**Aplikace matematiky:** Google PageRank

**1. Úvod**

V této práci se budu zabývat jednou z komerčně nejvýnosnějších aplikací matematiky. Popíšu a pokusím se vysvětlit algoritmus PageRank, který byl navržený zakladateli Googlu Larrym Pagem a Sergeyem Brinem, prvně vydaný roku 1998. Samotný název práce od Tanyi Leiseové a Kurta Bryana – The $25,000,000,000 eigenvector – the linear algebra behing Google, napovídá o komerční důležitosti této aplikace matematiky.

PageRank je důležitý z toho hlediska, že efektivně ohodnocuje stránky internetu na základě hledaného klíčového slova a to tak, že je uživateli vypíše sestupně dle důležitosti – právě dle PageRanku. To byl ve své době unikátní počin (a to mluvíme o době teprve o patnáct let nazpět), neboť tehdy takový efektivní algoritmus neexistoval a tudíž právě PageRank stál za následným tak masivním vzestupem Googlu.

Dále bych na úvod rád zmínil nějaké historické zajímavosti ohledně PageRanku. Suma, vyskytující se v názvu výše zmíněného článku od Leiseové a Bryana, odpovídá odhadované hodnotě společnosti Google v roce 2004, kdy se objevila na veřejnosti. Název PageRank působí sám o sobě zmatečně v kontextu s jeho autory a anglickým názvem pro stránku. Ohledně toho se Google vyjádřil, že název pochází z příjmení od Larryho Page. Zajímavostí je, že PageRank byl patentován Stanfordskou univerzitou (kde Page studoval) a následně Google za práva PageRank využívat dal univerzitě 336 milionů dolarů ve svých akciích.

**2. Definice PageRanku**

**2.1. Úmluva**

V následujícím textu budu místo názvu PageRank používat zkratku PR.

**2.2. Základní myšlenka**

Představme si internet jako orientovaný graf. Vrcholy grafu jsou internetové stránky. Mezi dvěma vrcholy **A** a **B** vede hrana s orientací k vrcholu **B** právě tehdy, když stránka **A** ve svém obsahu odkazuje na stránku **B**.

Uživatel, který zadá do internetového vyhledavače nějaký dotaz očekává, že mu vyhledavač vypíše stránky, odpovídající jeho požadavkům a zároveň budou setříděné dle důležitosti. Ta je právě určena důležitostí stránek, které na danou stránku odkazují. Výpočet této důležitosti je výsledkem PR algoritmu.

**Definice 1**. (PageRank)

Mějme síť n internetových stránek reprezentovaných orientovaným grafem **G** s n vrcholy takovým, že z vrcholu **A** vede hrana do vrcholu **B** tehdy, odkazuje-li stránka **A** na stránku **B**. Pak PR stránky **G** (označený jako *r(Pi)* ) vypočteme jako:

V této definici (1) udává inicializační proces algoritmu (tedy to, že na začátku algoritmu mají všechny stránky stejné ohodnocení) a (2) je cyklus opakující se do té doby, dokud PR hodnoty nekonvergují k nějaké výsledné hodnotě (tedy k -> ∞). Rovnice (2) nám udává, že sčítáme PR stránek odkazující na naši ohodnocovanou stránku . Tuto množinu stránek označujeme jako . Takto posčítané hodnoty následně vydělíme počtem stránek odkazujících na stránku , což značíme jako .

Právě k jakým hodnotám jednotlivá ohodnocení stránek konvergují a jak tyto hodnoty vypočítat bude předmětem následujících stránek této práce. Půjdeme od začátku, přes web jako graf, postupně budeme narážet na určité problémy, které budeme překonávat a ve výsledku dostaneme hledaný stochastický PageRank vektor, jehož složky budou požadovaná ohodnocení stránek (a jehož součet složek bude jednička).

**3. Základy teorie Markovových řetězců**

V následující fázi textu si zadefinujeme a dozvíme se něco o Markovových řetězcích. To, že budeme při výkladu o PR používat teorii Markovových řetězcích nám značně zjednoduší pochopení a také výpočet výsledného ohodnocení. Tím si celkovou práci poměrně zjednodušíme, neboť při prvotní konstrukci algoritmu o Markovových řetězcích nepadla údajně ani zmíňka.

**Definice 2. *Maticí sousednosti grafu******G*** *o n vrcholech definujeme jako matici* ***A*** *typu n x n takovou, že.*

**Definice 3.** *Posloupnost celočíselných náhodných veličin se nazývá* ***Markovův řetězec*** *jestliže*

P(Xn+1=j I Xn = in,…, X0 = i0) = P(Xn+1=j I Xn = in) (1)

*pro každé n ≥ 0, a i0, …, in* celočíselná taková, že P(Xn+1=j I Xn = in,…, X0 = i0) > 0.

Pokud pravděpodobnosti přechodu nezávisejí na n, pak Markovův řetězec nazveme **homogenním.** (dále uvažujeme již pouze homogenní Markovovy řetězce).

**Poznámka:** Vztah (1), vyjadřující **markovskou vlastnost** chápeme tak, že výsledek v čase n + 1 je závislý pouze na stavu v přítomnosti – v čase n.

**Absolutní pravděpodobnost** stavu j v čase n značíme jako P (Xn = j) = pj(n).

**Definice 4. Matice přechodu (1. řádu) Markovova řetězce W** definujeme jako wij = P(Xn+1=j I Xn = i) = pij(n, n + 1), jinak řečeno matice přechodu Markovova řetězce je na pozici wij určena pravděpodobností přechodu systému ze stavu i do stavu j. **Matice přechodu n-tého řádu** **Markovova řetězce Wn** definujeme tak, že na pozici wij je podmíněná pravděpodobnost přechodu daného systému při n-tém pokusu ze stavu i do stavu j. Tím, že uvažujeme pouze homogenní Markovovy řetězce, tak pro nás splývají obě dvě předchozí části definice – matice přechodu nezávisí na n.

**Definice 5.** Transponovaný vektor **πT** nazveme **stacionární vektor**, jestliže platí: **πT = πT P** , kde **P** je matice přechodu Markovova řetězce.

**Definice 6.** Matice se nazývá **sloupcově (řádkově) stochastická**, je-li součet prvků v každém jejím sloupci (řádku) roven 1.

**Tvrzení 7.** (Vlastnosti matice přechodu Markovova řetězce)

Mějme Markovův řetězec s k stavy a **W** je jeho matice přechodu. Poté platí:

1. **W** je čtvercová k x k
2. wij
3. **W** je řádkově stochastická

Důkaz: Triviální důsledky definice pravděpodobnosti.

**Tvrzení 8.** *(Vztah matice přechodu a matice sousednosti)*

Nechť **G** je orientovaný graf o n vrcholech, **A** je matice sousednosti grafu **G**. Pak matici přechodu Markovova řetězce je čtvercová matice **W** = (wij) řádu n takové, že

Důkaz:

Tvrzení vychází z předpokladu, že z v libovolném čase je pravděpodobnost, že se dostaneme z libovolného uzlu A do uzlu B stejná, jako z uzlu A do uzlu B, existují li příslušné hrany. To ale odpovídá matici přechodu homogenního Markovova řetězce.

□

Toto tvrzení nám dává jednu z možností, jak zkonstruovat požadovanou Markovovu matici, která, jak se na následujících stránkách dozvíme, je pro PR klíčová.

**Definice 9.**  **Odkazovou matici H** n stránek definujeme jako čtvercovou matici řádu n takovou, že

hij = ,pokud zde existuje hrana ze stránky Pi do stránky Pj a hij = 0 v jiném případě, přičemž I*Pi*I udává, na kolik stránek celkem stránka *Pi* odkazuje.

Následuje jednoduchý případ odkazové matice **H**:

1

4

3

2

5

(obrázek 1)

**Definice 10. PR vektor** i-té iterace grafu **G** s n vrcholy definujeme jako n-složkový řádkový vektor [**πT**]**i**, udávající PR postupně všech n vrcholů grafu **G**. Výsledný PR vektor, ohodnocující stránky PR algoritmem je definovaný pro i 🡪 , tedy vektor, ke kterému konvergují PR hodnoty stránek webu.

**Tvrzení 11.** *(PR vektor a odkazová matice)*

Nechť **H** je odkazová matice grafu **G** a **πT** je řádkový PR vektor grafu **G**. Poté platí: [**πT**]k+1 = [**πT**]k **H**. Tvrzení 11. lze induktivně přepsat jako [**πT**]k+1 = [**πT**]0 **H**k+1, což je již známý problém mocnění matice.

**Definice 12.** Matici **P** nazveme **permutační maticí**, jestliže vznikla prohozením řádků (či sloupců) z jednotkové matice.

**Definice 13.** Čtvercová matice **A** řádu n se nazývá **reducibilní**, jestliže existuje permutační matice P řádu n taková, že  , kde **X a Z** jsou čtvercové bloky.

Příkladem reducibilní matice je například matice **A** = , neboť existuje permutační matice **P** = pro kterou platí: =, což odpovídá struktuře v definici 13.

Ekvivalentě na základě definice 13 definujeme ireducibilní matice, které jsou pro zavedení PR důležitější. Ireducibilními maticemi jsou právě ty matice, které nejsou reducibilní, tedy pro každou permutační matici **P** neplatí rovnost v definici 13. Získat příklady ireducibilních matic je obtížnější, protože by to skýtalo vynásobit danou matici všemi permutačními maticemi stejného řádu. To, že je ireducibilní (popřípadě reducibilní) je možné snadněji nahlédnout při sestrojení orientovaných grafů na základě matic (jako se například uvádí v knize [7], na stránce 171, kde lze také nalézt nějaké příklady ireducibilních matic), což ale není pro celkové pochopení algoritmu tak důležité, tudíž teorii vyšetření (i)reducibility dané matice se již dále věnovat v tomto textu nebudeme.

**Definice 14.**

Matici přechodu **W** Markovova řetězce nazveme **aperiodickou**, jestliže neexistuje přirozené číslo *k*

(k ≥ 1) takově, že **W** = **W*k***.

**Věta 15.** *(Konvergence Markovova řetězce)*

Nechť **W** je matice přechodu homogenního Markovova řetězce, která je aperiodická, ireducibilní a stochastická. Dále předpokládejme že **πT** je stacionární vektor uvažovaného Markovova řetězce. Potom platí:

Důkaz:Z důvodu, že samotný důkaz této věty a zavedení potřebných pojmů a lemmat je několikastránkový text, tak pouze uvedu, kde je ho možné nalézt. Velmi kvalitně zpracovaný je v [5], na straně 126.

Nyní se znalostí věty 15. chceme tedy nalézt nějakou matici, která bude reprezentovat web a zároveň bude matice přechodu Markovova řetězce splňující předpoklady věty 15. To samé při konstrukci této matice, kterou dále budu označovat jako **Google matici G**, použili Brin a Page, aniž by jen uvažovali teorii Markovových řetězců. Nakonec spočteme její stacionární vektor a k němu podle věty 15 budou konvergovat PR hodnoty stránek námi uvažovaného webu.

První přídavek, takzvaný stochastický, řeší mimo to, že **G** bude stochastická i problém hromadících uzlů. Hromadící uzly jsou ty uzly, ze kterých neodchází žádný odkaz na jinou stránku a naopak nějaká ze stránek na ni odkazuje. Takovým příkladem může být uzel č. 3 na obrázku 1. Použijeme představu náhodně procházejícího uživatele, který prochází web libovolně. To znamená, že přijde-li uživatel na nějakou stránku s několika odkazy, pak si vybere nějakou z nich, na jakou se vydá. Tím je dána pravděpodobnost, na jakou ze stránek se uživatel dostane právě z hromadící stránky – je pro všechny stránky webu stejná (v tom je tedy započítána I pravděpodobnost, že na dané stránce zůstaneme. Touto úpravou, kdy změníme naší odkazovou **H** změníme na matici **S.**

Formálně matici **S** definujeme jako:

, kde n značí počet stránek, **1** je n-složkový jednotkový vektor (tedy všechny jeho složky jsou jedničky) a vektor **a** je vektor hromadícíh uzlů, který je definován ai = 1, je-li i-tá stránka hromadící uzel a ai = 0 jinak.

V našem případě, kdy uvažujeme graf znázorněný na obrázku 1, dostaneme touto stochastickou úpravou matici **S** jako:

, neboť vektor **a** je .

Následující úpravu nově získané matice **S** nazýváme primitivní úpravou, neboť dostaneme novou, již finální Google matici **G**, která je primitivní. Předtím si zavedeme pojem primitivní matice (v definici 17). Tím dostaneme již výslednou matici **G** (s využitím tvrzení 18), protože splníme předpoklady věty 15. Ta nám tedy říká, že pro jakýkoli počáteční vektor **π0** nám markovův proces, jehož matice přechodu je námi vytvořená Google matice **G**, konverguje k jednoznačně určenému stacionárnímu vektoru **π**.

**Definice 16.**

Matici **A** typu n x m nazveme kladnou, jestliže platí aij 0. To, že je matice kladná zkratkovitě zapisujeme jako . Obdobně definujeme nezáporné a záporné matice.

**Definice 17**.

Matici **A** nazveme primitivní, jestliže existuje takové, že **A**k 0.

**Tvrzení 18.**

Matice **A** je primitivní právě tehdy, když je aperiodická a ireducibilní.

Důkaz:

Obdobně jako Věta 15, tak i toto tvrzení by vyžadovalo zavedení několika dalších pojmů, které bychom již dále nevyužili, tudíž uvedu jen odkaz na důkaz [8] – sekce 1.3., kde je jej možné podrobně nalézt.

Tedy díky tvrzení 18 nám stačí k tomu, abychom splnili předpoklady V. 15 o konvergenci Markovova řetězce zkonstruovat takovou matici **G**, která je aperiodická a ireducibilní. Díky tvrzení 18 toho dosáhneme tím, že matice **G** bude primitivní.

Nechť matice **G** je definována následovně:

(2)

, kde je **ohodnocovací parametr**, n je počet stránek webu, **1** je n složkový jednotkový vektor – tudíž **11**T je čtvercová matice řádu n, která má všechny prvky jedničky. Jaké vlastnosti má takto vzniklá matice **G**? V první řadě je opět stochastická, neboť je konvexní kombinací dvou stochastických matic (S je stochastická díky předcházející stochastické úpravě) a čtvercová jednotková matice **11T** řádu n je stochastická díky tomu, že ji násobíme skalárem 1/n, tudíž součet prvků v každém řádku (i sloupci) bude jedna. To, že konvexní kombinace dvou stochastických matic bude opět stochastická matice je zřejmé.

Jak je to s druhým požadavkem ohledně toho, aby **G** byla primitivní? Použijeme definici primitivní matice. Matice **G** primitivní je, neboť již pro k = 1 dostáváme, že **G**k 0. To je díky ohodnocovacímu faktoru , respektive díky tomu, že . Matice **S**je nezáporná, tedy některé její prvky můžou být nulové a zbylé kladné. Tím, že je prvky přenásobíme kladným parametrem jejich nezápornost zřejmě nezměníme. V druhé části k této matici přičítáme matici, jejíž všechny prvky jsou 1/n a přenásobené opět kladným skalárem 1 – α. Tedy takováto výsledná matice je díky tomu, že n je zřejmě kladné (udává počet stránek webu), také kladná. Sečtením nezáporné a kladné matice dostaneme kladnou matici, protože tam, kde byly u matice α**S** nuly, tak tam nyní bude kladné číslo a tam, kde již kladné číslo bylo, tak přičtením dalšího kladného čísla zřejmě bude výsledek opět kladný. Tedy z definice **G** rovnicí (2) dostáváme, že **G**1 0 a proto je **G** primitivní.

Nyní již máme matici, která vyhovuje větě 15 a konečně tedy můžeme spočítat požadovaný PR vektor. Ještě předtím ale určeme matici **G** webu znázorněného na obrázku 1 s předpokladem, že = 0.85, tak, jak to používají Brin a Page. K tomuto ohodnocovacímu faktoru se ještě vrátím později. Naše matice **G** tedy vypadá následovně:

**G** =0.85+ 0.15 =

(3)

Což je výsledná Google matice, take matice přechodu Markovova řetězce, která splňuje všechny předpoklady věty 15.

**Ohodnocovací faktor α**

Jak jsem již zmínil, α je skalár v rozmezí od 0 do 1. Tak, jak ho zvolili Brin a Page, hodnotou α = 0,85, udává, že 85% uživatelů pokračuje v grafové struktuře webu – tedy uživatel pokračuje v „klikání“ na další stránky, na které nynější odkazuje. Zbylých 1 – α (v našem případě tedy 0.15, což odpovídá 15%) naopak strukturu nedodržuje a přenese se na nějakou jinou stránku (třeba právě díky výpisu stránek z vyhledavače Google). To, že je proces přenesení náhodný s pravděpodobností přechodu na jakoukoli stránku , nám zaručí matice tím, že všechny její prvky mají stejnou hodnotu.

Právě na ohodnocovacím faktoru závisí rychlost konvergence k nějaké hodnotě PageRanku. Na základě údajů uvedených v knize [7], tak čím je hodnota ohodnocovacího faktoru blíže k jedničce, tím roste (exponenciálně) počet nutných iterací v definici PR.

Hodnoty požadovaných iterací jsou následující:

|  |  |
| --- | --- |
| α | Počet iterací |
| 0.5 | 34 |
| 0.75 | 81 |
| **0.85** | **142** |
| 0.95 | 449 |
| 0.99 | 2 292 |
| 0.999 | 23 015 |

Tyto hodnoty nejsou nikterak překvapivé. Z toho, jak je parametr α definován, tak dostáváme, že čím více se blíží k jedničce, tak tím spíše bude uživatel pokračovat v grafové struktuře webu (v „klikání“ na další stránky) a bude nutno poměrně více iterací, než se uživatel na nějakou stránku vrátí, aby jí předal PR té, ze které se na ni dostal.

**4. Výpočet PR vektoru**

K úspěšnému dokončení PR algoritmu, bychom potřebovali vypočítat samotný PR vektor. Ještě předtím, než se ale k této závěrečné fázi dostaneme, tak si ale zavedeme jednu definici a dokážeme si jedno tvrzení, které následně použijeme. Tento způsob výpočtu je aplikovatelný pouze pro malý web a je ideální pro „ruční“ počítání, které si zde chceme ukázat. Pro samotnou aplikaci tohoto algoritmu na webu (tak, jak byl zaveden v definici 1) je použita jiná implementace, zmíněná na konci této práce.

**Definice 19.**

Řádkový vektor **yT** odpovídající vlastnímu číslu λ matice **A** nazveme **levý vlastní vektor**, jestliže vyhovuje rovnici **yTA** = λ**yT.**

**Tvrzení 20.** (*Vlastní čísla stochastické matice*)

Nechť **A** je řádkově stochastická matice (s nezápornými prvky). Potom λ = 1 je největší vlastní číslo matice **A.**

**Definice 21.**

Nechť **V**je vektorový prostor nad tělesem **T**, {**x**1, **x**2, …, **x**n} je množina n vektorů z **V** a (a1, a2, … , an) je n-tice čísel z **T**. Pak lineární kombinaci nazýváme **konvexní kombinace**, právě když je součet jejich koeficientů roven jedné a přitom jsou všechny koeficienty nezáporné.

Důkaz tvrzení 20:

Prvně dokážu, že 1 je vlastní číslo stochastické matice **A.**

Uvažujme jednotkový vektor **1**. Pokud dokážeme, že **1** je vlastní vektor **A** příslušný vlastnímu číslu 1, pak z definice vlastního čísla bude 1 vlastní číslo řádkově stochastické matice **A**. Tedy chceme dokázat, že platí: **A1** = 1**1** (kde 1 je skalár a **1** jednotkový vektor). Pravá strana rovnosti je rovna jednotkovému vektoru, podívejme se tedy na levou stranu. Na levé straně z definice násobení matice a vektoru vždy násobíme každý řádek matice s jednotkovým vektorem, což je definováno jako standartní skalární součin řádků matice **A** s jednotkovým vektorem, což je ve výsledku 1, neboť **A** je řádkově stochastická. Z toho plyne, že součin **A1** je jednotkový vektor a je roven pravé straně dokazované rovnice. Tedy λ = 1 je vlastní číslo libovolné řádkově stochastické matice.

Nyní dokážeme, že neexistuje vlastní číslo námi uvažované matice **A** větší jak 1. Pro spor předpokládejme, že existuje λ > 1. Dosazením do definice 19 dostáváme na levé straně rovnice vektor

**xT****=** **yTA** =, tedy vektor, jehož složky jsou konvexní kombinací složek vektoru **yT** (díky tomu, že **A** je řádkově stochastická a všechny její prvky jsou nezáporné). Chtějme nyní dokázat, že platí následující nerovnost:

a1y1 + a2y2 + … + an-1 yn-1 +anyn < yn, kde yn = ymax.

Dostáváme:

a1y1 + a2y2 + … + an-1 yn-1 < (1-an)yn

a1y1 + a2y2 + … + an-1 yn-1 < (a1 + a2 +…+ an-1)yn (neboť )

a závěrečnou úpravou:

a1(y1 – yn) + a2(y2 – yn) +… + an-1(yn-1 – yn) < 0, což platí, protože pro každé i = 1… (n -1) platí, že yn > yi.

Opačnou nerovnost (a tedy spor) získáme z druhé strany nerovnosti, neboť alespoň jeden prvek vektoru λ**yT** je větší jak ymax, neboť λ > 1. Což je hledaný spor.

□

Nyní se dostaneme k již k závěrečné části práce – k výpočtu samotného PR vektoru webu, jenž jsme grafově definovali obrázkem 1. PR vektor určíme z následujících rovnic:

πT= πT**G** (4)

πT **1** = 1 (5)

Rovnicí (4) hledáme levý vlastní vektor matice **G** příslušný vlastnímu číslu 1, které díky tvrzení 20 existuje. Zároveň ale tuto rovnici chápeme tak, že vektor je stacionární vektor, matice **G** takéz hlediska teorie Markovových řetězců. To je zaručeno tím, že přenásobením tohoto vektoru opět maticí **G** musíme dostat hledaný stacionární vektor **π**, neboť celý proces k tomuto vektoru konverguje (díky větě 15). Rovnice (5) již pouze výsledný vektor normalizuje, abychom dostali vektor pravděpodobností (tedy že součet všech jeho složek dá 1).

Označme **πT** = (ψ1, ψ2, ψ3, ψ4, ψ5)T.

Potom spolu s výsledkem (3) dostáváme soustavu pěti lineárních rovnic o pěti neznámých:

ψ1 = 3/100 ψ1 + 47/150 ψ2+1/5 ψ3+ 3/100 ψ4 + 3/100 ψ5

ψ2 = 3/100 ψ1 + 3/100 ψ2+1/5 ψ3+ 3/100 ψ4 + 91/200 ψ5

ψ3 = 91/200 ψ1 + 47/150 ψ2+ 1/5 ψ3+ 91/200 ψ4 + 3/100 ψ5

ψ4 = 91/200 ψ1 + 3/100 ψ2+ 1/5 ψ3+ 3/100 ψ4 + 91/200 ψ5

1 = ψ1 + ψ2 + ψ3 + ψ4 + ψ5

, což maticově po jednoduchých úpravách zapíšeme jako:

A tuto matici nyní Gaussovou eliminační metodou vyřešíme. Výsledkem je s přesností na tři desetinná čísla náš hledaný PageRank vektor: **πT = (0,124 0,168 0,270 0,221 0,217)**T

Když se podíváme na náš web, tak to, že stránka č. 3 má nejvyšší ohodnocení, není překvapující.Je to primárně dáno tím, že na ni nejvíce stránek odkazuje.

Takový postup výpočtu je pro čtvercovou matici, která je řádu vice než miliarda (v roce 2014 byl překročen milník miliardy webových stránek), nemyslitelný. Proto se nejčastěji využívá tzv. mocninná metoda, která přímo vychází z definice 9 PageRanku, kdy hledaný stacionární vektor určíme podle následující podmínky:

, kde je obvykle stanoveno na 10-10 pro libovolný počáteční vektor π0 a k jdoucí od 1 → ∞ (dle nutnosti, v závislosti na zvoleném ε). Pomocí mocninné metody určíme dominantní vlastní vektor (který bude náš hledaný PR vektor) příslušný největšímu vlastnímu číslu naší matice **G**.) V tvrzení 20 jsme dokázali, že toto dominantní vlastní číslo je jednička.

Algoritmus mocninné metody je následující:

Vybereme libovolný počáteční vektor **q**

Dokud pak:

*zk =* ***G****qk-1*

*qk =* (normalizace)

*λk = [T****G***

konec.

Právě tento algoritmus (a nějaké jeho vylepšení) jsou pro implementaci PR v praxi využívané, mou zmíněná metoda výpočtu je praktikovatelná pro ruční počítání s webem o malém rozsahu stránek.

**Zdroje:**

<http://www.karlin.mff.cuni.cz/~dvorak/teaching/2014_2015/NMFM310/NMFM310_2015_kap4.pdf> [1]

<http://is.muni.cz/th/150529/prif_m/diplomka.pdf> [2]

<http://homel.vsb.cz/~bri10/Teaching/Statistika%20II/skriptum/8_Nahodne%20procesy_I.PDF> [3]

http://math.stackexchange.com/questions/40320/proof-that-the-largest-eigenvalue-of-a-stochastic-matrix-is-1 [4]

[http://www.math.uoc.gr/~nikosf/Probability2013/3.pdf [5](http://www.math.uoc.gr/~nikosf/Probability2013/3.pdf%20%5b5)]

[http://pef.czu.cz/~domeova/EMMISYI/prednasky/p6\_soubory/frame.htm [6](http://pef.czu.cz/~domeova/EMMISYI/prednasky/p6_soubory/frame.htm%20%5b6)]

Amy N. Langville, Carl D. Meyer – Google´s PageRank and Beyond: The Science of Search Engine Rankinfs [7]

Gregory F. Lawler-  Introduction to Stochastic Processes [8]

[http://www.cs.huji.ac.il/~csip/tirgul2.pdf [9](http://www.cs.huji.ac.il/~csip/tirgul2.pdf%20%5b9)]

https://cs.wikipedia.org/wiki/Line%C3%A1rn%C3%AD\_kombinace [10]