal, this means that for each coordinate j with $x_j = 0$ we would have that $x_j + (A\mathbf{x})_j = 0$. Let $J = \{j : x_j > 0\}$. For any $j \notin J, r \in J$, we have $(A\mathbf{x})_j = \sum_k a_{jk}x_k = 0$ and $x_r > 0$. It must be that $a_{jr} = 0$. It follows from Property 1.1.11 that A is reducible, which is a contradiction and the claim is proved. Thus $(I + A)\mathbf{x}$ has at most $n - 2$ zero coordinates. Continuing in this manner we conclude that $(I + A)^{n-1}\mathbf{x} > 0$. Let $\mathbf{x} = \mathbf{e}_i$, then the corresponding column of $(I + A)^{n-1}$ must be positive. Thus (2) holds.

(2) $\Rightarrow$ (3): We have $(I + A)^{n-1} > 0$, $A \geq 0$, so $A \neq 0$ and

$$A(I + A)^{n-1} = \sum_{k=1}^{n} \binom{n-1}{k-1} A^k > 0.$$

Thus for any i, j at least one of the matrices $A, A^2, \ldots, A^n$ has its (i, j)-th element entry positive.

(3) $\Rightarrow$ (1): Suppose A is reducible. Then for some permutation matrix P,

$$PAP^T = \begin{pmatrix} B_1 & 0 \\ C_1 & D_1 \end{pmatrix}$$

where B_1 and D_1 are square matrices. Furthermore, we know from Property 1.1.3 that $PAP^TPAP^T = PA^2P^T$, whence for some square matrices B_2, C_2 we have:

$$PA^2P^T = \begin{pmatrix} B_2 & 0 \\ C_2 & D_2 \end{pmatrix}$$

More generally, for some matrix C_t and square matrices B_t and D_t,

$$PA^tP^T = \begin{pmatrix} B_t & 0 \\ C_t & D_t \end{pmatrix}$$

Thus $(PA^tP^T)_{rs} = 0$ for $t = 1, 2, \ldots$ and for any r, s corresponding to an entry of the zero submatrix in PAP^T. Now, for $t = 1, \ldots, n$:

$$0 = (PA^tP^T)_{rs} = \sum_k \sum_l p_{rk} a_{kl}^{(t)} p_{st}$$

By using the same reasoning as in 1.1, choose k, l so that $p_{rk} = p_{sl} = 1$. Then $a_{kl}^{(t)} = 0$ for all t, contradicting the hypothesis. This completes the proof. □

Corollary 1.1.13. *If A is irreducible then $I + A$ is primitive.*

Corollary 1.1.14. *A^T is irreducible whenever A is irreducible.*

Property 1.1.15. *No row or column of an irreducible matrix A can vanish. This means that A cannot have a row or a column of zeros.*

Proof. Suppose that A has a zero row, then it could be permuted to

$$PAP^T = \begin{pmatrix} 0 & 0\dots 0 \\ \mathrm{c}_1 & \\ \vdots & D \\ \mathrm{c}_n & \end{pmatrix}$$

by some permutation matrix P. It follows from Definition 1.1.10 that A is reducible. Similarly, if A has zero column, it can be permuted to

$$PAP^T = \begin{pmatrix} & 0 \\ B & \vdots \\ & 0 \\ c_1 \dots c_n & 0 \end{pmatrix}$$

again from Definition 1.1.10 we conclude that A is reducible.

□

1.2 Perron-Frobenius Theorem

1.2.1 Spectral radii of nonnegative matrices

Definition 1.2.1. *Let A be an $n \times n$-matrix with complex entries and eigenvalues λ_i, $1 \leq i \leq n$. Then:*

$$\rho(A) = \max_{1 \leq i \leq n} |\lambda_i|$$

*is called the **spectral radius** of the matrix A.*

Geometrically, if all the eigenvalues λ_i of A are plotted in the complex plane, then $\rho(A)$ is the radius of the smallest disk $|z| \leq R$, with center at the origin, which includes all the eigenvalues of the matrix A.

We now establish a series of lemmas on nonnegative irreducible square matrices. These lemmas will allow us to prove the Perron-Frobenius at the end of this section.

If $A \geq 0$ is an irreducible $n \times n$-matrix and $\mathbf{x}$, a vector of size n with $\mathbf{0} \neq \mathbf{x} \geq 0$, let

$$r_{\mathbf{x}} = \min \left\{ \frac{\sum_{j=1}^{n} a_{ij} x_j}{x_i} \right\} \tag{1.2}$$

where the minimum is taken over all i for which $x_i > 0$. Clearly, $r_{\mathbf{x}}$ is a nonnegative real number and is the supremum of all $p \geq 0$ for which

$$A\mathbf{x} \geq p\mathbf{x} \tag{1.3}$$

We now consider the nonnegative quantity r defined by

$$r = \sup_{\substack{\mathbf{x} \geq 0 \\ \mathbf{x} \neq \mathbf{0}}} \{r_{\mathbf{x}}\} \tag{1.4}$$

As $r_{\mathbf{x}}$ and $r_{\alpha\mathbf{x}}$ have the same value for any scalar $\alpha > 0$, we need consider only the set B of vectors $\mathbf{x} \geq 0$ with $||\mathbf{x}|| = 1$, and we correspondingly let Q be the set of all vectors $\mathbf{y} = (I + A)^{n-1}\mathbf{x}$ where $\mathbf{x} \in B$. From Theorem 1.1.12 , Q consists only of positive vectors. Multiplying both sides of the inequality $A\mathbf{x} \geq r_{\mathbf{x}}\mathbf{x}$ by $(I + A)^{n-1}$, we obtain:

$$\forall \mathbf{y} \in Q : A\mathbf{y} \geq r_{\mathbf{x}}\mathbf{y},$$

and we conclude from (1.3) that $r_{\mathbf{y}} \geq r_{\mathbf{x}}$. Therefore, the quantity r of (1.4) can be defined equivalently as:

$$r = \sup_{\mathbf{y} \in Q}\{r_{\mathbf{y}}\} \tag{1.5}$$

As B is a compact set (in the usual topology) of vectors, so is Q, and as $r_{\mathbf{y}}$ is a continuous function on Q, we know from the extreme value theorem that there necessarily exists a positive vector $\mathbf{z}$ for which:

$$A\mathbf{z} \geq r\mathbf{z}, \tag{1.6}$$

and no vector $\mathbf{w} \geq 0$ exists for which $A\mathbf{w} > r\mathbf{w}$.

Definition 1.2.2. *We call all nonnegative, nonzero vectors* $\mathbf{z}$ *satisfying (1.6)* ***extremal vectors*** *of the matrix* A.

Lemma 1.2.3. *If* $A \geq 0$ *is an irreducible* $n \times n$*-matrix, the quantity* r *of (1.4) is positive.*

Proof. If $\mathbf{x}$ is the positive vector whose coordinates are all unity, then since the matrix A is irreducible, we know from Property 1.1.15 that no row of A can vanish, and consequently no component of $A\mathbf{x}$ can vanish. Thus, $r_{\mathbf{x}} > 0$, proving that $r > 0$. □

Lemma 1.2.4. *If* $A \geq 0$ *is an irreducible* $n \times n$*-matrix, each extremal vector* $\mathbf{z}$ *is a positive eigenvector of* A *with corresponding eigenvalue* r *of (1.4), i.e.,* $A\mathbf{z} = r\mathbf{z}$ *and* $\mathbf{z} > 0$.

Proof. Let $\mathbf{z}$ be an extremal vector with $A\mathbf{z} - r\mathbf{z} = \mathbf{t}$. If $\mathbf{t} \neq \mathbf{0}$, then some coordinate of $\mathbf{t}$ is positive; multiplying through by the matrix $(I + A)^{n-1}$, we have:

$$A\mathbf{w} - r\mathbf{w} > 0, \text{ with } \mathbf{w} = (I + A)^{n-1}\mathbf{z}$$

from Theorem 1.1.12 we know that $\mathbf{w} > 0$. It would then follow that $r_{\mathbf{w}} > r$, contradicting the definition of r in (1.5). Thus $A\mathbf{z} = r\mathbf{z}$, and since $\mathbf{w} > 0$ and $\mathbf{w} = (1 + r)^{n-1}\mathbf{z}$, then we have $\mathbf{z} > 0$, completing the proof. □

Lemma 1.2.5. *Let* $A \geq 0$ *be an irreducible* $n \times n$*-matrix, and let* B *be an* $n \times n$*- complex matrix with* $|B| \leq A$. *If* β *is any eigenvalue of* B, *then*

$$|\beta| \leq r, \tag{1.7}$$

where r *is the positive quantity of (1.4). Moreover, equality is valid in (1.7), i.e.,* $\beta = re^{i\phi}$, *if and only if* $|B| = A$, *and where* B *has the form:*

$$B = e^{i\phi}DAD^{-1}, \tag{1.8}$$

and D *is a diagonal matrix whose diagonal entries have modulus unity.*

Proof. If $\beta\mathbf{y} = B\mathbf{y}$ where $\mathbf{y} \neq \mathbf{0}$, then

$$\beta y_i = \sum_{j=1}^{n} b_{ij} y_i, \text{ with } 1 \leq i \leq n.$$

Using the hypotheses of the lemma and the notation of Definition 1.1.8, it follows that:

$$|\beta||\mathbf{y}| \leq |B||\mathbf{y}| \leq A|\mathbf{y}|, \tag{1.9}$$

which implies that $|\beta| \leq r_{|\mathbf{y}|} \leq r$, proving (1.7). If $|\beta| = r$, then $|\mathbf{y}|$ is an extremal vector of A. Therefore, from Lemma 1.2.4, $|\mathbf{y}|$ is a positive eigenvector of A corresponding to the positive eigenvalue r. Thus,

$$r|\mathbf{y}| = |B||\mathbf{y}| = A|\mathbf{y}|, \tag{1.10}$$

and since $|\mathbf{y}| > 0$, we conclude from (1.10) and the hypothesis $|B| \leq A$ that

$$|B| = A \tag{1.11}$$

For the vector $\mathbf{y}$, where $|\mathbf{y}| > 0$, let

$$D = \operatorname{diag}\left\{\frac{y_1}{|y_1|}, \ldots, \frac{y_n}{|y_n|}\right\}.$$

It is clear that the diagonal entries of D have modulus unity, and

$$\mathbf{y} = D|\mathbf{y}|. \tag{1.12}$$

Setting $\beta = re^{i\phi}$, then $B\mathbf{y} = \beta\mathbf{y}$ can be written as:

$$C|\mathbf{y}| = r|\mathbf{y}|, \tag{1.13}$$

where

$$C = e^{-i\phi} D^{-1} B D. \tag{1.14}$$

From (1.10) and (1.13), equiting terms equal to $r|\mathbf{y}|$ we have

$$C|\mathbf{y}| = |B||\mathbf{y}| = A|\mathbf{y}|. \tag{1.15}$$

From the definition of the matrix C in (1.14), $|C| = |B|$. Combining with (1.11), we have:

$$|C| = |B| = A. \tag{1.16}$$

Thus, from (1.15) we conclude that $C|\mathbf{y}| = |C||\mathbf{y}|$, and as $|\mathbf{y}| > 0$, it follows that $C = |C|$ and thus $C = A$ from (1.16). Combining this result with (1.14), gives the desired result that $B = e^{i\phi} D A D^{-1}$. Conversely, it is obvious that if B has the form in (1.8), then $|B| = A$, and B has an eigenvalue β with $|\beta| = r$, which completes the proof. □

Corollary 1.2.6. *If $A \geq 0$ is an irreducible $n \times n$-matrix, then the positive eigenvalue r of Lemma 1.2.4 equals the spectral radius $\rho(A)$ of A*

Proof. Setting $B = A$ in Lemma 1.2.5 immediately gives us this result. □

In other words, if $A \geq 0$ is an irreducible $n \times n$-matrix, its spectral radius $\rho(A)$ is positive, and the intersection in the complex plane of the circle $|z| = \rho(A)$ with the positive real axis is an eigenvalue of A.

Definition 1.2.7. *A **principal square submatrix** of an $n \times n$-matrix A is any matrix obtained by crossing out any j rows and the corresponding j columns of A, with $1 \leq j \leq n$.*

Lemma 1.2.8. *If $A \geq 0$ is an irreducible $n \times n$-matrix, and B is any principal square submatrix of A, then $\rho(B) < \rho(A)$.*

Proof. If B is any principal submatrix of A, then there is an $n \times n$-permutation matrix P such that $B = A_{11}$ where

$$C = \begin{pmatrix} A_{11} & 0 \\ 0 & 0 \end{pmatrix}; PAP^T = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \tag{1.17}$$

Here, A_{11} and A_{22} are, respectively, $m \times m$ and $(n-m) \times (n-m)$ principal square submatrices of PAP^T, $1 \leq m \leq n$. Clearly, $0 \leq C \leq PAP^T$, and $\rho(C) = \rho(B) = \rho(A_{11})$, but as $C = |C| \neq PAP^T$, the conclusion follows immediately from Lemma 1.2.5 and Corollary 1.2.6. □

The following lemma is used to prove that $\rho(A)$ is a simple eigenvalue of A in the Perron-Frobenius theorem. The proof uses the extension of the product rule of derivation for multilinear functions $M(a_1, \ldots, a_k)$. Suppose $x_1, .., x_k$ are differentiable vector functions, then $M(x_1, \ldots, x_k)$ is differentiable and:

$$\frac{\mathrm{d}}{\mathrm{d}t} M(x_{1,k}) = M(\frac{\mathrm{d}}{\mathrm{d}t} x_1, x_2, \ldots, x_k) + M(x_1, \frac{\mathrm{d}}{\mathrm{d}t} x_2, \ldots, x_k) + \ldots + M(x_1, x_2, \ldots, \frac{\mathrm{d}}{\mathrm{d}t} x_k)$$

The most important application of this rule is for the derivative of the determinant:

$$\frac{\mathrm{d}}{\mathrm{d}t} \det(x_1, \ldots, x_k) = \det(\frac{\mathrm{d}}{\mathrm{d}t} x_1, x_2, \ldots, x_k) + \det(x_1, \frac{\mathrm{d}}{\mathrm{d}t} x_2, \ldots, x_k) + \ldots + \det(x_1, x_2, \ldots, \frac{\mathrm{d}}{\mathrm{d}t} x_k)$$

Lemma 1.2.9. *Let A be an $n \times n$-matrix over the complex numbers and let $\phi(A, \lambda) = \det(\lambda I_n - A)$ be the characteristic polynomial of A. Let B_i be the principal submatrix of A formed by deleting the i-th row and column of A and let $\phi(B_i, \lambda)$ be the characteristic polynomial of B_i. Then:*

$$\phi'(A, \lambda) = \frac{\mathrm{d}\phi(A, \lambda)}{\mathrm{d}\lambda} = \sum_i \phi(B_i, \lambda)$$

Proof. The proof is immediately done by direct computation:

$$\phi(A, \lambda) = \det \begin{bmatrix} \lambda - a_{1,1} & -a_{1,2} & \ldots & -a_{1,n} \\ -a_{2,1} & \lambda - a_{2,2} & \ldots & -a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n,1} & -a_{n,2} & \ldots & \lambda - a_{n,n} \end{bmatrix}.$$

Using the extension of the product rule of derivation for multilinear functions

$$\phi'(A,\lambda) = \det\begin{bmatrix} 1 & 0 & \dots & 0 \\ -a_{2,1} & \lambda - a_{2,2} & \dots & -a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n,1} & -a_{n,2} & \dots & \lambda - a_{n,n} \end{bmatrix} + \det\begin{bmatrix} \lambda - a_{1,1} & -a_{1,2} & \dots & -a_{1,n} \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n,1} & -a_{n,2} & \dots & \lambda - a_{n,n} \end{bmatrix} + \dots$$
$$+ \det\begin{bmatrix} \lambda - a_{1,1} & -a_{1,2} & \dots & -a_{1,n} \\ -a_{2,1} & \lambda - a_{2,2} & \dots & -a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} = \sum_i \phi(B_i,\lambda).$$

□

We now collect the above results into the following main theorem: we finally arrived at the Perron-Frobenius Theorem:

Theorem 1.2.10. *(**Perron-Frobenius theorem**) Let $A \geq 0$ be an irreducible $n \times n$-matrix. Then,*

1. *A has a positive real eigenvalue equal to its spectral radius.*
2. *To $\rho(A)$ there corresponds an eigenvector $\mathbf{x} > 0$.*
3. *$\rho(A)$ increases when any entry of A increases.*
4. *$\rho(A)$ is a simple eigenvalue of A.*
5. *If $A\mathbf{x} = \rho(A)\mathbf{x}$ where $\mathbf{x} > 0$ and $\mathbf{x}$ is a normalized vector, then $\mathbf{x}$ is unique.*

Proof. (1) and (2) follow immediately from Lemma 1.2.4 and Corollary 1.2.6.

(3) Suppose we increase some entry of the matrix A, giving us a new irreducible matrix $\tilde{A}$ where $\tilde{A} \geq A$ and $\tilde{A} \neq A$. Applying Lemma 1.2.5, we conclude that $\rho(\tilde{A}) > \rho(A)$.

(4) $\rho(A)$ is a simple eigenvalue of A, i.e., $\rho(A)$ is a zero of multiplicity one of the characteristic polynomial $\phi(\lambda) = \det(\lambda I_n - A)$, we make use of Lemma 1.2.9 by using the fact that $\phi'(\lambda)$ is the sum of the determinants of the principal $(n-1) \times (n-1)$ submatrices of $\lambda I - A$. If A_i is any principal submatrix of A, then from Lemma 1.2.8, $\det(\lambda I - A_i)$ (with I the identity matrix with the same size as the principal submatrix A_i) cannot vanish for any $\lambda \geq \rho(A)$. From this it follows that:

$$\det(\rho(A)I - A_i) > 0,$$

and thus

$$\phi'(\rho(A)) > 0.$$

Consequently, $\rho(A)$ cannot be z zero of $\phi(\lambda)$ of multiplicity greater than one and thus $\rho(A)$ is a simple eigenvalue of A.

(5) If $A\mathbf{x} = \rho(A)\mathbf{x}$ where $\mathbf{x} > 0$ and $||x|| = 1$ ($||x||$ denotes the standard Euclidean norm), we cannot find another eigenvector $\mathbf{y} \neq s\mathbf{x}$, with s a scalar, of A with $A\mathbf{y} = \rho(A)\mathbf{y}$, so that the eigenvector $\mathbf{x}$, meaning that the normalized eigenvector $\mathbf{x}$ is uniquely determined. □

1.2.2 Example

To check wether a matrix with nonnegative entries is primitive, irreducible or neither, we just have to replace all nonzero entries by 1 since this does not affect the classification. The matrix

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

is strictly positive and thus primitive. The matrices

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

both have 1 as a double eigenvalue hence can not be irreducible. The matrix $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ satisfies:

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^2 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$

and hence is primitive. The same goes for

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix},$$

this matrix is irreducible but not primitive. Its eigenvalues are 1 and -1.

1.3 Numerical analysis

1.3.1 Bachmann-Landau notations

For comparing the computational cost of algorithms, it's important to know the *Bachmann-Landau notations*. These notations are used to describe the limiting behavior of a function in terms of simpler functions. These notations are used a lot in computer science to classify algorithms by how their number of steps depends on changes in input size. We are only interested in the effects on the number of steps for really large input sizes, so constants don't play any role in the classification.

Bigh Oh

Definition 1.3.1. ***(Big Oh)***
Big Oh is the set of all functions f that are bounded above by g asymptotically (up to constant factor).

$$O(g(n)) = \{f | \exists c, n_0 \geq 0 : \forall n \geq n_0 : 0 \leq f(n) \leq cg(n)\}$$

We now proof a very simple lemma to show that indeed constant factors doesn't matter for Big Oh:

Lemma 1.3.2. $\forall k > 0 : O(k.g(n)) = O(g(n))$

Proof.

$$\begin{aligned} O(k.g(n)) &= \{f|\exists c, n_0 \geq 0 : \forall n \geq n_0 : 0 \leq f(n) \leq k.c.g(n)\} \\ &= \{f|\exists c, n_0 \geq 0 : \forall n \geq n_0 : 0 \leq f(n) \leq (k.c).g(n)\} \\ \text{let c' = k.c} & \\ &= \{f|\exists c', n_0 \geq 0 : \forall n \geq n_0 : 0 \leq f(n) \leq c'.g(n)\} \\ &= O(g(n)) \end{aligned}$$

□

Small Oh

Definition 1.3.3. ***(Small Oh)***
Small Oh is the set of all functions f that are dominated by g asymptotically.

$$o(g(n)) = \{f|\forall \varepsilon > 0, \exists n_0 \forall n \geq n_0 : f(n) \leq \varepsilon.g(n)\}$$

Note that the small oh-notation is a much stronger statement than the corresponding big oh-notation: every function that is in the small oh of g is also in big oh, but the inverse isn't necessarily true. Intuitively, $f(x) \in o(g(x))$ means that $g(x)$ grows much faster than $f(x)$.

Asymptotical Equality

Definition 1.3.4. ***(Asymptotically Equal)***
Let f and g real functions, then f is asymptotically equal to $g \Leftrightarrow \lim_{x \to +\infty} \frac{f(x)}{g(x)} = 1$. Notation: $f \sim g$.

In fact asymptotical equality, can also be defined as an equivalency relation: $f \sim g \Leftrightarrow (f - g) \in o(g)$. It's trivially clear that as $f \sim g \Rightarrow f \in O(g)$.

1.3.2 The Power Method

We now introduce the *classical power method*, also called the *Von Mises iteration* ([GOLUB]) because adaptions of this iterative method will appear in the following chapters of this master thesis. The power method is an eigenvalue algorithm that, given a diagonalizable matrix A, finds the eigenvalue λ with the greatest magnitude and a corresponding eigenvector $\mathbf{v}$ such that:

$$A\mathbf{v} = \lambda\mathbf{v}.$$

The power method is special because it doesn't use any matrix decomposition technique for obtaining results, making it suitable for very large matrices. At the other hand, it only finds one eigenvalue with a corresponding eigenvector and the the iterative process might converge very slowly.

There are plenty of variations of the power method available that overcome all these difficulties (finding only 1 eigenvalue/eigenvector, slow convergence, A must be diagonalizable, ...) but we limit our discussion here to the very basic method, therefore we call it the *classical* power method.

We first introduce some needed definitions, theorems and notations.

Definition 1.3.5. *Consider a real $n \times n$-matrix A with (not necessarily different) eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. When*

$$|\lambda_1| > |\lambda_2| \leq |\lambda_3| \leq \ldots \leq |\lambda_n|,$$

*λ_1 is called the **dominant eigenvector**.*

Corollary 1.3.6. *The dominant eigenvector λ_1 of a $n \times n$-matrix A is real.*

Proof. This is trivial, if λ_1 would be complex, the complex conjugate of λ_1 would also be an eigenvalue with the same modulus. □

Theorem 1.3.7. *A $n \times n$ matrix A is diagonalizable if and only if it has an eigenbasis (a basis containing only lineair independent eigenvectors).*

Proof. $\Leftarrow$ Consider $A \in \mathbb{R}^{n \times n}$ with (not necessarily different) eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and the eigenbasis $\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$. Let:

$$X = [\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}]$$

Because the columns of X are linearly independent, X is invertible. Since:

$$AX_i = \lambda_i X_i$$

we also have:

$$AX = X\Lambda$$

with $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$, thus:

$$X^{-1}AX = \Lambda.$$

$\Rightarrow$ When $A \in \mathbb{R}^{n \times n}$ is diagonalizable, then there exist an invertible matrix T so:

$$TAT^{-1} = D$$

with D diagonal. Let $Y = T^{-1}$, then:

$$AY = YD.$$

This means that the columns of Y are eigenvectors where the corresponding eigenvalues are noted in the diagonal matrix D. Because T and Y are invertible, the columns of Y are lineair independent. This means that A has n lineair independent eigenvectors, forming an eigenbasis. □

Notation 1.3.8. *Let $\mathbf{x}^{(i)}$ denote vector $\mathbf{x}$ at iteration step i.*

The algorithm

Let $A \in \mathbb{R}^{n \times n}$ be a diagonalizable matrix with dominant eigenvalue λ_1 . We know that A has eigenbasis $V = \{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$. Every vector $\mathbf{x^{(0)}} \in \mathbb{R}^n$ can be written as a lineair combination of elements in X, because X spans the space $\mathbb{R}^n$. So:

$$\mathbf{x}^{(0)} = \sum_{i=1}^{n} \xi_i \mathbf{v_i}.$$

Now construct the sequence of vectors $\mathbf{x^{(k)}}$:

$$\mathbf{x^{(k)}} = A\mathbf{x^{(k-1)}} = A^k\mathbf{x^{(0)}}$$

Now:

$$\begin{aligned} A^k\mathbf{x^{(0)}} &= \sum_{i=1}^{n} \xi_i A^k \mathbf{v_i} \\ &= \sum_{i=1}^{n} \xi_i \lambda_i^k \mathbf{v_i} \\ &= \lambda_1^k \left\{ \xi_1 \mathbf{v_1} + \xi_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k \mathbf{v_2} + \ldots + \xi_n \left(\frac{\lambda_n}{\lambda_1}\right)^k \mathbf{v_n} \right\} \end{aligned}$$

Because $|\lambda_i| < |\lambda_1|$ for $i > 1$, we have that

$$\left(\frac{\lambda_i}{\lambda_1}\right)^k \to 0 \text{ for } k \to \infty$$

so:

$$\mathbf{x^{(k)}} = \lambda_1^k \xi_1 \mathbf{v_1} + o(1) \text{ for } k \to \infty \tag{1.18}$$

This means that for k large enough $\mathbf{x^{(k+1)}}$ is equal to λ_1 times $\mathbf{x^{(k)}}$. So when, the ratio between the corresponding vector entries in $\mathbf{x^{(k+1)}}$ and $\mathbf{x^{(k)}}$ becomes constant after k iteration steps, then this ratio will be equal to the dominant eigenvalue λ_1. $\mathbf{x^{(k)}}$ will be a corresponding eigenvector because it's is proportional to $\mathbf{v_1}$.

The start value $\mathbf{x^{(0)}}$ must only satisfy the condition that $\xi_1 \neq 0$, in other words: $\mathbf{x^{(0)}}$ must have a non-zero component belonging to the dominant eigenvector. In general, each randomly chosen option for $\mathbf{x^{(0)}}$ will normally fulfill this requirement. Even if we are so unlucky to pick a starting vector which doesn't, subsequent $\mathbf{x^{(k)}}$ will again fulfill the requirement because rounding errors sustained during the iteration will have a component in this direction.

A practical problem arises now when one of the components of $\mathbf{x^{(k)}}$ is equal to zero. If we want to take the ratio between the corresponding components of $\mathbf{x^{(k)}}$ and $\mathbf{x^{(k+1)}}$ we get a division by zero. We can solve this by a property that any norm function has:

$$\|\lambda_1 \mathbf{x^{(k)}}\| = |\lambda_1| \|\mathbf{x^{(k+1)}}\|.$$

Because $\mathbf{x^{(k)}} \neq 0$ we have that $\|\mathbf{x^{(k+1)}}\| \approx \|\lambda_1 \mathbf{x^{(k+1)}}\|$, so we calculate λ_1 in the power method by:

$$|\lambda_1| = \lim_{k \to \infty} \frac{\|\mathbf{x^{(k+1)}}\|}{\|\mathbf{x^{(k)}}\|}$$

To decide on the sign of λ_1, just divide two non-zero components of f $\mathbf{x^{(k+1)}}$ and $\mathbf{x^{(k)}}$.

Another issue to address is that the components of $\mathbf{x^{(k)}} = A^k\mathbf{x^{(0)}}$ can become very high or very low, which can cause an overflow or onderflow in the real number representation of computers. To avoid this, we use normed versions of the $\mathbf{x^{(k)}}$-vectors: we start with a vector $\mathbf{y^{(0)}}$ with $\|\mathbf{y^{(0)}}\| = 1$. Subsequently, we calculate for $k = 0, 1, \ldots$:

$$\mathbf{z^{(k+1)}} = A\mathbf{y^{(k)}}, \quad \mu_{k+1} = \|\mathbf{z^{(k+1)}}\|, \quad \mathbf{y^{(k+1)}} = \frac{\mathbf{z^{(k+1)}}}{\mu_{k+1}}.$$

The vectors $\mathbf{y^{(k)}}$ have all magnitude 1 and the components of $\mathbf{z^{(k)}}$ are restricted because:

$$\|\mathbf{z^{(k)}}\| = \|A\mathbf{y^{(k-1)}}\| \leq \|A\|\|\mathbf{y^{(k-1)}}\| = \|A\|$$

and when A is invertible we have $\|\mathbf{y^{(k-1)}}\| = 1 \leq \|A^{-1}\|\|\mathbf{z^{(k)}}\|$. Thus $\|\mathbf{z^{(k)}}\| \geq \frac{1}{\|A^{-1}\|}$. If we want to calculate the eigenvalue we can use:

$$\begin{aligned} A^k\mathbf{y^{(0)}} &= A^{k-1}A\mathbf{y^{(0)}} = A^{k-1}\mathbf{z^{(1)}} = A^{k-1}\mu_1\mathbf{y^{(1)}} \\ &= A^{k-2}\mu_1 A\mathbf{y^{(1)}} = A^{k-2}\mu_1\mathbf{z^{(2)}} = A^{k-2}\mu_1\mu_2\mathbf{y^{(2)}} \\ &= \dots \\ &= \mu_1\mu_2\dots\mu_k\mathbf{y^{(k)}} \end{aligned}$$

So:

$$|\lambda_1| = \lim_{k\to\infty}\frac{\|A^{k+1}\mathbf{y^{(0)}}\|}{\|A^k\mathbf{y^{(0)}}\|} = \lim_{k\to\infty}\frac{\mu_1\mu_2\dots\mu_{k+1}\|\mathbf{y^{(k+1)}}\|}{\mu_1\mu_2\dots\mu_k\|\mathbf{y^{(k)}}\|} = \lim_{k\to\infty}\mu_{k+1}$$

Because μ_{k+1} converges, a good choice for a stop condition for our numerical algorithm could be

$$|\mu_k - \mu_{k-1}| < Tol,$$

which guarantees an estimation error of at most Tol (usually $Tol = 10^{-5}$) for the approximation of the dominant eigenvalue. With all this information, we construct algorithm 1. Just to give an understandable algorithm, we used the Euclidean norm in this algorithm[1].

Data:
$\mathbf{y^{(0)}}$: a start vector with $\|\mathbf{y^{(0)}}\|_2 = 1$,
Tol: Tolerance for the estimation error.
Result:
$\mathbf{y^{(k)}}$: an estimation of a dominant eigenvector,
μ_k: an estimation of the dominant eigenvalue.
begin `power_method(`$\mathbf{y^{(0)}}$, k`)`
  $k = 1$;
  repeat
    $\mathbf{z^{(k)}} = A\mathbf{y^{(k-1)}}$;
    $\mu_k = \|\mathbf{z^{(k)}}\|_2$;
    $\mathbf{y^{(k)}} = \dfrac{\mathbf{z^{(k)}}}{\mu_k}$;
    $k = k + 1$
  until $k > 2$ *and* $|\mu_k - \mu_{k-1}| < Tol$,;
  if *the components* $\mathbf{y^{(k)}}$ *and* $\mathbf{y^{(k-1)}}$ *have a different sign* **then**
    $\mu_k = -\mu_k$;
  end
  return $\mathbf{y^{(k)}}, \mu_k$;
end

Algorithm 1: The Power method

[1] $||.||_2$ is the Euclidean vector norm.

Computational cost & Usage

The computational cost of the algorithm is determined by the speed at which the $o(1)$ terms in 1.18 go to zero. This is indicated by the slowest converging term $(\lambda_2/\lambda_1)^k$. This means that the algorithm converges slowly when there is an eigenvalue close in magnitude to the dominant eigenvalue. We get following expression for approximation μ_k of λ_1:

$$|\mu^k - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$$

In our algorithm we accepted an estimation error of 10^{-5}, so the number of steps n can be computed as:

$$\left|\frac{\lambda_2}{\lambda_1}\right|^n \approx Tol$$

So, for example, for $Tol = 10^{-5}$ we get $n = -5/\log \lambda_1/\lambda_2$, we become:

$$\texttt{power\_method} \in O\left(\frac{-1}{\log \lambda_2/\lambda_2}\right)$$

Note that this O holds for any estimation error Tol of the form 10^{-e} with $e \in \mathbb{N}$.

When $\lambda_2 \approx \lambda_1$ we see that the power method (almost) has infinitely many steps. Since we do not know the eigenvalues of A, this means we cannot know in advance whether the power method will work or not. Recall that the eigenvalues of a real matrix A are in general complex, and occur in conjugate pairs. This means, when the dominant eigenvalue of A is not real, the power method will certainly fail. Therefore, it is a good idea to apply the power method only to matrix whose eigenvalues are known to be real. The only thing that can go wrong with those matrices is that the dominant eigenvalue has an algebraic multiplicity larger than 1.

From the Perron-Frobenius theorem in 1.2.10, we also get another good choice: namely the irreducible matrices or any matrix with strictly positive entries. Indeed, The Perron-Frobenius theorem tells us that they have a unique dominant eigenvalue.

Example

Example 1.3.9. Consider the matrix:

$$A = \begin{pmatrix} 1 & -3 & 5 \\ -1 & -7 & 11 \\ -1 & -9 & 13 \end{pmatrix}$$

A has a dominant eigenvalue 3 and a double eigenvalue 2. A corresponding dominant eigenvector is $(1, 1, 1)$. Now we use the classical power method with start vector $\mathbf{y^{(0)}} = (1, 0, 0)$, $Tol = 10^{-5}$ and we become the values in Table 1.3.9. Because we know the eigenvalues of A, we can predict the number of steps $n = -5/\log \lambda_2/\lambda_2 = -5/(\log 2/3) \approx 28$. Because $\lambda_2/\lambda_1 = 2/3$ is not that small, the convergence here is also not that fast. We have for the approximation of a corresponding eigenvector

$$\mathbf{y^{(29)}} = (-0.577348, -0.577352, -0.577351)$$

which is (more or less) in proportion with $(1, 1, 1)$.

k	μ_k		k	μ_k
0	1.00000		15	3.00459
1	1.73205		16	3.00305
2	4.12311		17	3.00204
3	4.06564		18	3.00136
4	3.58774		19	3.00090
5	3.34047		20	3.00060
6	3.20743		21	3.00040
7	3.13055		22	3.00026
8	3.08385		23	3.00017
9	3.05457		24	3.00011
10	3.03582		25	3.00008
11	3.02363		26	3.00005
12	3.01564		27	3.00003
13	3.01037		28	3.00002
14	3.00689		29	3.00001

Table 1.1: The iteration values μ_k of Example 1.3.9.

1.4 Graphs

After introducing different kinds of matrices and proving the Perron-Frobenius theorem, we now take a closer look at graphs. Here too, we'll look at different families of graphs and prove some relevant properties about them. We also link the concept of graphs with different kinds of matrices, deepening our insight of some theorems of the previous section.

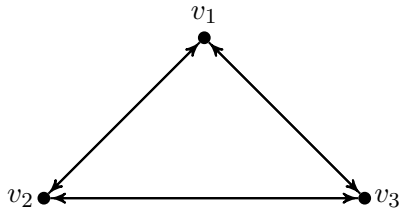
The definitions and results in this section are mainly based on the course 'Discrete Mathematics' by P. Cara [CARA].

1.4.1 General definitions

Definition 1.4.1. *A **graph** is a an ordered pair $(V, \to)$ where V is a set and $\to$ is a relation. The elements of V are called **vertices** and $\to$ is called the **adjacency relation**. Let $u, v \in V$, then the ordered pair (u, v) belonging to $\to$ is called an **arc** or **edge** and we write $u \to v$. We also say that u is **adjacent** to v. When $v \to v$ (with $v \in V$) we say that the graph has a **loop** in v. A graph $(V, \to)$ is most of the time denoted by calligraphic letters $\mathscr{G}, \mathscr{H}, \ldots$*

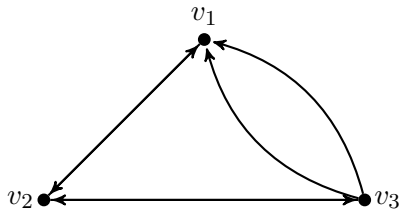
When the relation $\to$ is symmetric, we call the graph **undirected**, in this case we often write $\sim$ instead of $\to$.

Example 1.4.2. The graph



is an undirected graph with vertices v_1, v_2, v_3. The adjacency relation $\rightarrow$ equals $\{(v_1, v_2), (v_2, v_1), (v_2, v_3), (v_3, v_2), (v_3, v_1), (v_1, v_3)\}$.

There is a small problem with our definition, because not all graphs are taken into account, by example the graph below is not a graph following our definition because you can not define multiple edges between vertices in a relation.



Therefore we introduce a more general definition and introduce the concept of multiplicity of an edge:

Definition 1.4.3. *A graph is an ordered pair (V, μ) with V a set and $\mu : V \times V \rightarrow \mathbb{N}$ a function that gives the **multiplicity** of an edge. The function is defined as follows:*

- *when $\mu(u, v) = 0$ we say that u and v are not adjacent;*
- *when $\mu(u, v) = k > 0$ we say that there are k edges from u to v.*

It is clear that our previous definition fits perfectly in this more general definition, by constructing μ in this case as follows:

$$\mu(u, v) = \begin{cases} 1 & \text{if } (u, v) \in \rightarrow \\ 0 & \text{otherwise.} \end{cases}$$

Definition 1.4.4. *The **neighbourhood** of a vertex v of a graph $\mathscr{G} = (V, \mu)$ is the induced subgraph $\mathscr{G}_v$ with vertex set V' consisting of all vertices adjacent to v without v itself and with the multiplicity function μ', which is the restriction of μ to the vertices in V'. A vertex with a neighbourhood equal to the empty graph (a graph with an empty set of vertices) is called* isolated.

Definition 1.4.5. *The **order of a finite graph** $\mathscr{G}$ is the number of vertices of $\mathscr{G}$ and is denoted by $|\mathscr{G}|$.*

Definition 1.4.6. *The **degree of a vertex** v **in a graph** $\mathscr{G}$ is the number of edges containing v and is denoted by* $\deg(v)$*, so:*

$$\deg v = |\mathscr{G}_v|$$

Definition 1.4.7. *A **walk** in a graph $\mathscr{G}$ is a sequence of vertices*

$$v_0, v_1, \ldots, v_k$$

*such that $v_{i-1} \rightarrow v_i$ for each $i \in \{1, \ldots, k\}$. The **length** of the walk is k, one less than the number of vertices.*

Definition 1.4.8. *If all edges are distinct in a walk in a graph $\mathscr{G}$, we call the walk a **path**.*

Definition 1.4.9. *A **cycle** is a walk from v_0 to v_0 in which all vertices except v_0 are distinct.*

Definition 1.4.10. *A graph $\mathscr{G}$ is **connected** if it possible to establish a path from any vertex to any other vertex.*

Product graphs

Definition 1.4.11. *Take two graphs $\mathscr{G}(U, \to), \mathscr{H}(V, \to')$, the **product graph** $\mathscr{G} \times \mathscr{H}$ is the graph with $|\mathscr{G}|.|\mathscr{H}|$ vertices and that has an edge between vertices (u_i, v_j) and (u_k, v_l) if there is an edge between u_i and u_k in $\mathscr{G}$ and there is an edge between v_j and v_l in $\mathscr{H}$.*

Adjacency matrices

We now represent an undirected graph in the form of an adjacency matrix. This matrix gives a lot of useful information about the graph and vice versa.

Definition 1.4.12. *Let $\mathscr{G} : (V, \sim)$ be an undirected graph of order n and define a numbering on the vertices $v_1, \ldots, v_n$. Then the **adjacency matrix** $A_{\mathscr{G}}$ of $\mathscr{G}$ is the real $n \times n$-matrix with a_{ij} equal to the number of edges between i and j.*

Theorem 1.4.13. *Let $k > 0$. The element on place (i, j) in $A^k_{\mathscr{G}}$ contains the number of walks of length k from i to j.*

Proof. By induction on k.

For $k = 1$ we count the walks of length 1. These are edges and the result follows immediately from the defintion of $A_{\mathscr{G}}$.

Let v_l be a vertex of $\mathscr{G}$. If there are b_{ij} walks of length k from i to l and a_{lj} walks of length 1 (edges) from v_l to v_j, then there are $b_{il}a_{lj}$ walks of length $k+1$ from v_i to v_j passing vertex v_l. Therefore, the number of walks of length $k+1$ between v_i and v_j is equal to:

$$\sum_{l \in V(\mathscr{G})} b_{il}a_{lj} =: c_{ij}.$$

By the induction hypothesis we now that b_{il} equals the element on place (i, l) in $A^k_{\mathscr{G}}$ so c_{ij} is exactly the element on place (i, j) in the matrix product

$$A^k_{\mathscr{G}} A_{\mathscr{G}} = A^{k+1}_{\mathscr{G}}.$$

□

Example 1.4.14. The adjacency matrix of the graph in Example 1.4.2 is:

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

1.4.2 Simple graphs (andere implicatie!)

We now introduce simple graphs and show that the adjacency matrix of a simple graph $\mathscr{G}$ is irreducible if and only if $\mathscr{G}$ is connected.

Definition 1.4.15. *A **simple graph** is an undirected graph $\mathscr{G}(V, \mu)$ containing no loops and for all vertices $v_i, v_j \in V$, we have that the multiplicity $\mu(v_i, v_j)$ is at most 1.*

Theorem 1.4.16. *Let $\mathscr{G}$ be a simple graph with n vertices and adjacency matrix A. Then $\mathscr{G}$ is connected if and only if A is irreducible.*

Proof. A path between two vertices in the simple graph $\mathscr{G}$ has at most $n-1$ edges. So, $\mathscr{G}$ is connected if and only if $\forall i, j \in V(\mathscr{G})$ there exists a $k \leq n-1$ with a path of length k from i to j. So $\forall i, j \in V(\mathscr{G}) : \exists k < n : (A^k_{\mathscr{G}})_{ij} > 0$. Because $(I_n + A)^{n-1} = \sum_{k=0}^{n-1} \binom{n-1}{k} A^k$, we know from Theorem 1.1.12 that A is irreducible. □

1.4.3 Directed graphs

In this section, we take a closer look at *directed graphs*. We already showed that the adjacency matrix of a simple graph $\mathscr{G}$ is irreducible if and only if $\mathscr{G}$ is connected. But this is not appropriate for checking the irreducibility of matrices using graphs, because the only matrices we can check are the ones containing entries equal to 0 or 1. Can we find a more general method for checking irreducibility of a matrix using graphs? Luckily, we can and that's where directed graphs come in: we will introduce a method for turning a matrix in to a directed graph and vice versa and link the irreducibility of this matrix to a property of the directed graph.

Definition 1.4.17. *Andere gewone graffen en paden zijn toch ook gericht? A graph $\mathscr{G}(V, \to)$ is **directed** when the relation $\to$ is not symmetric. A directed graph $\mathscr{G}(V, \to)$ is **connected** if the underlying undirected graph (remove all arrows on the edges) is connected. $\mathscr{G}$ is **strongly connected** if for any two vertices v_i and v_j there exists a **directed path**:*

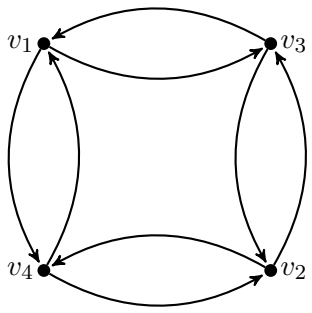
$$v_i \to v_{l_1}, v_{l_1} \to v_{l_2}, \ldots, v_{l_{r-1}} \to v_{l_r=j}.$$

We now link directed graphs with $n \times n$-matrices and vice versa. Consider any $n \times n$-matrix A, and consider any n distinct points $v_1, v_2, \ldots, v_n$ in the plane, which will be the nodes of the directed graph. For every nonzero entry a_{ij} of the matrix, we connect the node v_i to the node v_j by means of an arc $v_i \to v_j$ directed from v_i to v_j. In this way, with every $n \times n$-matrix A can be associated a **directed** graph $\mathscr{G}(A)$.

Example 1.4.18. Consider the matrix:

$$B = \begin{pmatrix} 0 & 0 & 89 & 7 \\ 0 & 0 & 2 & 2 \\ 123 & 9 & 0 & 0 \\ 14 & 89 & 0 & 0 \end{pmatrix}.$$

We get the directed graph:



Notice that this graph is strongly connected.

In the next proof, we study the equivalence of the matrix property of irreducibility of Definition 1.1.10 with the concept of the strongly connected directed graphs of a matrix:

Theorem 1.4.19. *An $n \times n$-matrix A is irreducible if and only if its directed graph $\mathscr{G}(A)$ is strongly connected.*

Proof. First assume that A is reducible. Consider:

$$PAP^T = \begin{pmatrix} B & 0 \\ C & D \end{pmatrix}$$

where D is a $k \times k$-submatrix and $1 \leq k \leq n-1$, and B is a square matrix of size at least 1. Consider any directed path from vertex v_i with $k < i$. The last segment of the path is determined by the presence of a nonzero element PAP^T_{ij} in the ith row of PAP^T. This row has zeros in the last k entries, so it is possible to make a connection from node i to node j only if $j < k$. Similarly, the path can be extended from node j only to another node smaller than k. Continuing in this way, we found that a directed path from node i with $i < k$ cannot be connected to a node greater than $k+1$. Hence the directed graph $\mathscr{G}(PAP^T)$ is not strongly connected. The directed graph $\mathscr{G}(A)$ is also not strongly connected: observe that the graph of PAP^T is obtained from that of A just by renumbering the nodes, and this operation does not affect the connectedness of the graph.

Now assume that $\mathscr{G}(A)$ is not strongly connected. Then there are distinct vertices v_i and v_j of $\mathscr{G}(A)$ for which there is no directed path from v_i to v_j. Let W_1 consist of v_j and all vertices from which there is directed path to v_j, let W_2 consist of v_i and all vertices to which there is a directed path from v_i. The sets W_1 and W_2 are nonempty and disjoint. Let W_3 be the set consisting of those vertices which belong to neither W_1 nor W_2 ($W_3 = V \backslash (W_1 \bigcup W_2)$). We now permute A so that the rows and columns corresponding to the vertices in W_2 come first followed by those corresponding to the vertices in W_3:

$$\begin{array}{c} \\ W_2 \\ W_3 \\ W_1 \end{array} \begin{array}{c} \begin{array}{ccc} W_2 & W_3 & W_1 \end{array} \\ \begin{pmatrix} B_1 & B_2 & O_1 \\ B_3 & B_4 & O_2 \\ C_3 & C_4 & D \end{pmatrix} \end{array}$$

Since there is no directed walk from v_i to v_j there is no arc from a vertex in W_2 to a vertex in W_1. Also there is no arc from a vertex v_k in W_3 to a vertex in W_1, because such an arc implies that v_k belongs to W_1. Hence $O_1 = 0$ and $O_2 = 0$, D is a square matrix and so is

$$B = \begin{array}{c} \\ W_2 \\ W_3 \end{array} \begin{array}{c} \begin{array}{cc} W_2 & W_3 \end{array} \\ \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix}. \end{array}$$

Hence, A is reducible. □

Chapter 2

Similarity on graphs

In the previous chapter all the basic terminology and results were introduced, now we take an extensive look at the concept of similarity on graphs. Similarity on graphs is a fairly new concept to compare the nodes of two graphs. The concept arose from the research on algorithms for web searching engines (like *Google*, *Yahoo*,...) in the late nineties. More specifically, Jon M. Kleinberg introduced in his paper 'Authoritative Sources in a Hyperlinked Environment' ([KLEINBERG]) the famous 'HITS algorithm' for extracting information from the link structure of websites. The method leads to an iterative algorithm where graphs represent the link structure of a collection of websites on a specific topic. Because this paper formed the basis of later research on similarity on graphs, the whole idea and algorithm of Kleinberg is introduced in the first section of this chapter. In 2004, V.D. Blondel et al. ([BLONDEL]) generalized the algorithm of Kleinberg, introducing the notion of similarity on directed graphs. This similarity is covered in the second section. With this similarity on directed graphs, there is a much wider scope of applications than just search algoritmes. Finally, we conclude by looking at similarity on colored graphs in the last section. Both sections on similarity on directed and colored graph conclude with an application on the Eurovision Song Contest results between 1980-1990.

2.1 The HITS algorithm of Kleinberg

2.1.1 History

Back in the nineties, internet became more and more popular by the public. The popular search engines back then where Altavista and Yahoo, but they weren't as advanced as search engines today. The main pitfall of the first search engines was that the search results were purely based on the number of occurrences of a word in a webpage. This was a pitfall for many reasons. The first reason was the growing popularity of the internet: as more and more webpages were put online, simply getting the relevant pages to a search query in this text-based manner, was a process that could possibly return millions of relevant pages. Also *content similarity* was an issue: a website owner can easily cheat in a text-based search system by just adding and repeating some very popular search words, making his website appear in the results of a large number of search queries. Two possible solutions were simultaneously invented in 1997 and 1998. The first one was the *PageRank*-system developed by Larry Page and Sergey Brin ([PAGE]). The PageRank system led to the foundation of the immensely popular Google search engine. Meanwhile, also Jon Kleinberg came up with his own solution,

the *HITS algorithm* (hyperlink-induced topic search) as a professor in the Computer Science Department at the Cornell University and researcher for IBM. The algorithm is used inter alia today by the Ask search engine (www.ask.com). Both these algorithms use the hyperlinks between webpages to rank search results. Because this master thesis is about similarity and this concept is introduced on graphs as a generalization of the HITS algorithm, we don't go to into further detail about the PageRank-algorithm. In the following paragraphs, the HITS algorithm is extensively explained.

2.1.2 Motivation

Kleinbergs work originates in the problem that arise with text-based searching the WWW. Text-based searching just counts all the occurrences of a given search query on webpages and returns a set of webpages ordered by decreasing occurency. When a user supplies a search query, we probably face an *abundance problem* with this method: the number of pages that could reasonably be returned as relevant is far too large for a human user to digest. To provide effective search results under these conditions, we need to filter the 'authoritative' ones. We face some complications when we want to filter the 'authoritative' webpages in a text-based system. By example, if we search for '`job offers in Flanders` ' the most authoritative page and expected first result in a search engine would be `www.vdab.be`. Unfortunately, the query '`job offers`' is used in over a million pages on the internet and `www.vdab.be` is not the one using the term most often. Therefore, there is no way to favor `www.vdab.be` in a text-based ranking function. This a recurring phenomenon, as another example if you search for the query '`computer brands`', there is no reason at all to be sure that the website of Apple or Toshiba even contain this search term.

The HITS algorithm solves these difficulties by analyzing the hyperlink structure among webpages. The idea is that hyperlinks encode a sort of human judgment and that this judgement is crucial to formulate a notion of authority. Specifically, when a page p includes a link to page q, it means that p gives a *conferred authority* on q. Again we face difficulties, because this conferred authority doesn't hold for every link. Links are created for a wide variety of reasons, for example, a large number of links are created for navigation within a website (e.g. "Return to homepage") and these have off course nothing to do with a notion of authority.

The HITS method is based on the relationship between the *authorities* for a topic and those pages that link to many related authorities, called *hubs*. Page p is called an *authority* for the query "`smartphone brand`" if it contains valuable information on the subject. In our example websites of smartphone manifacturers such as "`www.apple.com`", "`www.samsung.com`",... would be good authorities for this search query and these are the results a user expect from a search engine.

A hub is a second category of pages needed to find good authorities. Their role is to advertise authoritative pages. Hubs contain useful links toward these authorities. In our example, consumer websites with reviews on smartphones, websites of smartphone shops,... would be good hubs. In fact, hubs point the search process in the 'right direction'.

To really grasp the idea, we make an analogy with everyday life. If you tell a friend that you think of buying a new smartphone, he might tell you his experiences with smartphones and he will probably share some opinions he got from other friends. He might suggest you some good models and good brands. Now, you are more inclined to buy a smartphone that your friend suggested. Well, this idea is used in the HITS-method: your friend served as hub,

Data:
σ: a query string.
$\mathcal{E}$: a text-based search engine.
t: natural number (usually initiated to 200)
d: natural number (usually initiated to 50).
Result: A page set S_σ satisfying all the properties of our wish list.
begin `create_graph(`σ, $\mathcal{E}$, t, d`)`
  Let R_σ denote the top t results of $\mathcal{E}$ on σ;
  Set $S_\sigma := R_\sigma$;
  for *each page* $p \in R_\sigma$ **do**
    Let $\Gamma^+(p)$ denote the set of all pages p points to;
    Let $\Gamma^-(p)$ denote the set of all pages pointing to p;
    Add all pages in $\Gamma^+(p)$ to S_σ;
    if $|\Gamma^-(p)| \leq d$ **then**
      Add all pages in $\Gamma^-(p)$ to S_σ;
    else
      Add an arbitrary set of d pages from $\Gamma^-(p)$ to S_σ;
    end
  end
  return S_σ;
end

Algorithm 2: Algorithm to construct S_σ.

the brands and models he suggested are good authorities.

2.1.3 Constructing relevant graphs of webpages

Any collection of hyperlinked pages can be transformed to a directed graph $\mathscr{G} = (V, \to)$: the nodes correspond to the pages, and if there is a link from page p to page q, there is an arc $p \to q$. Suppose a search query is preformed, specified by a query σ. We wish to determine the authoritative pages by an analysis of the link structure. But first we have to construct a subgraph of the internet on which our algorithm will operate. We want to use the computational effort as efficient as possible, so we restrict the subgraph to the set Q_σ of all pages where the query σ occurs. For this, we could use any already existing text-based search engine. But, for our algorithm Q_σ is possibly much too big: it may contain millions of pages making it impossible for any computer to preform the algorithm. Moreover it is, as explained in the motivation in 2.1.2, possible that Q_σ does not contain some of the most important authorities because they never use the query string σ on their website.

Therefore, we wish to transform the set Q_σ to a set S_σ of pages following this 'wish list' of properties:

1. S_σ is relatively small,
2. S_σ is rich in relevant pages,
3. S_σ contains most of the strongest authorities.

By keeping S_σ small, the computational cost of preforming non-trivial algorithms can be kept under control. By the property of being rich in relevant pages, it will be easier to find good authorities.

To construct S_σ, we first construct a *root set* R_σ with the t highest-ranked pages for σ using a text-based search engine (they sort results based on the occurence of σ). Typically, t is set about 200. R_σ complies with properties 1 and 2 of our wish list, but because $R_\sigma \subset Q_\sigma$, it fails from satisfying property 3. Now we use the root set R_σ to create the set S_σ satisfying our complete wish list. When a strong authority is not in R_σ, it is very likely that at least one of the pages in R_σ points to this authority. Hence, by using the pages in R_σ, we can expand it to S_σ by looking at the links that enter and leave R_σ. We get algorithm 2.

Thus, we obtain S_σ by expanding R_σ to include any page pointed to by a page in R_σ. We also add d pages that points to a page in R_σ. d is usually initiated to 50. The parameter d is crucial to stay in accordance with property 1 of our wish list. Indeed, a webpage can be pointed to by several thousands and thousands of other webpages, and we don't want to include them all if we want to keep S_σ relatively small. Some experiments in [KLEINBERG] showed that this algorithm resulted in a S_σ with a size in the range of 1000 to 5000 web pages. Property 3 of our wish list is usually met because a strong authority need only be reference once in the t pages of the root set R_σ to be added to S_σ.

Denote the resulting graph of the page set S_σ by $\mathscr{G}[S_\sigma]$. Note that $\mathscr{G}[S_\sigma]$ will contain a lot of links serving only navigational purposes within a website. As mentioned before, these links have nothing to do with the the notion of authority and they must be removed from our final graph if we want a good determination of the authoritative pages by an analysis of the link structure. A very simple heuristic can be used to derive a subgraph of $\mathscr{G}[S_\sigma]$ leaving out all the navigational links: we make a distinction between *transverse* links and *intrinsic* links. Transverse links are links between different domain names (e.g. a link between `www.vub.ac.be` and `www.ua.ac.be`) and intrinsic links are links between the same domain name (e.g. a link between `www.vub.ac.be` and `dwis.vub.ac.be`). Intrinsic links exist to allow navigation within a website and they tell us very little about the authority of the pages they point to. Therefore, we delete all intrinsic links from $\mathscr{G}[S_\sigma]$, keeping only the arcs corresponding to transverse links.

Our graph still contains some meaningless links in the context of page authority. Suppose a large number of pages from the same domain name have a transverse link to the same page p. Most of the time, this means a form of advertisement (by example 'Website created by. . . .' at the bottom of each page). It is useful to only allow m pages (m is usually initiated to 6) from the same domain name to have a transverse link to the same page. If m is exceeded, the transverse links must be deleted from the graph. Note, however, that not all links to advertisements will be erased because on most web pages, advertisements change on every page which avoids the exceedance of m.

Applying the two described heuristics above on $\mathscr{G}[S_\sigma]$, we get a new graph $\mathscr{G}'_\sigma$ which is exactly what we need to preform our link analysis.

2.1.4 Hubs and Authorities

A very simple approach would now be to order the pages in $\mathscr{G}'_\sigma$ by their in-degree. Although this approach can sometimes return good search results, this heuristic is often too simple because S_σ will probably contain some web pages with a lot of incoming links without being very relevant to the search query σ (e.g. advertisements). With this incoming links, those

web pages are ranked high in the final search result, which we want to avoid.

Do we have to return to a text-based approach to avoid irrelevant web pages being on top of the search results? No, the link structure of $\mathscr{G}'_\sigma$ can tell us a lot more than it may seem at first glance. Authoritative pages relevant to query σ should indeed have a large in-degree, but there should also be a considerable overlap in the sets of pages that point to authoritative pages. These set of page that point to authoritative pages are called *hubs*. Hubs have links to several authoritative pages and they sort of "concentrate" all the authorities on query σ. Figure 2.1.1 shows what this means conceptually.

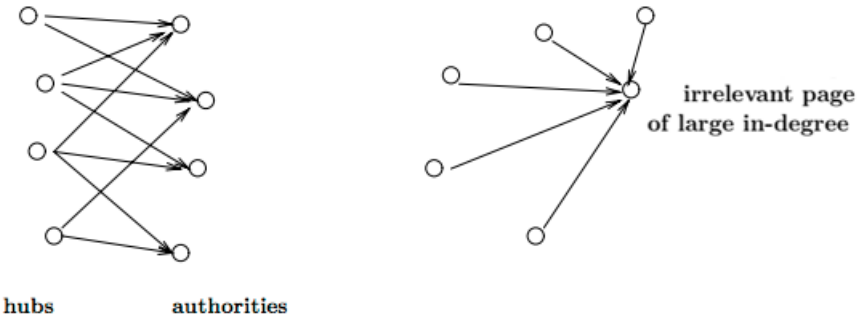


Figure 2.1.1: The concept of hubs and authorities (Source: [KLEINBERG])

So, for each page j we assign two scores, an *authority score* which estimates the value of the content of the page and a *hub score* which estimates the value of the outgoing links to other pages. We get now a *mutually reinforcing relation*: a good hub is a page pointing to many good authorities, a good authority is a page that is pointed to by many good hubs. This leads us to a *mutually reinforcing relation* resulting in an iterative method to break this circularity.

So let $\mathscr{G}'_\sigma = (V, \to)$ and let h_j and a_j be the hub and authority scores of vertex v_j (corresponding with page j). These scores must be initialized by some positive start values and then updated simultaneously for all vertices. This leads to a *mutually reinforcing relation* in which the hub score of v_j is set equal to the authority scores of all vertices pointed to by v_j and in an equal manner the authority score of v_j is set equal to the sum of the hub scores of all vertices pointing to v_j.

$$\begin{cases} h_j := \sum_{i:(v_j,v_i)\in\to} a_i, \\ a_j := \sum_{i:(v_i,v_j)\in\to} h_i. \end{cases}$$

The basic operations in which hubs and authorities reinforce one another are depicted in Figure 2.1.2.

Let B be the adjacency matrix of $\mathscr{G}'_\sigma$ and denote $\mathbf{a}$ as the authority vector with coordinates $(a_1, a_2, \ldots, a_n)$ (with $n = |\mathscr{G}'_\sigma|$, the number of pages) and $\mathbf{h}$ as the hub vector. The mutually reinforcing relation can now be rewritten as:

$$\begin{pmatrix} \mathbf{h} \\ \mathbf{a} \end{pmatrix}_{k+1} = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{h} \\ \mathbf{a} \end{pmatrix}_k, \quad k = 0, 1, \ldots,$$

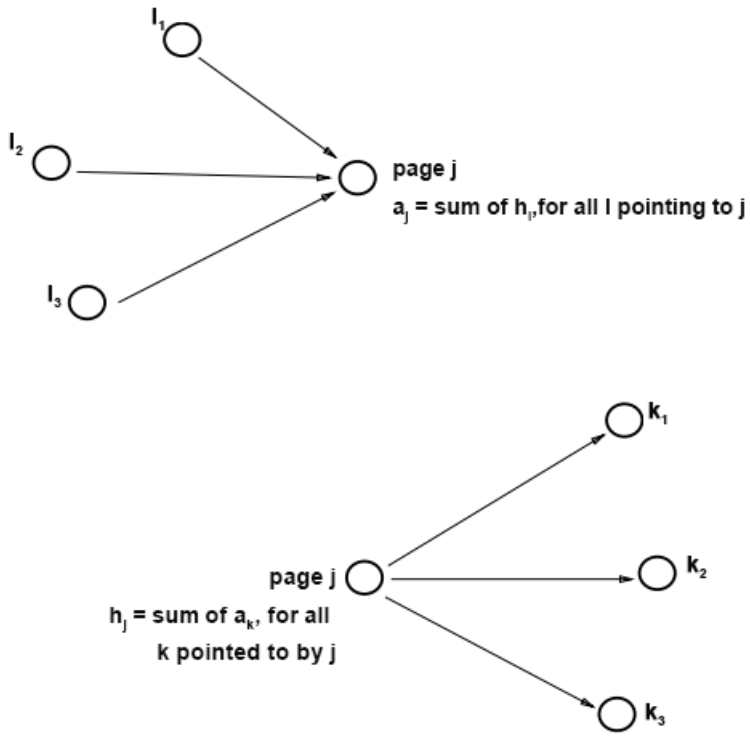


Figure 2.1.2: The basic operations in the reinforcing relation between hubs and authorities (Source: [KLEINBERG])

In compact form, we denote

$$\mathbf{x_{k+1}} = M\mathbf{x_k}, \quad k = 0, 1, \ldots,$$

where

$$\mathbf{x_k} = \begin{pmatrix} \mathbf{h} \\ \mathbf{a} \end{pmatrix}_k, M = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}$$

After each iteration, we have to normalize h_j and a_j. Indeed, we want to get the authority and hub weights for each page and in order to compare these after each iteration step, they must be normalized because only the relative differences do matter, otherwise the whole procedure would be meaningless. Pages with larger a_j-scores are viewed as being better authorities, pages with larger h_j-scores are better hubs.

We get the following sequence (with z_0 some positive start value) of normalized vectors:

$$\mathbf{z}_0 = \mathbf{x}^{(0)} > 0, \mathbf{z}_{k+1} = \frac{M\mathbf{z}_k}{||M\mathbf{z}_k||_2}, \quad k = 0, 1, \ldots, \tag{2.1}$$

How do we decide on $\mathbf{x}^{(0)}$? We will proof in Theorem **??** that any positive vector in $\mathbb{R}^{2n}$ is a good choice, but for the shake of simplicity, we choose the natural choice[1] $\mathbf{1} \in \mathbb{R}^{2n}$. The

[1] $\mathbf{1}$ is a matrix, or vector, whose entries are all equal to 1.

Data:
$\mathscr{G}$: a graph of n linked pages.
k: natural number.
Result: A vector $(\mathbf{h}, \mathbf{a})$ containing the hub and authority scores after k steps.
begin `hits(`$\mathscr{G}$`, k)`
  Set $\mathbf{a}^{(0)} = (1, 1, \ldots, 1) \in \mathbb{R}^n$;
  Set $\mathbf{h}^{(0)} = (1, 1, \ldots, 1) \in \mathbb{R}^n$;
  for $i = 1, 2, \ldots, k$ **do**
    Calculate
    $\mathbf{h}'^{(i)} = \left(\sum_{m:(v_1,v_m)\in\to} \mathbf{a}_m^{(i-1)}, \sum_{m:(v_2,v_m)\in\to} \mathbf{a}_m^{(i-1)}, \ldots, \sum_{m:(v_n,v_m)\in\to} \mathbf{a}_m^{(i-1)}\right)$;
    Normalize $\mathbf{h}'^{(i)}$ obtaining $\mathbf{h}^{(i)}$;
    Calculate $\mathbf{a}'^{(i)} = \left(\sum_{m:(v_m,v_1)\in\to} \mathbf{h}_m^{(i)}, \sum_{m:(v_m,v_2)\in\to} \mathbf{h}_m^{(i)}, \ldots, \sum_{m:(v_m,v_n)\in\to} \mathbf{h}_m^{(i)}\right)$;
    Normalize $\mathbf{a}'^{(i)}$ obtaining $\mathbf{a}^{(i)}$;
  end
  return $(\mathbf{h}^{(k)}, \mathbf{a}^{(k)})$;
end

Algorithm 3: The iterative HITS-algorithm.

Data:
$\mathscr{G}$: a graph of n linked pages.
k: natural number.
c: natural number.
Result: A vector $((\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_c), (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_c))$ containing exactly the nodes of the c top hubs and c top authorities.
begin `filter(`$\mathscr{G}$`, k, c)`
  $(\mathbf{h}, \mathbf{a})$ = `hits(`$\mathscr{G}$`, k)`;
  Sort the pages with the c largest values in $\mathbf{h}$, resulting in a vector of nodes $(\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_c)$;
  Sort the pages with the c largest values in $\mathbf{a}$, resulting in a vector of nodes $(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_c)$;
  return $((\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_c), (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_c))$;
end

Algorithm 4: Returning the top c hubs and authorities

limit to which the sequence converges results in 'definitive' hub and authority scores for each page in the graph $\mathscr{G}'_\sigma$.

To compute the iterative algorithm, we update the hub and authority scores in an alternating form (by each step we have to normalize the scores). Because we will proof that te sequence converges, theoretically we can keep on iterating until a fixed point is reached. But in most practical settings, we choose a fixed number of steps k to reduce the computational cost because we can not know beforehand how large k has to be to reach the limit. But off course, it is extremely important to know that method converges anyway. Let $\mathbf{x}^{(i)}$ denote vector $\mathbf{x}$ at iteration step i as in Notation 1.3.8, and we become Algorithm 3.

To filter the top c hubs and the top c authorities, you can use the trivial algorithm 4.

How do we decide on the values of k and c? It's immediately clear that c and k must be

propertional: for low c values, a lower value for the number of iteration steps k is appropriate and vice versa. Experiments in [KLEINBERG] showed that k set to 20 is sufficient to become stable for finding the 5 best hubs and authorities, thus for $c = 5$.

Instead of using a fixed number of iteration steps k, we already noted that we could change the algorithm 3 in such way that it keeps performing until a fixed point is reached. In other words, for arbitrarily large values of k, the sequence z_k converge to a limit. We now prove that this is true. In fact, the proof calculates this fixed point by solving an eigenvector problem. This makes it even better, because with this result we are not only assured that the `hits`-algorithm converges, but the algorithm could eventually be modified by using any eigenvector algorithm to compute $(\mathbf{h}, \mathbf{a})$. Before we can prove the convergence, note that adjacency matrices are nonnegative by definition, and thus the matrix M is nonnegative too. M is also clearly a symmetric matrix. We first show some very important results about symmetric matrices. Some theorems may perhaps seem a little bit exaggerated for now (some smaller theorems with simpler proofs are suffcient here), but we will also rely on these theorems later on. The basic idea of this series of theorems is as follows: we introduced the power method in 1.3.2 and we now want to show that nonnegative, symmetric matrices like M meet the two condition for this method (namely being a diagonalizable matrix and having a unique dominant eigenvalue). If we showed this, it follows from the power method that z_k converges.

Theorem 2.1.1. *If A is a symmetric, real $n \times n$-matrix, then it has n (not necessarily different) real eigenvalues corresponding to real eigenvectors.*

Proof. First, threat A as complex matrix. The characteristic polynomial $\det(A - \lambda I)$ has n roots in $\mathbb{C}$ and each root is an eigenvalue for A. Let $\lambda \in \mathbb{C}$ be any eigenvalue and $\mathbf{v} \in \mathbb{C}^n$ be a corresponding eigenvector for A. We have:

$$A\mathbf{v} = \lambda\mathbf{v}$$

. As $A = A^t$, we also get:

$$\mathbf{v}^t A = \lambda\mathbf{v}^t.$$

Taking the complex conjugate of both sides we get (A is a real matrix):

$$\bar{\mathbf{v}}^t A = \bar{\lambda}\bar{\mathbf{v}}^t$$

We get:

$$\bar{\mathbf{v}}^t A\mathbf{v} = (\bar{\mathbf{v}}^t A)\mathbf{v} = (\bar{\lambda}\bar{\mathbf{v}}^t)\mathbf{v} = \bar{\lambda}\bar{\mathbf{v}}^t\mathbf{v}.$$

We also have:

$$\bar{\mathbf{v}}^t A\mathbf{v} = \bar{\mathbf{v}}^t(A\mathbf{v}) = \lambda\bar{\mathbf{v}}^t\mathbf{v}.$$

Hence:

$$\bar{\lambda}\bar{\mathbf{v}}^t\mathbf{v} = \lambda\bar{\mathbf{v}}^t\mathbf{v}.$$

We conclude that $\lambda = \bar{\lambda}$ for $\mathbf{v} \neq 0$. We proved that every eigenvalue of A is real. If λ is an eigenvalue of A, then the matrix $(A - \lambda I)$ is not invertible so a vector $\mathbf{s} \in \mathbb{R}^n$ exists with

$$(A - \lambda I)\mathbf{s} = 0,$$

proving that also the corresponding eigenvector is real.

□

Theorem 2.1.2. ***(Symmetric Schur Decomposition)*** *Let A be a real symmetric matrix, then there exist an orthogonal matrix P such that:*

(i) $P^{-1}AP = D$, *a diagonal matrix,*

(ii) The diagonal entries of D are the eigenvalues of A,

(iii) The column vectors of P are the eigenvectors of the eigenvalues of A.

Proof. By induction on the order of the matrix. For $n = 1$ the theorem is trivial. Let A be a symmetric $n \times n$-matrix. A has at least one eigenvalue λ_1 by the previous theorem. Let $\mathbf{x_1}$ be a corresponding eigenvalue with $\|\mathbf{x_1}\| = 1$ and $A\mathbf{x_1} = \lambda_1\mathbf{x_1}$. By the Gram-Schmidt procedure, we construct an orthonormal basis $V_1 = \{\mathbf{x_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$ of $\mathbb{R}^n$. Let:

$$S_1 = [\mathbf{x_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}],$$

since S_1 is orthonormal, we get $S_1^t = S^{-1}$. Consider the matrix: $S_1^{-1}AS_1$. We have:

$$(S_1^{-1}AS_1)^t = (S_1^tAS_1)^t = S_1^tA^tS_1 = S_1^{-1}AS_1$$

Thus $S_1^{-1}AS_1$ is a symmetric matrix. Since $S_1\mathbf{e_1} = \mathbf{x_1}$, we get:

$$\begin{aligned} S_1^{-1}AS_1\mathbf{e_1} &= (S_1^{-1}A)(\mathbf{x_1}) \\ &= S_1^{-1}(\lambda_1\mathbf{x_1}) \\ &= \lambda_1(S_1^{-1}\mathbf{x_1}) \\ &= \lambda_1\mathbf{e_1} \end{aligned}$$

So we get:

$$S_1^{-1}AS_1 = \left(\begin{array}{c|c} \lambda_1 & \mathbf{0} \\ \hline \mathbf{0}^t & \mathbf{A_1} \end{array}\right),$$

with $\mathbf{0}$ a vector of zero entries of size $n-1$ and A_1 an $(n-1) \times (n-1)$ symmetric matrix. We know by induction that there exist a $(n-1) \times (n-1)$ orthogonal matrix S_2 such that $S_2^{-1}A_1S_2 = D'$ with D' an $(n-1) \times (n-1)$ diagonal matrix. Let:

$$S_2' = \left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0}^t & S_2 \end{array}\right),$$

and also S_2' is an orthogonal matrix, we get:

$$\begin{aligned} (S_2')^{-1}\, S_1^{-1}AS_1S_2' &= \left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0}^t & S_2^t \end{array}\right)\left(S_1^{-1}AS_1\right)\left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0}^t & S_2 \end{array}\right) \\ &= \left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0}^t & S_2^t \end{array}\right)\left(\begin{array}{c|c} \lambda_1 & \mathbf{0} \\ \hline \mathbf{0}^t & A_1 \end{array}\right)\left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0}^t & S_2 \end{array}\right) \\ &= \left(\begin{array}{c|c} \lambda_1 & \mathbf{0} \\ \hline \mathbf{0}^t & S_2^tA_1S_2 \end{array}\right) \\ &= \left(\begin{array}{c|c} \lambda_1 & \mathbf{0} \\ \hline \mathbf{0}^t & D' \end{array}\right) \end{aligned}$$

Thus, if we put

$$\begin{aligned} P &= S_1 S_2' \\ D &= \left(\begin{array}{c|c} \lambda_1 & \mathbf{0} \\ \hline \mathbf{0}^t & D' \end{array}\right), \end{aligned}$$

we have proved (1). From the definition of diagonalizable matrices and Theorem 1.3.7 (ii) and (iii) immediately follow. □

De definitie van de perron root wordt best in het hoofdstuk over de Perron-Frobenius geplaatst. Dit moet nog gebeuren

Theorem 2.1.3. *Let M be a symmetric, nonnegative $n \times n$ matrix with spectral radius ρ. Then the the algebraic and geometric multiplicity of the Perron root ρ are both equal to* 1.

Proof. From the previous theorem we know that any symmetric matrix can be permuted to a diagonal matrix consisting of eigenvalues on the diagonal, from the Perron-Frobenius theorem in 1.2.10 we already know that ρ has the algebraic multiplicity 1. From these combined facts it follows that the algebraic and geometric multiplicity of the Perron root ρ is equal to 1. □

Theorem 2.1.4. *Giving a graph $\mathscr{G}$ with n linked pages, the sequence as defined in the previous paragraph:*

$$\mathbf{z}_0 = \mathbf{1} \in \mathbb{R}^n, \mathbf{z}_{k+1} = \frac{M\mathbf{z}_k}{||M\mathbf{z}_k||_2}, \quad k = 0, 1, \ldots$$

, converges.

Proof. Since 1) M is diagonalizable as symmetric matrix by Theorem 2.1.2 and 2) M has a dominant eigenvalue, namely the Perron root ρ by the Perron-Frobenieus theorem in 1.2.10 and an associated eigenvalue wit □

2.1.5 Final reflection

Iets over topic drift vermelden en het feit dat het algoritme specifiek is voor elke query. E.g. `Facebook` is nowadays a universally popular website, almost every website contains a 'like' or 'share' button linking to Facebook, the same goes for other social media websites and some advertisements.

2.2 Similarity on directed graphs

This section summarizes the paper '*A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching*'[BLONDEL] of V. D. Blondel e.a. An overview of the idea and results of the paper is presented, but the proofs are developed far more extensively than they are are in te paper. The paper concludes with a detailed example about the automatic extraction of synonyms in a monolingual dictionary, but this example is left out and replaced by an own example about .... .

2.2.1 Introduction

The method of Kleinberg

The concept of similarity between directed graphs arises as a generalization of hubs and authorities introduced by Kleinberg[KLEINBERG]. Web searching engines like Google are using graphs with vertices and edges that represent the links between pages on the web. To get information out af such a graph, the idea is to identify in a set of pages relevant to a query search, the subset of pages that are good *hubs* and the subset of pages that are good *authorities*. For example, searching for 'super markets', the web-pages of Carrefour, Colruyt,... and other super markets are good authorities, whereas web-pages that point to these home-pages (like Test Aankoop, websites with recipes,...) are good hubs. A good hub points to good authorities and vice versa. The whole idea of Kleinberg was to derive an iterative method that assigns an 'authority score' and a 'hub score' to every vertex of a given graph. These scores can be obtained as the limit of a converging iterative process. We will now describe this iterative process.

ssss Let B be the adjacency matrix of $\mathscr{G}$ and let h and a be the vectors of hub and authority scores. The mutually reinforcing relation can now be rewritten in:

$$\begin{pmatrix} h \\ a \end{pmatrix}_{k+1} = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} h \\ a \end{pmatrix}_k, k = 0, 1, \ldots,$$

In compact form, we denote

$$x_{k+1} = Mx_k, k = 0, 1, \ldots,$$

where

$$x_k = \begin{pmatrix} h \\ a \end{pmatrix}_k, M = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}$$

Because adjacency matrices are nonnegative by definition, the matrix M is nonnegative too. M is also clearly a symmetric matrix. Now we make a sequence of normalized vectors because only the relative scores do matter[2].

$$z_0 = x_0 > 0, z_{k+1} = \frac{Mz_k}{||Mz_k||_2}, k = 0, 1, \ldots \tag{2.2}$$

We are now interested in the limit of the sequence z_k and take this as the definition for the hub and authority scores, but this would be a little bit problematic. The sequence doesn't always converge in the first place. We will prove that sequences associated with symmetric, nonnegative matrices M oscillate between the limits:

$$z_{\text{even}} = \lim_{k\to\infty} z_{2k} \;\&\; z_{\text{odd}} = \lim_{k\to\infty} z_{2k+1}.$$

A second problem is that the limit vectors z_{even} and z_{odd} do in general depend on the initial vector z_0 and that there is no clear natural choice for z_0. The set of all limit vectors obtained when starting from any positive initial vector is given by:

$$Z = \{z_{\text{even}}(z_0), z_{\text{odd}}(z_0) : z_0 > 0\},$$

[2] $||.||_2$ is the Euclidean vector norm.

and we have to select one vector in Z. A good choice is the vector z_{even} obtained for $z_0 = \mathbf{1}$, because it has several nice properties that make it a good choice. First, it is easy to compute and second, it has the extremal property of being the unique vector in Z with the largest possible Manhattan norm[3]. This will also be proved later in this section. Because of these properties, we take the subvectors of $z_{\text{even}}(\mathbf{1})$ as the definitions for the hub and authority scores.

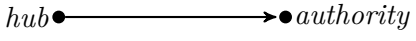
Finally, notice that the second power of the matrix M has the form:

$$M^2 = \begin{pmatrix} BB^T & 0 \\ 0 & B^TB \end{pmatrix},$$

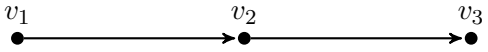
meaning that if the invariant subspaces associated with BB^T and B^TB have dimension 1, then the normalized hub and authority scores are given by the eigenvectors of BB^T and B^TB. Also notice that when the invariant subspace has dimension 1, any positive vector z_0 would give the same result as starting vector $\mathbf{1}$.

Developing a generalization of Kleinbergs method

We introduce the concept of similarity on directed graphs by generalizing the construction of the previous paragraph. The authority score of vertex v_j of a graph $\mathscr{G}$ can be thought of as a similarity score between v_j of $\mathscr{G}$ and vertex *authority* of the graph



and, conversely, the hub score of vertex v_j of $\mathscr{G}$ can be seen as a similarity score between v_j and vertex *hub*. We call the hub-authority graph a *structure graph* for this mutually reinforcing relation. Now the mutually reinforcing updating iteration used in the previous paragraph, can be generalized to graphs with other structure graphs. We start with an example. In our example, we use as structure graph a path graph with three vertices v_1, v_2, v_3.



Let $\mathscr{G}(W, \rightarrow)$ be a graph. With each vertex w_j of $\mathscr{G}$ we now associate three scores x_{i1}, x_{i2} and x_{i3}, one for each vertex of the structure graph. We initialize these score with a positive value and then update them according to the mutually reinforcing relation:

$$\begin{cases} x_{i1} := & & \sum_{j:(w_i,w_j)\in\rightarrow} x_{i2}, \\ x_{i2} := & \sum_{j:(w_j,w_i)\in\rightarrow} x_{i1} & + \sum_{j:(w_i,w_j)\in\rightarrow} x_{i3}, \\ x_{i3} := & & \sum_{j:(w_j,w_i)\in\rightarrow} x_{i2}, \end{cases}$$

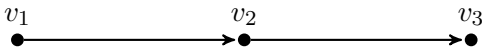
or, in matrix form ($\mathbf{x_j}$ denotes the column vector with entries x_{ij}),

[3]The $||.||_1$-norm or the Manhattan norm of a vector $\mathbf{x}$ is $||\mathbf{x}_1|| = \sum_{i=1}^{n} |x_i|$.

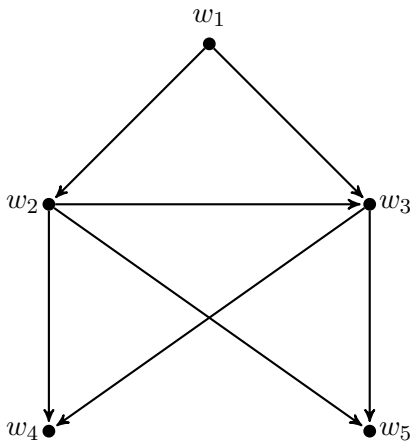
$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_{k+1} = \begin{pmatrix} 0 & B & 0 \\ B^T & 0 & B \\ 0 & B^T & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_k$$

which we again can denote by $x_{k+1} = Mx_k$. The example is now exactly the same as the previous example with hubs and authorities. The matrix M is symmetric and nonnegative, the normalized even and odd iterates converge and the limit $z_{\text{even}}(\mathbf{1})$ is, among all possible limits, the unique vector with the largest possbile Manhattan norm. The three components of the extremal limit $z_{\text{even}}(\mathbf{1})$ are now defined as the *similarity scores* s_1, s_2, s_3 and the *similarity matrix* is defined by $\mathbf{S} = [s_1\ s_2\ s_3]$. We now give a numerical example.

Example 2.2.1. Take as structure graph again the path graph with three vertices v_1, v_2, v_3:



Let $\mathscr{G}(W, \to)$ be the following graph:



Then the adjacency matrix B is:

$$B = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

By using the described mutually reinforcing updating iteration we become the following similarity matrix (a numerical algorithm to calculate this is presented later on in this section together with some proofs that facilitate the calculation):

$$S = \begin{pmatrix} 0.4433 & 0.1043 & 0 \\ 0.2801 & 0.3956 & 0.0858 \\ 0.0858 & 0.3956 & 0.2801 \\ 0.2216 & 0.0489 & 0.2216 \\ 0 & 0.1043 & 0.4433 \end{pmatrix}$$

The similarity score of w_4 with v_2 of the structure graph is equal to 0.0489.

The general case

We now construct the general case. Take two directed graphs $\mathscr{G}(U, \to)$ and $\mathscr{H}(V, \to')$ with $n_{\mathscr{G}}$ and $n_{\mathscr{H}}$ the order of the graphs. We think of $\mathscr{G}$ as the structure graph (such as the graphs hub $\to$ authority and the graph $1 \to 2 \to 3$ in the previous paragraphs). We consider the real scores x_{ij} for $i = 1, \ldots, n_{\mathscr{H}}$ and $j = 1, \ldots, n_{\mathscr{G}}$ and we get the following mutually reinforcing updating iteration with the following updating equations:

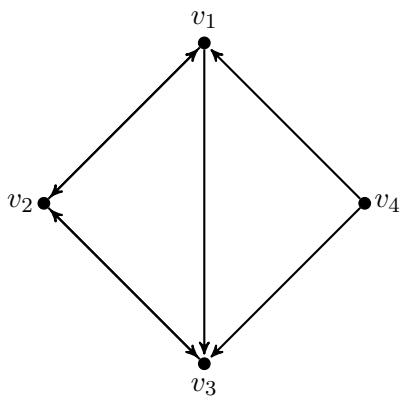
$$x_{ij} := \sum_{r:(v_r,v_i)\in\to',s:(u_s,u_j)\in\to} x_{rs} + \sum_{r:(v_i,v_r)\in\to',s:(u_j,u_s)\in\to} x_{rs}$$

Consider the product graph $\mathscr{G} \times \mathscr{H}$. The above updating equation is equivalent to replacing the scores of all vertices of the product graph by the sum of the scores of the vertices linked by an incoming or outgoing edge. Equation 2.2 can also be rewritten in a a more compact matrix form. Let X_k be the $n_{\mathscr{H}} \times n_{\mathscr{G}}$ matrix of entries x_{ij} at iteration k, and A and B are the adjacency matrices of $\mathscr{G}$ and $\mathscr{H}$. Then the updating equations can be written as:

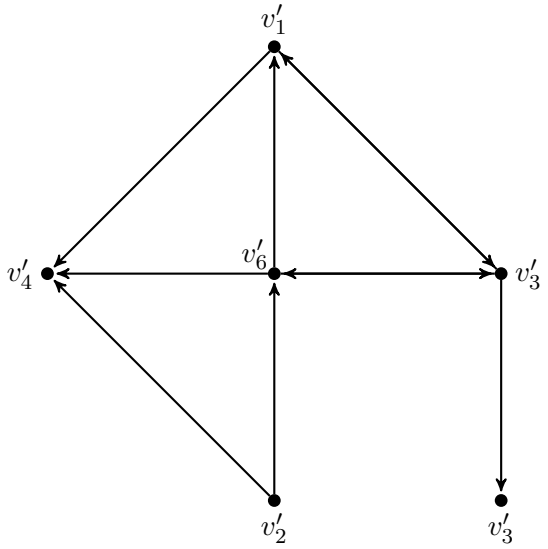
$$X_{k+1} = BX_kA^T + B^TX_kA, \quad k = 0, 1, \ldots, \tag{2.3}$$

We'll prove that the normalized even and odd iterates of this updating equation converge and that the limit $z_{\text{even}}(\mathbf{1})$ is the limit with the largest Manhattan norm. This limit is the definition of the similarity matrix. The following example shows a calculated similarity matrix of two directed graphs.

Example 2.2.2. Let $\mathscr{G}_A(V, \to)$ be the following graph:



Let $\mathscr{G}_B(V', \to')$ be the following graph:



We become the following similarity matrix (a numerical algorithm to calculate this matrix is introduced later in this section):

$$S = \begin{pmatrix} 0.2636 & 0.2786 & 0.2723 & 0.1289 \\ 0.1286 & 0.1286 & 0.0624 & 0.1268 \\ 0.2904 & 0.3115 & 0.2825 & 0.1667 \\ 0.1540 & 0.1701 & 0.2462 & 0 \\ 0.0634 & 0.0759 & 0.1018 & 0 \\ 0.3038 & 0.3011 & 0.2532 & 0.1999 \end{pmatrix}$$

We see for example, that vertex v_2 of $\mathscr{G}_A$ is most similar to vertex v_3' in $\mathscr{G}_B$ because the similarity score s_{32} is the highest among the similarity scores in s_2.

2.2.2 Convergence of the sequence z_k

In the introduction, we depend on the result that the sequence in Equation 2.2 converges for even and odd iterates. We now investigate this hypothesis and prove a theorem about this convergence with the help of the Perron-Frobenius theory of the first chapter. Before we can start with this theorem, we first need some more corollaries of the Perron-Frobenius theorem for matrices that are not only nonnegative, but also symmetric.

Theorem 2.2.3. *Let M be a symmetric nonnegative matrix with spectral radius ρ. Then the algebraic and geometric multiplicity of the Perron root ρ are equal; there is a nonnegative matrix X whose columns span the invariant subspace associated with the Perron root; and the elements of the orthogonal projector Π on the vector space associated with the Perron root of M are all nonnegative.*

Theorem 2.2.4. *Let M be a symmetric nonnegative matrix of spectral radius ρ. Let $z_0 > 0$ and consider the sequence*

$$z_{k+1} = \frac{Mz_k}{||Mz_k||_2}, k = 0, 1, \ldots$$

Two convergence cases can occur depending on whether or not $-rho$ is an eigenvalue of M. When $-\rho$ is not an eigenvalue of M, then the sequence of z_k simply converges to $\frac{\Pi z_0}{||\Pi z_0||_2}$,

where Π *is the orthogonal projector on the invariant subspace associated with the Perron root* ρ*. When* $-\rho$ *is an eigenvalue of* M*, then the subsequences* z_{2k} *and* z_{2k+1} *converge to the limits*

$$z_{even}(z_0) = \lim_{k\to\infty} z_{2k} = \frac{\Pi z_0}{||\Pi z_0||_2} \;\&\; z_{odd}(z_0) = \lim_{k\to\infty} z_{2k+1} = \frac{\Pi M z_0}{||\Pi M z_0||_2}.$$

and the vector $z_{even}(\mathbf{1})$ *is the unique vector of largest possible Manhattan norm in that set.*

2.2.3 Similarity matrices

Bibliography

[EVES] H. Eves, *Elementary Matrix Theory*, Dover Publications, 2012.

[BAPAT] . B. Bapat, T.E.. Raghavan, *Nonnegative Matrices and Applications*, Encyclopedia of Mathematics and its Applications 64, Cambridge University Press, 1997.

[STERN2010] S. Sternberg, *The Perron-Frobenius theorem*, Chapter 9 in Dynamical Systems, Dover Publications, 2010.

[NOUT2008] D. Noutsos, *Perron-Frobenius theory and some extensions* (lecture notes), Department of Mathematics, University of Ioannina, May 2008.

[VARGA] . S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, 1962.

[1] . A. Brualdi, H. J. Ryser, *Combinatorial Matrix Theory*, Cambridge University Press, 1991

[2] . Lancaster, M. Tismenetsky, *The Theory of Matrices: With Applications*, Academic Press, 1985

[MEU2011] W. De Meuter, *Algoritmen en Datastructuren I*, Vrije Universiteit Brussel, 2011.

[KIEBOOM2011] R. Kieboom, *Aanvulling Lineaire Algebra*, Vrije Universiteit Brussel, 2011.

[KIEBOOM2011] R. Kieboom, *Aanvulling Lineaire Algebra*, Vrije Universiteit Brussel, 2011.

[ADHEMAR] . Bultheel, *Inleidng tot de numerieke wiskunde*, Acco, 2006.

[BIN1977] K.G. Binmore, *Mathematical Analysis: A Straightforward Approach*, Cambridge Universitiy Press, 1977.

[GOLUB] G. Golub, C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

[CARA] P. Cara, *Discrete Wiskunde*, Vrije Universiteit Brussel - Dienst Uitgaven, 2011

[LIFSHITS] Y. Lifshits, *Four Results of Jon Klienberg: A Talk for St. Petersburg Mathematical Society*, Steklov Insitute of Mathematics at St. Petersburg, 2007.

[PAGE] S. Brin, L. Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, Seventh International World-Wide Web Conference, 1998.

[BLONDEL] V. D Blondel, A. Gajardo, M. Heymans, P. Senellart, P. Van Dooren, *A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching*, Society for Industrial and Applied Mathematics, 2004.

[KLEINBERG] J. M. Kleinberg, *Authoritative sources in a hyperlinked environment*, J. ACM, 1999.