Semirings

Matrices over semirings

Networks and matrices

Real matrices

Markov chains

Resources

Network Analysis 2
Statistical Approaches and Modeling

Networks and matrices

Vladimir Batagelj

IMFM Ljubljana, IAM UP Koper, and NRU HSE Moscow

XIII Summer School of the ANR-Lab
Network Analysis and Contemporary Decision Sciences
International Laboratory for Applied Network Research
NRU HSE, Moscow, August 2022

Outline

Semirings

Matrices over semirings

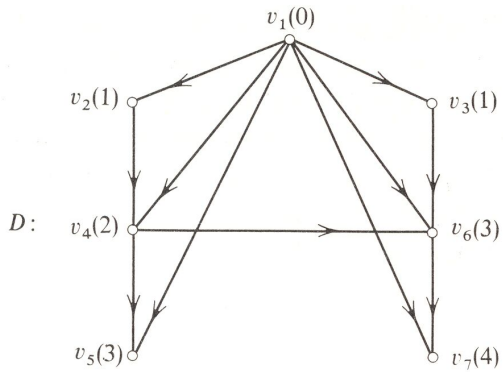
Networks and matrices

Real matrices

Markov chains

Resources

1 Semirings
2 Matrices over semirings
3 Networks and matrices
4 Real matrices
5 Markov chains
6 Resources



$$N(D) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array} \begin{array}{c} \begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ \infty & 0 & \infty & 1 & 2 & 2 & 3 \\ \infty & \infty & 0 & \infty & \infty & 1 & 2 \\ \infty & \infty & \infty & 0 & 1 & 1 & 2 \\ \infty & \infty & \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix} \end{array} \begin{array}{c} a_i \\ 6 \\ 8 \\ 3 \\ 4 \\ 0 \\ 1 \\ 0 \end{array}$$

F. Harary &: Structural models. JW 1965, p. 273

Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Current version of slides (August 17, 2022 at 03 : 54): slides PDF

Semirings

Let $\mathbb{K}$ be a set and a, b, c elements from $\mathbb{K}$. A *semiring* is an algebraic structure $(\mathbb{K}, \oplus, \odot, 0, 1)$ with two binary operations (addition $\oplus$ and multiplication $\odot$) where:

- $(\mathbb{K}, \oplus, 0)$ is an abelian monoid with the neutral element 0 (zero):

$$a \oplus b = b \oplus a \quad \text{commutativity}$$
$$(a \oplus b) \oplus c = a \oplus (b \oplus c) \quad \text{associativity}$$
$$a \oplus 0 = a \quad \text{existence of zero}$$

- $(\mathbb{K}, \odot, 1)$ is a monoid with the neutral element 1 (unit):

$$(a \odot b) \odot c = a \odot (b \odot c) \quad \text{associativity}$$
$$a \odot 1 = 1 \odot a = a \quad \text{existence of a unit}$$

- Multiplication $\odot$ distributes over addition $\oplus$:

$$a \odot (b \oplus c) = a \odot b \oplus a \odot c$$
$$(b \oplus c) \odot a = b \odot a \oplus c \odot a$$

In formulas we assume precedence of multiplication over addition.

. . . Semirings

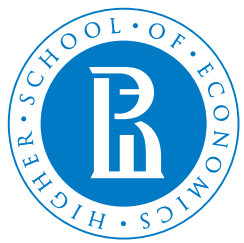
A semiring $(\mathbb{K}, \oplus, \odot, 0, 1)$ is *complete* iff the addition is well defined for countable sets of elements and the commutativity, associativity, and distributivity hold in the case of countable sets.

These properties are generalized in this case; for example, the distributivity takes the form

$$\left(\bigoplus_i a_i\right) \odot \left(\bigoplus_j b_j\right) = \bigoplus_i \left(\bigoplus_j (a_i \odot b_j)\right) = \bigoplus_{i,j} (a_i \odot b_j).$$

The addition is *idempotent* iff $a \oplus a = a$ for all $a \in \mathbb{K}$. In this case the semiring over a finite set $\mathbb{K}$ is complete.

The *power* a^n, $n \in \mathbb{N}$ of an element $a \in \mathbb{K}$ is defined by $a^0 = 1$ and $a^{n+1} = a^n \odot a$ for $n \geq 0$.



...Semirings

A semiring $(\mathbb{K}, \oplus, \odot, 0, 1)$ is *closed* iff for the additional (unary) *closure* operation $*$ it holds for all $a \in \mathbb{K}$:

$$a^* = 1 \oplus a \odot a^* = 1 \oplus a^* \odot a.$$

A *strict closure* is defined as $\bar{a} = a \odot a^*$.

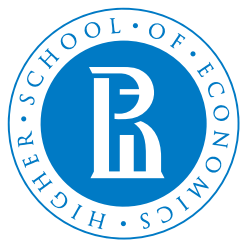
Different closures over the same semiring can exist. A complete semiring is always closed for the closure

$$a^* = \bigoplus_{k \in \mathbb{N}} a^k.$$

In a semiring $(\mathbb{K}, \oplus, \odot, 0, 1)$ the *absorption law* holds iff for all $a, b, c \in \mathbb{K}$:

$$a \odot b \oplus a \odot c \odot b = a \odot b.$$

Because of the distributivity, it is sufficient for the absorption law to check the property $1 \oplus c = 1$ for all $c \in \mathbb{K}$.



Examples of semirings

Combinatorial Semiring $(\mathbb{N}, +, \cdot, 0, 1)$
This is the most commonly used semiring. Also some other sets are used: $\mathbb{R}, \mathbb{R}_0^+, \mathbb{Q}$. For $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$, the semiring is closed for $a^* = \sum_{k \in \overline{\mathbb{N}}} a^k$ because it is a complete semiring. An example of a closure for $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ is $a^* = 1/(1-a)$ for $a \neq 1, \infty$ and $0^* = 1$, $1^* = \infty$, and $\infty^* = \infty$. It is commutative. It is not idempotent.

Reachability Semiring $(\{0, 1\}, \vee, \wedge, 0, 1)$
The logical (boolean, reachability) semiring is suitable for solving the connectivity questions in networks. The multiplication is commutative and the absorption law holds. It is closed for $a^* = 1 \vee a \wedge a^* = 1$.

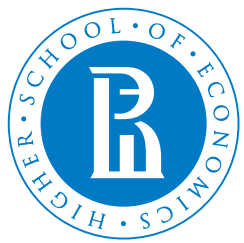
Shortest Paths Semiring $(\overline{\mathbb{R}}_0^+, \min, +, \infty, 0)$
The semiring is commutative and closed for $a^* = \min(0, a + a^*) = 0$ (0 is the smallest element in the set $\mathbb{R}_0^+$). Since $\min(0, a) = 0$, the absorption law also holds. For the set $\overline{\mathbb{N}}$, the semiring is called a *tropical* semiring. Another set is $\overline{\mathbb{R}}$ and in this case the semiring is isomorphic ($x \to -x$) to *max-plus* semiring $(\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$.

. . . Examples of semirings

Pathfinder Semiring $(\overline{\mathbb{R}}_0^+, \min, \boxed{\mathrm{r}}, \infty, 0)$
The multiplicative operation is the *Minkowski operation* $a \boxed{\mathrm{r}} b = \sqrt[r]{a^r + b^r}$. This semiring is closed for $a^* = 0$ and the absorption law holds in it.

In Pathfinder algorithm the value r for the Minkowski operation is selected according to a dissimilarity measure. For a value $r = 1$, the semiring is the shortest path semiring, and for a value $r = \infty$, the semiring is the min-max semiring.

```
> r = 2
> '%Pf+%' <- function(a,b){return(min(a,b))}
> '%Pf*%' <- function(a,b){return((a^r+b^r)^(1/r))}
> 3 %Pf+% 4
[1] 3
> 3 %Pf*% 4
[1] 5
> Pf0 <- Inf
> Pf1 <- 0
```

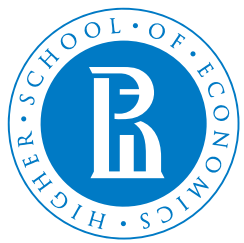


Matrices over semirings

An $m \times n$ matrix $\mathbf{A} = [a_{ij}]$ over a set $\mathbb{K}$ is a rectangular array of elements from the set $\mathbb{K}$ that consists of m *row*s and n *column*s. The entry in the i-th row and j-th column is denoted by a_{ij}. If $m = n$ the matrix $\mathbf{A}$ is called a *square* matrix. The matrix with all entry values equal to 0 is called the *zero* matrix and is denoted by $\mathbf{0}_{mn}$.

The *transpose* of a matrix $\mathbf{A}$ is a matrix $\mathbf{A}^T$ in which the rows of $\mathbf{A}$ are written as the columns of $\mathbf{A}^T$: $a^T_{ij} = a_{ji}$. A square matrix $\mathbf{A}$ is *symmetric* if $\mathbf{A} = \mathbf{A}^T$.

A *diagonal matrix* is a square matrix $\mathbf{A}$ such that only diagonal elements are nonzero: $a_{ij} = 0$, for $i \neq j$. If $a_{ii} = 1, i = 1, \ldots, n$, a diagonal matrix is called the *identity* matrix I_n of order n. A square matrix $\mathbf{A}$ is *upper triangular* if $a_{ij} = 0, i > j$, and its transpose is a *lower triangular* matrix.



. . . Matrices over semirings

Let $\mathcal{M}_{mn}(\mathbb{K})$ be a set of matrices of order $m \times n$ over the semiring $(\mathbb{K}, \oplus, \odot, 0, 1)$ in which we additionally require

$$\forall a \in \mathbb{K} : a \odot 0 = 0 \odot a = 0,$$

and let $\mathcal{M}(\mathbb{K})$ be a set of all matrices over the $\mathbb{K}$.

The operations $\oplus$ and $\odot$ can be extended to the $\mathcal{M}(\mathbb{K})$:

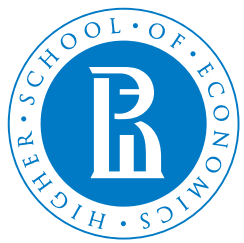
$$\mathbf{A}, \mathbf{B} \in \mathcal{M}_{mn}(\mathbb{K}) : \mathbf{A} \oplus \mathbf{B} = [a_{uv} \oplus b_{uv}] \in \mathcal{M}_{mn}(\mathbb{K})$$
$$\mathbf{A} \in \mathcal{M}_{mk}(\mathbb{K}), \mathbf{B} \in \mathcal{M}_{kn}(\mathbb{K}) : \mathbf{A} \odot \mathbf{B} = [\bigoplus_{t=1}^{k} a_{ut} \odot b_{tv}] \in \mathcal{M}_{mn}(\mathbb{K}).$$

Then:

- $(\mathcal{M}_{mn}(\mathbb{K}), \oplus, \mathbf{0}_{mn})$ is an abelian monoid.
- $(\mathcal{M}_{n^2}(\mathbb{K}), \odot, \mathbf{I}_n)$ is a monoid.
- $(\mathcal{M}_{n^2}(\mathbb{K}), \oplus, \odot, \mathbf{0}_n, \mathbf{I}_n)$ is a semiring.

For matrices $\mathbf{A}$ and $\mathbf{B}$, it holds

$$(\mathbf{A} \odot \mathbf{B})^T = \mathbf{B}^T \odot \mathbf{A}^T.$$

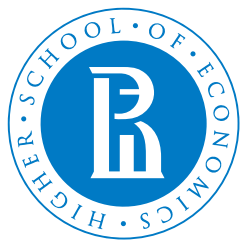


Networks and matrices

Let $\mathcal{N} = ((\mathcal{I}, \mathcal{J}), \mathcal{A}, w)$ be a *simple two-mode network*, where $\mathcal{I}$ and $\mathcal{J}$ are (sub)sets of nodes, $\mathcal{V} = \mathcal{I} \cup \mathcal{J}$. We assume that the set of nodes is finite $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$. $\mathcal{A} \subseteq \mathcal{I} \times \mathcal{J}$is a set of arcs (directed links) linking $\mathcal{I}$ and $\mathcal{J}$, and the mapping $w : \mathcal{A} \to \mathbb{K}$ is the *arcs value function* also called a *weight*. We can assign to a network its *value matrix* $\mathbf{W} = [w_{ij}]$ with elements

$$w_{ij} = \begin{cases} w((i,j)) & (i,j) \in \mathcal{A} \\ 0 & \text{otherwise.} \end{cases}$$

The problem with value matrices in computer applications is their size. The value matrices of large networks are sparse. There is no need to store the zero values in a matrix, and different data structures can be used for saving and working with value matrices: special dictionaries and lists.

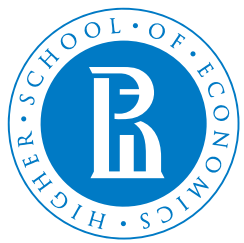


Network multiplication

Let $\mathcal{N}_{\mathbf{A}} = ((\mathcal{I}, \mathcal{K}), \mathcal{A}_{\mathbf{A}}, w_{\mathbf{A}})$ and $\mathcal{N}_{\mathbf{B}} = ((\mathcal{K}, \mathcal{J}), \mathcal{A}_{\mathbf{B}}, w_{\mathbf{B}})$ be a pair of networks with corresponding matrices $\mathbf{A}_{\mathcal{I}\times\mathcal{K}}$ and $\mathbf{B}_{\mathcal{K}\times\mathcal{J}}$, respectively. Assume also that $w_{\mathbf{A}} : \mathcal{A}_{\mathbf{A}} \to \mathbb{K}$, $w_{\mathbf{B}} : \mathcal{A}_{\mathbf{B}} \to \mathbb{K}$, and $(\mathbb{K}, \oplus, \odot, 0, 1)$ is a semiring. We say that such networks/matrices are *compatible*.

The *product* $\mathcal{N}_{\mathbf{A}} \star \mathcal{N}_{\mathbf{B}}$ of networks $\mathcal{N}_{\mathbf{A}}$ and $\mathcal{N}_{\mathbf{B}}$ is a network $\mathcal{N}_{\mathbf{C}} = ((\mathcal{I}, \mathcal{J}), \mathcal{A}_{\mathbf{C}}, w_{\mathbf{C}})$ for $\mathcal{A}_{\mathbf{C}} = \{(i,j); i \in \mathcal{I}, j \in \mathcal{J}, c_{ij} \neq 0\}$ and $w_{\mathbf{C}}((i,j)) = c_{ij}$ for $(i,j) \in \mathcal{A}_{\mathbf{C}}$, where $\mathbf{C} = [c_{ij}] = \mathbf{A} \odot \mathbf{B}$.

If all three sets of nodes are the same ($\mathcal{I} = \mathcal{K} = \mathcal{J}$), we are dealing with ordinary one-mode networks (square matrices).



Network multiplication

When do we get an arc in the product network?

$$c_{ij} = \bigoplus_{k \in \mathcal{K}} a_{ik} \odot b_{kj}.$$

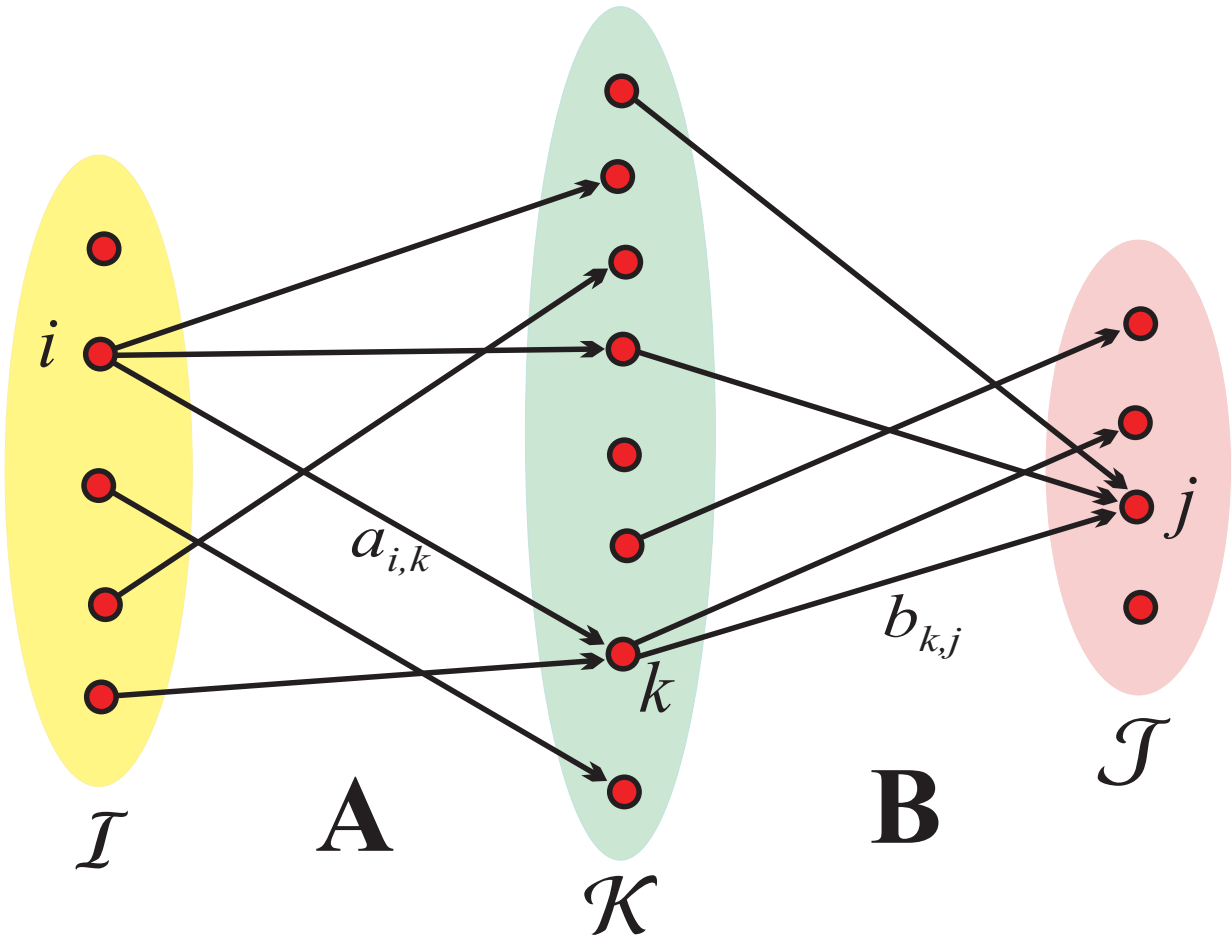
There is an arc $(i, j) \in \mathcal{A}_{\mathbf{C}}$ iff c_{ij} is nonzero. Therefore at least one term $a_{ik} \odot b_{kj}$ is nonzero, but this means that both a_{ik} and b_{kj} should be nonzero, and thus $(i, k) \in \mathcal{A}_{\mathbf{A}}$ and $(k, j) \in \mathcal{A}_{\mathbf{B}}$ (see figure):

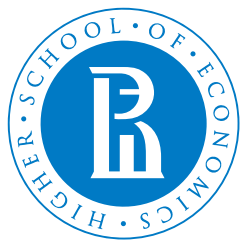
$$c_{ij} = \bigoplus_{k \in N_{\mathbf{A}}(i) \cap N_{\mathbf{B}}^{-}(j)} a_{ik} \odot b_{kj},$$

where $N_{\mathbf{A}}(i)$ are the *successors* of node i in the network $\mathcal{N}_{\mathbf{A}}$ and $N_{\mathbf{B}}^{-}(j)$ are the *predecessors* of node j in the network $\mathcal{N}_{\mathbf{B}}$. The value of the entry c_{ij} equals to the value of all paths (of length 2) from $i \in \mathcal{I}$ to $j \in \mathcal{J}$ passing through some node $k \in \mathcal{K}$.

Multiplication is about "traveling" on network.

Network multiplication



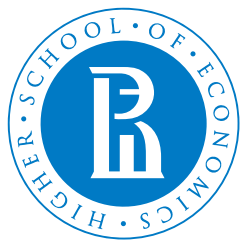


Network multiplication

The standard procedure to compute the product of matrices $\mathbf{A}_{\mathcal{I}\times\mathcal{K}}$ and $\mathbf{B}_{\mathcal{K}\times\mathcal{J}}$ has the complexity $O(|\mathcal{I}| \cdot |\mathcal{K}| \cdot |\mathcal{J}|)$ and is therefore too slow to be used for large networks. Since the matrices of large networks are usually sparse, we can compute the product of two networks much faster considering only nonzero entries:

for $k \in \mathcal{K}$ **do**
  for $i \in N^-_{\mathbf{A}}(k)$ **do**
    for $j \in N_{\mathbf{B}}(k)$ **do**
      if $\exists c_{ij}$ **then**
        $c_{ij} = c_{ij} \oplus a_{ik} \odot b_{kj}$
      else
        $c_{ij} = a_{ik} \odot b_{kj}$

In general the multiplication of large sparse networks is a “dangerous” operation since the result can “explode” – it is not sparse. In many cases also the result is sparse [3].



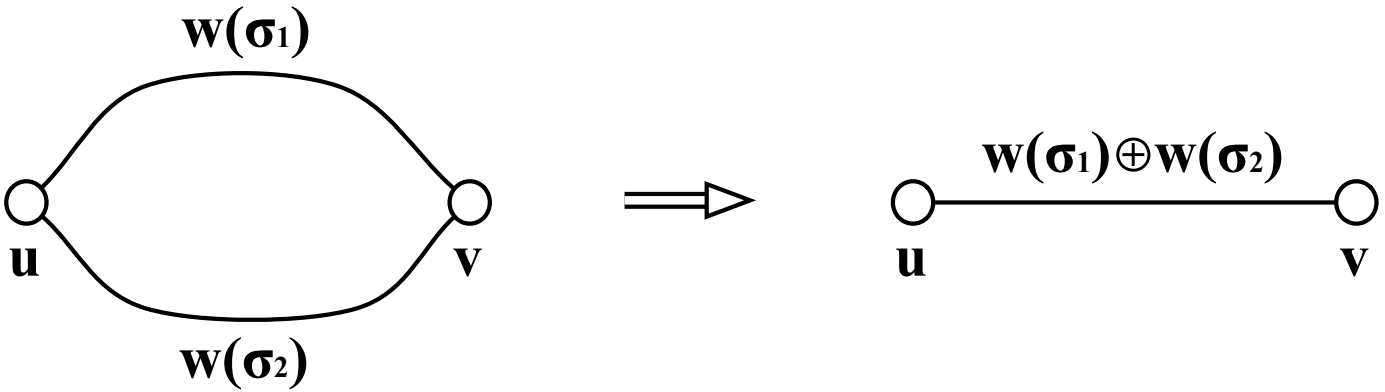
The Algebraic Path Problem

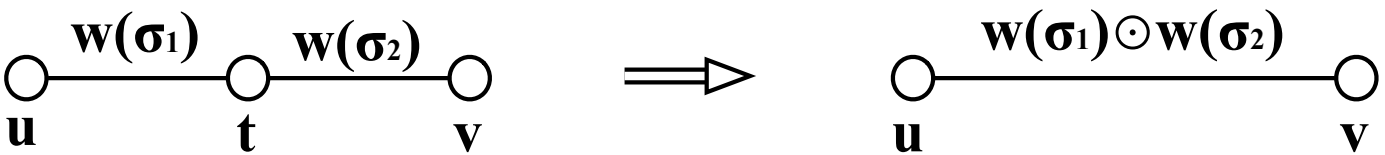
The use of a special semiring and a multiplication of networks can lead us to the essence of the shortest path problem [1]. Many other network problems can be solved by replacing the usual addition and multiplication with the corresponding operations from an appropriate semiring.

Let $\mathcal{N} = (\mathcal{V}, \mathcal{A}, w)$ be a network where $w : \mathcal{A} \to \mathbb{K}$ is the value (weight) of arcs such that $(\mathbb{K}, \oplus, \odot, 0, 1)$ is a semiring. We denote the number of nodes as $n = |\mathcal{V}|$ and the number of arcs as $m = |\mathcal{A}|$.

A finite sequence of nodes $\sigma = (u_0, u_1, u_2, \ldots, u_{p-1}, u_p)$ is a *walk* of *length* p on $\mathcal{N}$ iff every pair of neighboring nodes is linked: $(u_{i-1}, u_i) \in \mathcal{A}$, $i = 1, \ldots, p$. A finite sequence σ is a *semiwalk* or chain on $\mathcal{N}$ iff every pair of neighboring nodes is linked neglecting the direction of an arc: $(u_{i-1}, u_i) \in \mathcal{A} \vee (u_i, u_{i-1}) \in \mathcal{A}$, $i = 1, \ldots, p$. A (semi)walk is *closed* iff its end nodes coincide: $u_0 = u_p$. A walk is *simple* or a *path* iff no node repeats in it. A closed walk with different nodes, except first and last, is called a *cycle*.

Semiring operations and values of walks





Extending the weight

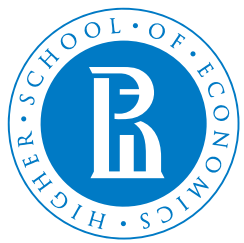
We can extend the weight w to walks and sets of walks on $\mathcal{N}$ by the following rules (see figure):

- Let $\sigma_v = (v)$ be a null walk in the node $v \in \mathcal{V}$; then $w(\sigma_v) = 1$.
- Let $\sigma = (u_0, u_1, u_2, \ldots, u_{p-1}, u_p)$ be a walk of length $p \geq 1$ on $\mathcal{N}$; then

$$w(\sigma) = \bigodot_{i=1}^{k} w(u_{i-1}, u_i).$$

- For empty set of walks $\emptyset$ it holds $w(\emptyset) = 0$.
- Let $S = \{\sigma_1, \sigma_2, \ldots\}$ be a set of walks in $\mathcal{N}$; then

$$w(S) = \bigoplus_{\sigma \in S} w(\sigma).$$



Concatenation

Let σ_1 and σ_2 be *compatible* walks on $\mathcal{N}$: the end node of the walk σ_1 is equal to the start node of the walk σ_2. Such walks can be concatenated in a new walk $\sigma_1 \circ \sigma_2$ for which holds

$$w(\sigma_1 \circ \sigma_2) = \begin{cases} w(\sigma_1) \odot w(\sigma_2) & \sigma_1 \text{ and } \sigma_2 \text{ are compatible} \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be finite sets of walks; then

$$w(\mathcal{S}_1 \cup \mathcal{S}_2) \oplus w(\mathcal{S}_1 \cap \mathcal{S}_2) = w(\mathcal{S}_1) \oplus w(\mathcal{S}_2).$$

In the special case when $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$, it holds $w(\mathcal{S}_1 \cup \mathcal{S}_2) = w(\mathcal{S}_1) \oplus w(\mathcal{S}_2)$. Also the *concatenation* of walks can be generalized to sets of walks:

$$\mathcal{S}_1 \circ \mathcal{S}_2 = \{\sigma_1 \circ \sigma_2 : \sigma_1 \in \mathcal{S}_1,\ \sigma_2 \in \mathcal{S}_2,\ \sigma_1 \text{ and } \sigma_2 \text{ are compatible}\}.$$

It also holds $\mathcal{S} \circ \emptyset = \emptyset \circ \mathcal{S} = \emptyset$.

Sets of walks

We denote by:

- $\mathcal{S}_{uv}^{k}$ the set of all walks of length k from node u to node v
- $\mathcal{S}_{uv}^{(k)}$ the set of all walks of length at most k from node u to node v
- $\mathcal{S}_{uv}^{*}$ the set of all walks from node u to node v
- $\overline{\mathcal{S}}_{uv}$ the set of all nontrivial walks from node u to node v
- $\mathcal{E}_{uv}$ the set of all simple walks (paths) from node u to node v

The following relations hold among these sets:

$$\mathcal{S}_{uv}^{k} \subseteq \mathcal{S}_{uv}^{(k)} \subseteq \mathcal{S}_{uv}^{*}$$

$$k \neq l \;\Leftrightarrow\; \mathcal{S}_{uv}^{k} \cap \mathcal{S}_{uv}^{l} = \emptyset$$

$$\mathcal{S}_{uv}^{(k)} = \bigcup_{i=0}^{k} \mathcal{S}_{uv}^{i}, \quad \mathcal{S}_{uv}^{*} = \bigcup_{k=0}^{\infty} \mathcal{S}_{uv}^{k}$$

$$k \geq n-1 : \mathcal{E}_{uv} \subseteq \mathcal{S}_{uv}^{(k)}$$

$$w(\mathcal{S}_{uv}^{(k)}) = \sum_{i=0}^{k} w(\mathcal{S}_{uv}^{i}).$$

Unique factorization

A set of walks $\mathcal{S}$ is *uniquely factorizable* to sets of walks $\mathcal{S}_1$ and $\mathcal{S}_2$ iff $\mathcal{S} = \mathcal{S}_1 \circ \mathcal{S}_2$, and for all walks $\sigma_1, \sigma_1' \in \mathcal{S}_1$, $\sigma_2, \sigma_2' \in \mathcal{S}_2$, $\sigma_1 \neq \sigma_1'$, $\sigma_2 \neq \sigma_2'$, it holds $\sigma_1 \circ \sigma_2 \neq \sigma_1' \circ \sigma_2'$.

For example, for s, $0 < s < k$, a nonempty set $\mathcal{S}^k_{uv}$ is uniquely factorizable to sets $\mathcal{S}^s_{u\bullet}$ and $\mathcal{S}^{k-s}_{\bullet v}$, where $\mathcal{S}^s_{u\bullet} = \bigcup_{t \in \mathcal{V}} \mathcal{S}^s_{ut}$, etc.

Theorem: Let the finite set $\mathcal{S}$ be uniquely factorizable to $\mathcal{S}_1$ and $\mathcal{S}_2$ or a semiring is idempotent. Then it holds

$$w(\mathcal{S}_1 \circ \mathcal{S}_2) = w(\mathcal{S}_1) \odot w(\mathcal{S}_2).$$

□

The k-th power $\mathbf{W}^k$ of a square matrix $\mathbf{W}$ over $\mathbb{K}$ is unique because of associativity.

The k-th power of value matrix

Theorem: The entry w_{uv}^k of k-th power $\mathbf{W}^k$ of a value matrix $\mathbf{W}$ is equal to the value of all walks of length k from node u to node v:

$$w(\mathcal{S}_{uv}^k) = \mathbf{W}^k[u,v] = w_{uv}^k.$$

□

Attention, w_{ij}^k **is not** the k-th power of w_{ij} !!!

Proof by induction. It holds for $k = 0$ and $k = 1$.
Induction step: $k \Rightarrow k+1$. Let K denote the set of indices of nodes that can be reached from the node v_i by a walk of length k and an arc leads from them to the node v_j. If $K = \emptyset$ then also $\mathcal{S}_{ij}^{k+1} = \emptyset$ and therefore $w_{ij}^{k+1} = 0$. In the other case $K \neq \emptyset$. Because each walk of length $k+1$ is uniquely factorizable to a walk of length k and walk of length 1 we have:

$$\begin{aligned} w(\mathcal{S}_{ij}^{k+1}) &= \sum_{t \in K} w(\mathcal{S}_{i(t)j}^{k+1}) = \sum_{t \in K} w(\mathcal{S}_{it}^k \circ \mathcal{S}_{tj}^1) = \\ &= \sum_{t \in K} w(\mathcal{S}_{it}^k) \cdot w(\mathcal{S}_{tj}^1) = \sum_{t \in K} w_{it}^k \cdot w_{tj} = \sum_{t=1}^{n} w_{it}^k \cdot w_{tj} = w_{ij}^{k+1} \end{aligned}$$

□

Matrices and sets of walks

Therefore if a network $\mathcal{N}$ is acyclic, then it holds for a value matrix $\mathbf{W}$:

$$\exists k_0 : \forall k > k_0 : \mathbf{W}^k = \mathbf{0},$$

k_0 is the length of the longest walk in the network.

If $\mathbf{W}$ is the network adjacency matrix over the combinatorial semiring, the entry w_{uv}^k counts the number of different walks of length k from u to v.

Let us denote

$$\mathbf{W}^{(k)} = \bigoplus_{i=0}^{k} \mathbf{W}^i.$$

In an idempotent semiring, it holds $\mathbf{W}^{(k)} = (\mathbf{1} \oplus \mathbf{W})^k$.

Theorem: $w(S_{uv}^{(k)}) = \mathbf{W}^{(k)}[u, v] = w_{uv}^{(k)}$. □

For the combinatorial semiring and the network adjacency matrix $\mathbf{W}$, the entry $w_{uv}^{(k)}$ counts the number of different walks of length at most k from u to v.

. . . Matrices and sets of walks

The matrix semiring over a complete semiring is also complete and therefore closed for $\mathbf{W}^* = \bigoplus_{k=0}^{\infty} \mathbf{W}^k$.

Theorem: For a value matrix $\mathbf{W}$ over a complete semiring with closure $\mathbf{W}^*$ and strict closure $\overline{\mathbf{W}}$ hold:

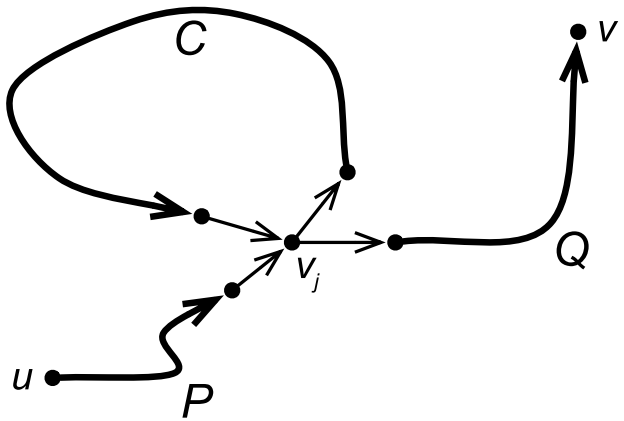
$$w(\mathcal{S}^*_{uv}) = \mathbf{W}^*[u,v] = w^*_{uv} \qquad \text{and}$$
$$w(\overline{\mathcal{S}}_{uv}) = \overline{\mathbf{W}}[u,v] = \overline{w}_{uv}.$$

□

For the reachability semiring and the network adjacency matrix $\mathbf{W}$, the matrix $\overline{\mathbf{W}}$ is its transitive closure.

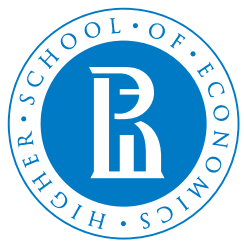
For the shortest paths semiring and the network value matrix $\mathbf{W}$, the entry w^*_{uv} is the value of the shortest path from u to v.

Each walk contains path



Let $(\mathbb{K}, \oplus, \odot, 0, 1)$ be an absorptive semiring and σ be a nonsimple walk from a set $\mathcal{S}^*_{uv}$. Therefore at least one node v_j appears more than once in σ. The part of a walk between its first and last appearance is a closed walk C (see figure). The whole walk can be written as $\sigma = P \circ C \circ Q$ where P is the initial segment of σ from u to the first appearance of v_j, and Q is the terminal segment of σ from the last appearance of v_j to v. Note that $P \circ Q$ is also a walk. The value of both walks together is $w(\{P \circ Q, P \circ C \circ Q\}) = w(P \circ Q)$. We see that the walks that are not paths do not contribute to the value of walks. Therefore $w(\mathcal{S}^*_{uv}) = w(\mathcal{E}_{uv})$. This equality holds also for $\mathcal{S}^*_{uv} = \emptyset$.

Since the node set $\mathcal{V}$ is finite, also the set $\mathcal{E}_{uv}$ is finite which allows us to compute the value $w(\mathcal{S}^*_{uv})$. We already know that $\mathbf{W}^* = \mathbf{W}^{(k)} = (\mathbf{1} \oplus \mathbf{W})^k$ for k large enough.



Fletcher's algorithm

To compute the closure matrix $\mathbf{W}^*$ of a given matrix $\mathbf{W}$ over a complete semiring $(\mathbb{K}, \oplus, \odot, 0, 1)$, we can use the Fletcher's algorithm [8]:

$\mathbf{C}_0 = \mathbf{W}$
for $k = 1, \ldots, n$ **do**
  for $i = 1, \ldots, n$ **do**
    for $j = 1, \ldots, n$ **do**
      $c_k[i,j] = c_{k-1}[i,j] \oplus c_{k-1}[i,k] \odot (c_{k-1}[k,k])^* \odot c_{k-1}[k,j]$
    $c_k[k,k] = 1 \oplus c_k[k,k]$
$\mathbf{W}^* = \mathbf{C}_n$

If we delete the statement $c_k[k,k] = 1 \oplus c_k[k,k]$, we obtain the algorithm for computing the strict closure $\overline{\mathbf{W}}$. If the addition $\oplus$ is idempotent, we can compute the closure matrix in place – we omit the subscripts in matrices $\mathbf{C}_k$. If the absorption law holds we have further $a^\star = 1$. In general case we need two matrices.

The Fletcher's algorithm is a generalization of a sequence of algorithms (Kleene, Warshall, Floyd, Roy) for computing closures on specific semirings.

Betweenness

Important are also the nodes that can control the information flow in the network. If we assume that this flow uses only the shortest paths (geodesics) we get a measure of *betweenness* (Anthonisse 1971, Freeman 1977)

$$b(v) = \frac{1}{(n-1)(n-2)} \sum_{\substack{u,t \in \mathcal{V}: g_{u,t} > 0 \\ u \neq v, t \neq v, u \neq t}} \frac{g_{u,t}(v)}{g_{u,t}}$$

where $g_{u,t}$ is the number of geodesics from u to t; and $g_{u,t}(v)$ is the number of those among them that pass through node v.

For computation of geodesic matrix see Brandes.

`Network/Create Vector/Centrality/Betweenness`

Geodesic semiring

In the set $A = (\mathbb{R}_0^+ \cup \{\infty\}) \times (\mathbb{N} \cup \{\infty\})$ we define operations:

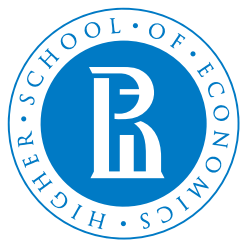
addition:

$$(a,i) \oplus (b,j) = (\min(a,b), \begin{cases} i & a < b \\ i+j & a = b \\ j & a > b \end{cases})$$

and **multiplication**: $(a,i) \odot (b,j) = (a+b, i \cdot j)$.

The obtained structure is a complete and closed semiring.

$$(a,i)^\star = \begin{cases} (0,\infty) & a = 0, i \neq 0 \\ (0,1) & \text{otherwise} \end{cases}$$



Betweenness and geodesic semiring

From matrices of shortest distances $[d_{u,v}]$ and number of geodesics $[g_{u,v}]$ we can get also values $g_{u,v}(t)$ as follows:

$$g_{u,v}(t) = \begin{cases} g_{u,t} \cdot g_{t,v} & d_{u,t} + d_{t,v} = d_{u,v} \\ 0 & \text{otherwise} \end{cases}$$

For a simultaneous computation of both matrices $[(d_{u,v}, g_{u,v})]$ we can use the Fletcher's algorithm over the geodesic semiring applied on a coupled matrix:

$$(d,n)_{u,v} = \begin{cases} (1,1) & (u,v) \in \mathcal{A} \\ (\infty,0) & (u,v) \notin \mathcal{A} \end{cases}$$

Betweenness in R


```
mat.geodesics <- function(m)
{ n <- nrow(m)
  md <- m; md[m==0] <- Inf;  mc <- m; mc[m>0] <- 1
  for(k in 1:n) { for(u in 1:n){ for(v in 1:n){
    dst <- md[u,k] + md[k,v]
    if(md[u,v] >= dst) {
      cnt <- mc[u,k]*mc[k,v];
      if (md[u,v] == dst) {mc[u,v] <- mc[u,v] + cnt }
      else{ md[u,v] <- dst; mc[u,v] <- cnt }
  }}}}
  return(list(dis=md,cnt=mc))
}
vec.Closeness <- function(dis)
{ n <- nrow(dis); return((n-1)/rowSums(dis)) }
vec.Betweenness <- function(dis,cnt){
  n <- nrow(dis); bw <- rep(0,n)
  for (v in 1:n) {
    b <- 0
    for(u in 1:n) { for(w in 1:n) {
      if((cnt[u,w] > 0) && (u != w) && (u != v) && (v != w) &&
            ((dis[u,v] + dis[v,w]) == dis[u,w]))
        {b <- b + cnt[u,v]*cnt[v,w] / cnt[u,w]}
    }}
    bw[v] <- b/((n-1)*(n-2))
  }
  return(bw)
}
vec.betweenness <- function(m)
{mt <- mat.geodesics(m); return(vec.Betweenness(mt$dis,m$tcnt))}
```

Medieval Russia trade network

Pitts network

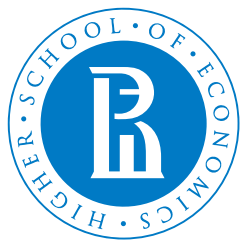
```
> setwd("C:/Users/batagelj/Documents/papers/2018/moskva/NetR/net
> pitts <- read.paj("PittsGeo.net")
> list.vertex.attributes(pitts)
> R <- as.matrix(pitts); nam <- rownames(R)
> x <- pitts %v% "x"; y <- pitts %v% "y"
> Geo <- mat.geodesics(R)
> Gcl <- vec.Closeness(Geo$dis); names(Gcl) <- nam
> Gbw <- vec.Betweenness(Geo$dis,Geo$cnt); names(Gbw) <- nam
> q <- order(Gbw,decreasing=TRUE)
> cbind(bw=Gbw[q],cl=Gcl[q])
                              bw        cl
 Kolomna                0.351141367 0.3584906
 Moskva                 0.341112917 0.3486239
 Ksnyatin               0.296721534 0.3166667
 Kozelsk                0.214065569 0.3220339
 Dorogobuzh             0.189087584 0.3089431
 Tver                   0.159225767 0.2923077
 Vladimir               0.157159791 0.2814815
 Vyazma                 0.141979950 0.3304348
 Bryansk                0.116639572 0.2878788
> plot(Gbw,Gcl,pch=16,col="red",xlim=c(-0.02,0.35),
+ main="Pitts - binary")
> text(Gbw,Gcl,nam,cex=0.5)
```

For the binary network Kolomna has larger betweenness than Moskva.

. . . Medieval Russia trade network

Let's try to compute the betweenness considering geographical distances among places:

```
> mat.dist <- function(R,x,y){
+   n <- nrow(R); D <- R
+   for(i in 1:n) { for(j in 1:n) {
+     if(R[i,j]>0) D[i,j] <- sqrt((x[i]-x[j])^2+(y[i]-y[j])^2)
    }}
+   return(D)
+ }
> Deo <- mat.geodesics(mat.dist(R,x,y))
> Dcl <- vec.Closeness(Deo$dis); names(Dcl) <- nam
> Dbw <- vec.Betweenness(Deo$dis,Deo$cnt); names(Dbw) <- nam
> p <- order(Dbw,decreasing=TRUE)
> cbind(bw=Dbw[p],cl=Dcl[p])
                              bw          cl
 Moskva                0.238975818 0.21250622
 Ksnyatin              0.217638691 0.19895479
 Kolomna               0.207681366 0.20971455
 Dorogobuzh            0.170697013 0.18185725
 Kozelsk               0.159317212 0.19084617
 Mozhaysk              0.145092461 0.20713976
 Dedoslavl             0.133712660 0.19798117
 Tver                  0.132290185 0.19174094
 Vyazma                0.129445235 0.19372181
 Bryansk               0.109530583 0.17079453
> plot(Dbw,Dcl,pch=16,col="red",xlim=c(-0.02,0.24),
+ main="Pitts - distance")
> text(Dbw,Dcl,nam,cex=0.5)
```



Multiplication of Matrix and Vector

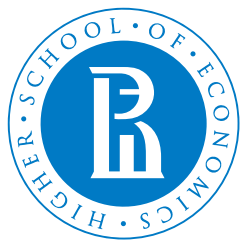
Let e_i be a unit vector of length n – the only nonzero element is at the i-th position and it is equal to 1. It is essentially a $1 \times n$ matrix. The product of a unit vector and a value matrix of a network can be used to calculate the values of walks from a node i to all the other nodes.

Let us denote $q_1^T = e_i^T \odot \mathbf{W}$. The values of elements of the vector q_1 are equal to the values of walks of the length 1 from a node i to all other nodes: $q_1[j] = w(\mathcal{S}_{ij}^1)$. We can calculate iteratively the values of all walks of the length s, $s = 2, 3, \ldots, k$ that start in the node i: $q_s^T = q_{s-1}^T \odot \mathbf{W}$ or $q_s^T = e_i^T \odot \mathbf{W}^s$ and $q_s[j] = w(\mathcal{S}_{ij}^s)$.

Similarly we get ${q^{(k)}}^T = e_i^T \odot \mathbf{W}^{(k)}$, $q^{(k)}[j] = w(\mathcal{S}_{ij}^{(k)})$ and ${q^*}^T = e_i^T \odot \mathbf{W}^*$, $q^*[j] = w(\mathcal{S}_{ij}^*)$.

This can be generalized as follows. Let $\mathcal{I} \subseteq \mathcal{V}$ and $e_\mathcal{I}$ is the characteristic vector of the set $\mathcal{I}$ – it has value 1 for elements of $\mathcal{I}$ and is 0 elsewhere. Then, for example, for $q_k^T = e_\mathcal{I}^T \odot \mathbf{W}^k$, it holds $q_k[j] = w(\bigcup_{i \in \mathcal{I}} \mathcal{S}_{ij}^k)$.

How to in Pajek? Temporal networks, TQ



Permutation matrices

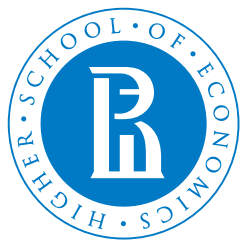
A *permutation* matrix is a matrix in which every row and column contains precisely a single 1 with 0s everywhere else. A permutation matrix of permutation π is

$$p_{ij} = \begin{cases} 1 & \pi(i) = j \\ 0 & \text{otherwise} \end{cases}$$

Matrices **A** and **B** are *permutation-similar*, $\mathbf{A} \equiv \mathbf{B}$, iff there exists a permutation matrix **P** such that $\mathbf{B} = \mathbf{P} \cdot \mathbf{A} \cdot \mathbf{P}^T$.

Permutation-similarity of matrices is an equivalence relation.

Matrices of isomorphic graphs are permutation-similar.



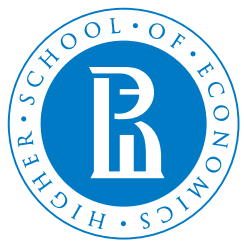
Inverse matrix

A square matrix $\mathbf{A}$ is called *singular* iff $\det \mathbf{A} = 0$.
For a nonsingular matrix $\mathbf{A}$ there always exists a unique matrix $\mathbf{B}$ such that $\mathbf{AB} = \mathbf{1}$. It is called the *inverse* matrix of matrix $\mathbf{A}$ and is denoted by $\mathbf{A}^{-1}$. It holds

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{1}$$

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

In R we can get the inverse matrix of $\mathbf{A}$ using the function `solve(A)`.



Eigenvalues and eigenvectors

Let for a square matrix **A** exist a number λ and a vector $\mathbf{x} \neq \mathbf{0}$ such that

$$\mathbf{Ax} = \lambda\mathbf{x}$$

then λ is called an (right, column) *eigenvalue* and **x** an *eigenvector* of matrix **A**.

An eigenvector is determined up to positive scaling.Usually the factor is selected so that $||\mathbf{x}|| = \sqrt{\mathbf{x}^T\mathbf{x}} = 1$.

All eigenvalues are solutions of the equation $\det(\mathbf{A} - \lambda\mathbf{1}) = 0$. In general case they can be also complex numbers.

Let **T** be a nonsingular square matrix and $\mathbf{B} = \mathbf{T}^{-1}\mathbf{AT}$. Then matrices **A** and **B** have the same eigenvalues.

In R we get eigenvalues and eigenvectors using the function `eigen(A)`, or `eigen(A,only.values=TRUE)` if we need only eigenvales.

. . . Eigenvalues and eigenvectors

Let $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n)$ and $\mathbf{V} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_n]$ be a matrix composed of corresponding eigenvectors. Then it holds

$$\mathbf{AV} = \mathbf{V}\Lambda$$

If all eigenvalues are different the corresponding eigenvectors are independent and therefore the matrix $\mathbf{V}$ is nonsingular. We get

$$\mathbf{V}^{-1}\mathbf{AV} = \Lambda \qquad \text{or} \qquad \mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{-1}$$

Let $\mathbf{T}^{-1}\mathbf{AT} = \Lambda$. Then also $\mathbf{T}^{-1}\mathbf{A}^k\mathbf{T} = \Lambda^k = \mathrm{diag}(\lambda_i^k)$. Therefore matrices $\mathbf{A}$ and $\mathbf{A}^k$ have the same eigenvectors.

The relation $\mathbf{A}^k = \mathbf{T}\Lambda^k\mathbf{T}^{-1}$ can be used for efficient computation of powers of matrix $\mathbf{A}^k$.

. . . Eigenvalues and eigenvectors

Consider now $\mathbf{A}^\star = \sum_{k=0}^{\infty} \mathbf{A}^k$

$$\mathbf{T}^{-1}\mathbf{A}^\star\mathbf{T} = \sum_{k=0}^{\infty} \Lambda^k = \mathrm{diag}(\sum_{k=0}^{\infty} \lambda_i^k)$$

If for all $|\lambda_i| < 1$ we get

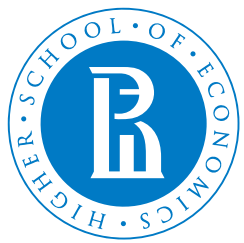
$$\mathbf{T}^{-1}\mathbf{A}^\star\mathbf{T} = \mathrm{diag}(\frac{1}{1-\lambda_i}) \qquad \text{or} \qquad \mathbf{A}^\star = (\mathbf{1}-\mathbf{A})^{-1}$$

Left (or row) *eigeinvectors* satisfy the equation $\mathbf{yA} = \lambda\mathbf{y}$. We collect left eigenvectors in a matrix $\mathbf{U}$. We have

$$\mathbf{UA} = \Lambda\mathbf{U} \qquad \text{or} \qquad \mathbf{UAU}^{-1} = \Lambda$$

Therefore, selecting appropriate multipliers, we get

$$\mathbf{U} = \mathbf{V}^{-1} \qquad \text{or} \qquad \mathbf{UV} = \mathbf{1}$$



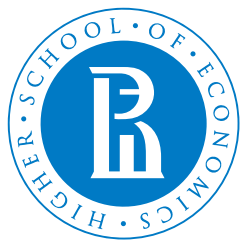
Eigenvalues of symmetric matrices

All eigenvalues of symmetric matrix are real numbers.

The corresponding eigenvectors are *orthogonal* – for every pair of eigenvectors $\mathbf{x} \neq c\mathbf{y}$ it holds $\mathbf{x}^T\mathbf{y} = 0$.

A matrix $\mathbf{T}$ is called *orthogonal* iff $\mathbf{T}^T = \mathbf{T}^{-1}$.

For a symmetric matrix $\mathbf{A}$ we have $\mathbf{V}^{-1}\mathbf{A}\mathbf{V} = \Lambda$ where $\mathbf{V}$ is an orthogonal matrix and Λ a real matrix.

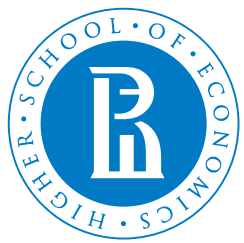


Non-negative matrices

We say that a real matrix $\mathbf{A} = [a_{ij}]$ is

- *positive*, if $\forall i, j : a_{ij} > 0$
- *non-negative*, if $\forall i, j : a_{ij} \geq 0$

A non-negative matrix is *irreducible* if its graph is strongly connected, and is *primitive* if it is also aperiodic.



Perron Frobenius theorem

Let **A** be a non-negative matrix. It has the (*Perron Frobenius*) eigenvalue λ^* and the associated eigenvector $\mathbf{x}^*$ such that:

a. λ^* is a non-negative real number;
b. for any other eigenvalue λ we have $|\lambda| \leq \lambda^*$;
c. the eigenvector $\mathbf{x}^*$ is non-negative

If **A** is irreducible then also

d. if $\mathbf{y} \geq \mathbf{0}$, $\mathbf{y} \neq \mathbf{0}$ is a vector and μ is a number such that $\mathbf{Ay} \leq \mu\mathbf{y}$ then $\mathbf{y} > \mathbf{0}$ and $\mu \geq \lambda^*$ with $\mu = \lambda^*$ iff **y** is a multiple of **x**.
a'. the eigenvalue λ^* is positive with multiplicity 1;
c'. eigenvector $\mathbf{x}^*$ is positive;
e. eigenvector $\mathbf{x}^*$ is the only non-negative eigenvector;
f. let $s = \min_i \sum_j a_{ij}$ and $S = \max_i \sum_j a_{ij}$. Then either $s < \lambda^* < S$ or $\lambda^* = s = S$. Similarly for columns.

Markov chains

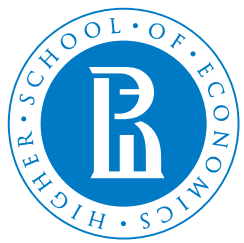
A square matrix **P** is *stochastic* iff it is nonegative and its rows sum to 1. A product of stochastic matrices is again a stochastic matrix. A power of stochastic matrix is a stochastic matrix.

A (discrete time, finite) *Markov chain* (MC) has for its *transition* matrix a stohastic matrix. It can be represented by the corresponding *transition network* (graph). Nodes of MC are ususlly called *states*. A strong component of transition network is said to be *terminal* if no arc is leaving it. The states of MC can be partitioned into:

- *transient* – not in terminal strong component;
- *recurrent* (or persistent) – belongs to a terminal strong component. They can be further partitioned to *aperiodic* ($p = 1$) and *periodic* ($p > 1$).

There is always at least one recurrent state.

Entry p_{ij}^k of the k-th power of transition matrix $\mathbf{P}^k$ is equal to the probability that the process starting in state v_i is after k steps in state v_j. Let $\mathbf{p}(k) = \mathbf{p}(0)\mathbf{P}^k$. Then $\mathbf{p}(k)_i$ is equal to the probability that after k steps the process is in state v_i if the selection of starting state is based on distribution $\mathbf{p}(0)$.

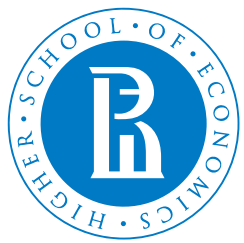


Regular Markov chains

A MC with a single strong component is called *ergodic* (irreducible). An ergodic MC is *regular* iff all its states are aperiodic.

In a regular MC the power matrix $\mathbf{P}^k$ tends with increasing k to the matrix $\mathbf{W}$ that has all its rows equal to the stochastic vector $\mathbf{w}$ satisfying the equation $\mathbf{wP} = \mathbf{w}$. The vector $\mathbf{w}$ is positive. It holds $\mathbf{PW} = \mathbf{WP} = \mathbf{W}$. For an arbitrary initial stochastic vector $\mathbf{p}(0)$ vectors $\mathbf{p}(k) = \mathbf{p}(0)\mathbf{P}^k$ tend to the vector $\mathbf{w}$ – a *stationary* distribution.

To compute the vector $\mathbf{w}$ we rewrite the equation $\mathbf{wP} = \mathbf{w}$ as $\mathbf{P}^T\mathbf{w}^T = \mathbf{w}^T$ or $(\mathbf{P}^T - \mathbf{1})\mathbf{w}^T = \mathbf{0}$. These equations are not independent, but we have an additional relation $\sum w_i = 1$ to complete the system. To get the vector $\mathbf{w}$ we solve the system.



Ergodic MCs

A sequence of matrices $\mathbf{A}_i$ is *Cesaro-summable* to a matrix $\mathbf{A}$ iff

$$\lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} \mathbf{A}_i = \mathbf{A}$$

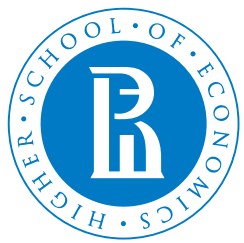
For ergodic MCs hold in Cesaro sense the same properties as for regular MCs.

The *fundamental matrix* of an ergodic MC is

$$\mathbf{Z} = (\mathbf{1} - (\mathbf{P} - \mathbf{W}))^{-1}$$

The matrix $\mathbf{Z}$ always exists and it holds (for periodic)

$$\mathbf{Z} = \mathbf{1} + \sum_{i=1}^{\infty} (\mathbf{P}^i - \mathbf{W})$$



An application of the fundamental matrix

Let $\mathbf{E} = [e_{ij}]$ be a matrix with:

- $e_{ij}, i \neq j$ expected number of steps to reach the first time the state v_j starting in state v_i;
- e_{ii} expected number of steps to return the first time to the starting state v_i;

It is called the *mean first passage time* matrix. It holds

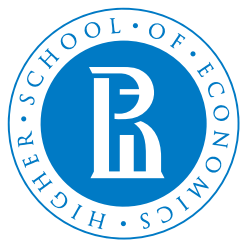
$$\mathbf{E} = (\mathbf{1} - \mathbf{Z} + \mathbf{J} \cdot \mathrm{diag}(\mathbf{Z})) \cdot \mathrm{diag}(\frac{1}{w_i})$$

where $\mathbf{J}$ is a square matrix with all entries 1. For diagonal entries we have $e_{ii} = \frac{1}{w_i}$.

An application of the fundamental matrix in R


```
# summer weather
# F.S. Roberts: Discrete mathematical models, p.267
S <- c( 1/3, 1/2, 1/6,
        1/2, 1/3, 1/6,
        1/3, 1/3, 1/3 )
n <- c('hot','moderate','cool')
S <- matrix(nrow=3,byrow=T,dimnames=list(n,n),data=S)

n <- nrow(S)
A <- t(S)-diag(n); A[n,] <- rep(1,n)
b <- rep(0,n); b[n] <- 1
w <- solve(A,b)
W <- matrix(nrow=n,ncol=n,byrow=T,w,dimnames=dimnames(S))
Z <- solve(diag(n) - (S - W))
E <- (diag(n) - Z + matrix(nrow=n,ncol=n,byrow=T,diag(Z)))
     %*% diag(1/w)
dimnames(E) <- dimnames(S)
```



MCs with transient states

Terminal strong components in a MC are *absorbing* — when it is entered it can not be leaved anymore. If we shrink strong components and reorder the so obtained reduced transition matrix **P** by listing first all absorbing states we get:

$$\mathbf{P} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{R} & \mathbf{Q} \end{bmatrix}$$

For its powers we have:

$$\mathbf{P}^k = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{R}_k & \mathbf{Q}^k \end{bmatrix}$$

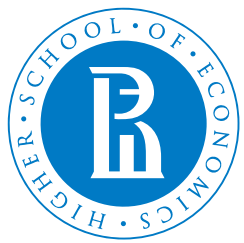
The sequence $\mathbf{Q}^k \to \mathbf{0}$ for $k \to \infty$. It always exists a matrix $\mathbf{N} = \sum_{i=0}^{\infty} \mathbf{Q}^i = (\mathbf{1} - \mathbf{Q})^{-1}$. Its entry n_{ij} is equal to the expected time of staying in state v_j if starting in state v_i.

The entry b_{ij} of the matrix $\mathbf{B} = \mathbf{NR}$ is equal to the probability that the process starting in a transient state v_i will be absorbed by the absorbing state v_j,

MCs with transient states in R


```
# gambler's ruin
# F.S. Roberts: Discrete mathematical models, p.268
p <- 1/3
G <- c( 1 , 0 , 0 , 0, 0,
      1-p, 0 , p , 0, 0,
       0 ,1-p, 0 , p, 0,
       0 , 0 ,1-p, 0, p,
       0 , 0 , 0 , 0, 1 )
n <- c('$0','$1','$2','$3','$4')
G <- matrix(nrow=5,byrow=T,dimnames=list(n,n),data=G)

n <- 5; m <- 2
G <- mat.perm(G,c(1,5,2,3,4))
Q <- G[(m+1):n,(m+1):n]
N <- solve(diag(n-m)-Q)
R <- G[(m+1):n,1:m]
B <- N %*% R
```



Applications

Semirings

Matrices over semirings

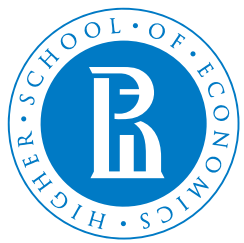
Networks and matrices

Real matrices

Markov chains

Resources

Modeling sequences of events.
Population slides
Google slides
Zweig, page 104 - Generating networks



Resources I

Baras JS, Theodorakopoulos G (2010) Path problems in networks. Morgan & Claypool, Berkeley

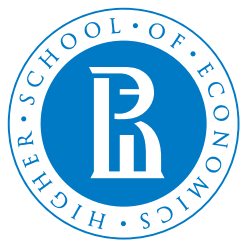
Batagelj, V.: *Semirings for Social Networks Analysis*. Journal of Mathematical Sociology, **19**(1994)1, 53-68.

Batagelj V, Cerinšek M (2013) On bibliographic networks. Scientometrics 96(3):845–864

Batagelj V, Praprotnik S (2016) An algebraic approach to temporal network analysis based on temporal quantities. Social Network Analysis and Mining 6(1): 1-22

Carré B (1979) Graphs and networks. Clarendon, Oxford

Cerinšek M, Batagelj V (2018) Semirings and Matrix Analysis of Networks. in Reda Alhajj and Jon Rokne, Encyclopedia of Social Network Analysis and Mining. Springer.



Resources II

Grinstead CM and Snell JL (2003): Introduction to Probability. (Chapter 11: Merkov chains). AMS. WWW

Fletcher JG (1980) A more general algorithm for computing closed semiring costs between vertices of a directed graph. Commun ACM 23(6): 350–351 MathSciNet

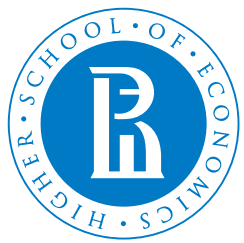
Gondran M, Minoux M (2008) Graphs, dioids and semirings: new models and algorithms. Springer, New York

Kemeny, J.G., Snell, J.L.: *Finite Markov Chains*. Van Nostrand, New Jersey, 1960. Amazon.

Kemeny, J.G., Snell, J.L., Thompson, G.L.: *Introduction to Finite Mathematics*. WWW.

Kepner J, Gilbert J (2011) Graph algorithms in the language of linear algebra. SIAM, Philadelphia

Lancaster, K.: Mathematical Economics. Dover, 2011



Resources III

Roberts, F.S.: *Discrete Mathematical Models*. Prentice-Hall, New Jersey, 1976. Amazon.

Schvaneveldt RW, Dearholt DW, Durso FT (1988) Graph theoretic foundations of Pathfinder networks. Comput Math Appl 15(4):337–345. MathSciNet