STUDIA INFORMATICA 2011
Volume 32 Number 2A (96)

Zygmunt MAZUR, Konrad WIKLAK
Politechnika Wrocławska, Instytut Informatyki

METODA POZYCJONOWANIA WYNIKÓW W SYSTEMACH WYSZUKIWANIA INFORMACJI MUZYCZNYCH

Streszczenie. W artykule przedstawiono modyfikację algorytmu PageRank pozycjonowania wyników zwracanych przez wyszukiwarki tekstowe. Dokonano oceny jego przydatności pod kątem wykorzystania w systemie wyszukiwania informacji muzycznych. Przedstawiony w artykule autorski algorytm MusicPageRank oparty jest na istniejących połączeniach pomiędzy stronami WWW oraz plikami muzycznymi. Dzięki wykorzystaniu tych połączeń opracowano metodę tworzenia rankingu dla wyników zwróconych przez wyszukiwarkę plików muzycznych, niezależną od zapytania sformułowanego przez użytkownika.

Słowa kluczowe: informacje muzyczne, pozycjonowanie wyników, wyszukiwanie informacji muzycznych, MusicPageRank

A METHOD OF POSITIONING SEARCH RESULTS IN MUSIC INFORMATION RETRIEVAL SYSTEMS

Summary. The article describes modification of the PageRank text search engine results ranking algorithm. An assessment of its suitability in use in music information retrieval system has been made. Authors propose new MusicPageRank algorithm that is described in article and is based on existing connections (links) between Websites and music files. With using these connections, a new method of creating rank for music search engine results, independent from the user query, have been developed.

Keywords: music information, query results ranking, music information retrieval, MusicPageRank

1. Wstęp

Wyszukiwanie informacji muzycznych, pomimo że jest dziś bardzo dobrze rozwiniętą dziedziną, nadal jest w kręgu zainteresowań wielu osób. Dominujące do niedawna tekstowe

wyszukiwarki plików MP3 są zastępowane przez systemy umożliwiające formułowanie kwerend przez użytkownika na bazie tzw. zapytań przez zanucenie (ang. *query by humming*), podanie zapisu sekwencji dźwięków lub konturu melodycznego szukanej piosenki.

Pliki dźwiękowe są dość specyficznym źródłem danych – zarówno zapisane w nich częstotliwości sygnału dźwiękowego, jak i dodatkowe informacje tekstowe w postaci metadanych umożliwiają wykonywanie różnych typów zapytań. W niniejszym artykule są wykorzystywane metadane w postaci informacji tekstowych, pobieranych z pliku. Dzięki użyciu jednej z wielu dostępnych baz danych utworów muzycznych – jak np. GraceNote czy MusicBrainz – możliwa jest weryfikacja poprawności metadanych. Konstrukcja własnego systemu identyfikacji informacji tekstowych, przechowywanych w plikach jest zadaniem stosunkowo trudnym i nie jest przedmiotem tego artykułu. W prosty sposób natomiast można sprawdzić, czy podstawowe informacje o utworze takie, jak: tytuł, wykonawca, album, rok wydania, gatunek czy prawa autorskie są kompletne. Poprawność i kompletność informacji przechowywanych w pliku w postaci metadanych powinny być głównym kryterium sortowania wyników dla zapytania użytkownika, zwracanych przez wyszukiwarkę systemu wyszukiwania informacji muzycznych. Takie podejście do tematu pozycjonowania wyników ma duże zalety – daje pewność, że pliki muzyczne zwracane przez wyszukiwarkę będą zawierały utwór muzyczny, odpowiadający metadanym, które te pliki zawierają. Istotne znaczenie ma też szybkość obliczania miary poprawności i kompletności metadanych pliku dźwiękowego. Nietrudno dostrzec również wady takiego podejścia. Plik muzyczny jest traktowany jako pojedynczy byt, niemający żadnych powiązań z innymi plikami muzycznymi, dokumentami czy treściami znajdującymi się w Internecie. Niestety takie podejście nie oddaje rzeczywistego stanu rzeczy. Popularność plików dźwiękowych zależy bowiem nie tylko od indywidualnego gustu użytkownika wyszukiwarki, ale również od popularności portali internetowych, które udostępniają te pliki. Zatem lepszym rozwiązaniem będzie wykorzystanie kryteriów popularności stron przechowujących odnośniki do plików muzycznych przez klasyczne algorytmy, wykorzystywane przez dzisiejsze wyszukiwarki tekstowe (np. Google). W artykule dokonano krótkiego przeglądu wybranych algorytmów pozycjonowania wyników zwracanych przez wyszukiwarki tekstowe. Dokonano oceny ich przydatności, ze względu na możliwość zastosowania w systemach wyszukiwania informacji muzycznych. Zaproponowano również modyfikację algorytmu PageRank, ze względu na specyfikę danych, jakimi są pliki dźwiękowe.

2. Opis zagadnienia pozycjonowania wyników wyszukiwarek muzycznych

Niezależnie od sposobu sformułowania kwerendy przez użytkownika[1] system wyszukiwania informacji muzycznych zwraca wynikowy zestaw plików muzycznych, odpowiadających zapytaniu. Do posortowania plików muzycznych, zwracanych przez wyszukiwarkę, możliwe jest zastosowanie wielu rodzajów miar: dynamicznych – zależnych od zapytania – oraz statycznych – bazujących na innych informacjach niż podawane przez użytkownika. Do pierwszej grupy można zaliczyć miary obliczane na podstawie metadanych pliku dźwiękowego. Pliki muzyczne są sortowane w zależności od relewantności metadanych przechowywanych w pliku bądź sekwencji dźwięku lub w zależności od tzw. odcisku palca (ang. *fingerprint*) do danych wejściowych, podanych przez użytkownika. Relewantność może być wyrażona za pomocą ustalonych deskryptorów, wag i miar podobieństwa, w zależności od wybranej metody formułowania kwerendy. Do grupy statycznych właściwości można zaliczyć miary charakteryzujące plik dźwiękowy, na przykład: poprawność informacji wprowadzonych do pliku w postaci metadanych (w pracy oznaczana jako *MetaRank*), format pliku, rodzaj kompresji, rozmiar pliku itp. Są to jednak informacje charakteryzujące wyłącznie plik muzyczny.

Bazując wyłącznie na miarach dotyczących pliku tracimy informacje o kontekście – ten sam plik może być umieszczony na stronie WWW odwiedzanej codziennie przez tysiące internautów oraz na stronie, którą odwiedza zaledwie kilka osób rocznie. Pomijając informacje dotyczące lokalizacji pliku, możliwy jest przypadek, gdy plik z mało uczęszczanej strony znajdzie się w wynikach wyszukiwania wyżej niż ten, ze strony często odwiedzanej. Dodatkowo, nie wszystkie formaty plików dźwiękowych umożliwiają dodawanie metadanych, co praktycznie wyklucza ich użycie jako miary poprawności informacji przechowywanych w pliku. Dlatego konieczne jest dodanie rankingowania wszystkich stron WWW, zawierających odnośniki do danego pliku muzycznego. Ten rodzaj rankingowania jest przedmiotem artykułu. Dla każdej strony WWW, wskazującej na plik muzyczny, zostają przypisane wagi. Miarą kontekstu dla pliku, oznaczaną jako *ContextRank*, jest maksymalna waga spośród stron WWW, zawierających do niego odnośniki, czyli max(*ContextRank*). Wagi stron WWW są generowane z użyciem klasycznych algorytmów, bazujących na tzw. linkach pomiędzy stronami.

Niezależną od zapytania miarą dla pliku muzycznego (*UserRank*) może być również ranking popularności plików przez użytkowników, na zasadzie indywidualnej oceny pliku przez użytkownika w wyszukiwarce. Miara *UserRank* powinna być odpowiednio skonfigurowana ze względu na stosunkowo łatwe, możliwe, celowe działania, mające na celu podwyższenie rankingu plików w wyszukiwarce.

[1] Może być to na przykład wpisanie ciągu znaków reprezentującego metadane utworu, tekst piosenki, sekwencję wysokości dźwięków zagranych przez wybrany instrument lub zaśpiewanie, zagwizdanie czy zanucenie fragmentu utworu.

Miarę *FileRank* niezależną od zapytania można zapisać jako:

$$FileRank = MetaRank + \max(ContextRank) + UserRank$$

Niniejszy artykuł opisuje wstępne prace poprzedzające praktyczną realizację autorskiej koncepcji algorytmu hybrydowego pozycjonowania wyników zwracanych przez wyszukiwarki muzyczne, zaproponowanego w pracach [7] i [10].

Klasyczne algorytmy pozycjonowania wyników, dające dodatkowe informacje na temat kontekstu wyszukiwania, dotyczą przede wszystkim dokumentów tekstowych. Wyniki zwracane przez wyszukiwarkę miały odzwierciedlać relewantność dokumentów do zadanego przez użytkownika zestawu deskryptorów i wag [6]. W zależności od tego jak reprezentowany był dokument w kwerendzie możliwe były różne rozwiązania, jeśli chodzi o miarę podobieństwa dokumentu, do zapytania użytkownika. Dokument mógł być reprezentowany jako element przestrzeni wektorowej. Innym modelem wykorzystywanym w zagadnieniach wyszukiwania i indeksowania dokumentów jest model probabilistyczny. W każdym z modeli użytkownik mógł dodatkowo indywidualnie ocenić przydatność wyszukanych dokumentów tak, aby przyszłe wyniki, zwracane jako odpowiedź na kwerendę, lepiej odpowiadały oczekiwaniom użytkownika. Niestety algorytmy wykorzystujące np. elementy modeli wektorowych i probabilistycznych, takie jak LSA, silnie zależą od zadanego przez użytkownika zapytania – dla każdego zwróconego dokumentu sprawdzana jest relewantność do zapytania użytkownika. Ponieważ jednak algorytmy te nie są przystosowane do pracy z dużymi kolekcjami dokumentów, a rodzajem takiej kolekcji jest sieć Web, należy rozważyć wykorzystanie innego rodzaju algorytmów – bazującego na połączeniach pomiędzy dokumentami.

Kolejna grupa algorytmów dotyczy oceny ważności dokumentu wykorzystując fakt, jakie dokumenty są w nim cytowane oraz w jakich dokumentach dany dokument jest cytowany. Najbardziej znane z tej grupy są HITS (*Hypertext Induced Topic Selection*), PageRank oraz SALSA [2, 5]. Wszystkie trzy algorytmy znalazły swoje zastosowanie w wyszukiwarkach internetowych, a PageRank [3, 4, 9] do dzisiaj stanowi podstawę działania systemu pozycjonowania wyszukiwarki Google. W sieciowej wersji omawianych algorytmów, dokumenty zostały zastąpione przez strony WWW, a cytowania przez odnośniki do innych stron, czyli przez tzw. linki. Wszystkie trzy algorytmy wykorzystują tzw. macierz sąsiedztwa (ang. *adjacency matrix*) stron WWW, zbudowaną na bazie grafu skierowanego, ilustrującego połączenia pomiędzy stronami. Opisy algorytmów są dostępne w literaturze (np. w [2]), dlatego pominięto tutaj omawianie ich podstawowych wersji. Warto jedynie wspomnieć, że macierz sąsiedztwa dokumentów L oraz macierze LL^T i L^TL w algorytmie HITS nie są macierzami stochastycznymi i mogą być redukowalne [8], przez co algorytm HITS może dawać różne wektory wynikowe dla tej samej wartości własnej, w zależności od przyjętego wektora startowego dla metody potęgowej. W literaturze wszystkie zaproponowane metody tworzenia macierzy nieredukowalnej dla algorytmu HITS są bardzo niedokładne, dlatego też metoda ta została pominięta w badaniach.

3. Modyfikacje algorytmów pozycjonowania na potrzeby wyszukiwania informacji muzycznych

W okresie formułowania algorytmów HITS i PageRank użytkownicy wykorzystywali przede wszystkim odnośniki między stronami. Ówczesne wyszukiwarki, np. Yahoo czy AltaVista, stanowiły jedynie podstawową pomoc przy próbie znalezienia strony na interesujący użytkownika temat. Klasyczne wersje algorytmów, oparte na macierzy sąsiedztwa stron, doskonale oddają ten model. Obecnie, po kilkunastu latach korzystania z wyszukiwarek, zachowanie użytkowników uległo zmianie. Dzisiejszy użytkownik nadal korzysta z wyszukiwarki, ale jeśli po odwiedzeniu danej strony stwierdza, że nie ma na niej interesującej treści i nie ma odnośników do stron o interesującej go tematyce, przegląda kolejny dokument z listy wyników zwróconych przez wyszukiwarkę.

Można postawić pytanie: jak dzisiejsze zachowanie użytkowników można uwzględnić w modyfikacji klasycznych algorytmów pozycjonowania dokumentów, bazujących na odnośnikach do stron? Intuicyjnie można założyć, że wynikową wartością dla pliku muzycznego będzie największa wartość, jaką zwróci algorytm pozycjonowania, spośród wszystkich stron WWW, które zawierają odnośnik do tego pliku. Ponieważ dzisiaj wiele stron zawiera pliki audio, dlatego też integralnym elementem strony będzie wprowadzone dodatkowe kryterium, czy strona rzeczywiście tematycznie dotyczy muzyki. Jednym z kryteriów może być przyjęcie dolnego progu tolerancji (TL) dla liczby różnych plików muzycznych, do których strona się odwołuje (zwykle będzie to od 1 do 5 plików). Jeżeli liczba plików będzie poniżej progu tolerancji, strona taka nie będzie brana pod uwagę w algorytmie pozycjonowania, przez co pozycja pliku muzycznego, do którego strona się odwołuje będzie maleć. Eliminacja stron niedotyczących tematycznie muzyki pozwoli na zmniejszenie skali problemu do rozwiązania, co jest niezwykle pożądane, biorąc pod uwagę fakt, że w 2011 roku liczba zaindeksowanych stron WWW przez Google przekracza 36 mld. Mimo rozwoju technologicznego, wyznaczenie metodą potęgową dominującej wartości własnej z macierzy rzadkiej takich rozmiarów, stanowi nadal ogromne wyzwanie.

Strukturę połączeń stron muzycznych WWW można opisać za pomocą skierowanego grafu połączeń pomiędzy stronami WWW (ang. *directed graph of connections between Websites*), oznaczonego jako GM(V,E), gdzie: GM – Graf Muzyczny (ang. *Music Graph*), V – zbiór wierzchołków grafu (stron WWW), E – zbiór skierowanych krawędzi (połączeń pomiędzy stronami).

Tworzenie grafu GM(V,E) ze stronami WWW powyżej progu tolerancji TL można formalnie zapisać:

$For\ i = 1\ to\ Liczba\ stron\ WWW\ w\ wyszukiwarce$
$\quad If\ card(O_i) > TL\ Then\ dołącz\ i\text{–}tą\ stronę\ do\ zbioru\ wierzchołków\ V\ grafu\ GM(V,E)$
$End\ For$
$\forall_{i,j \in V}\ If\ \big(link(i,j)\big)\ Then\ dołącz\ krawędź(i,j)\ do\ zbioru\ krawędzi\ E\ grafu\ GM(V,E)$

Wyrażenie link(i, j) przyjmuje wartość true, gdy istnieje połączenie z i-tej do j-tej strony WWW, co odpowiada krawędzi w grafie GM, natomiast $card(O_i)$ oznacza liczbę odnośników do różnych plików (nieprowadzących do tego samego pliku muzycznego), wychodzących z i-tej strony WWW. Tak więc $card(O_i)$ jest to liczność zbioru odnośników do plików muzycznych z i-tej strony WWW.

Graf GM(V,E) jest wykorzystywany do utworzenia zredukowanej macierzy sąsiedztwa RAM (*Reduced Adjacency Matrix*) dla stron WWW, zawierających pliki muzyczne. Bezpośrednie generowanie macierzy RAM, odpowiadającej grafowi GM, można przedstawić za pomocą następującego pseudokodu:

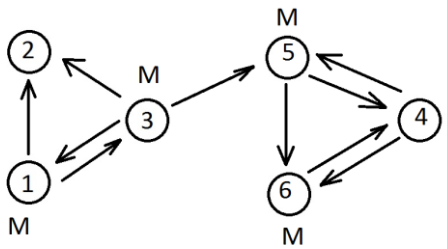
```
For i = 1 To Liczba stron WWW w wyszukiwarce
    If card(O_i) > TL Then
        RAMIndex := RAMIndex + 1            //zwiększenie rozmiaru macierzy
        RAM[1..RAMIndex, RAMIndex] := 0     //dodanie zerowego wiersza
        RAM[RAMIndex, 1..RAMIndex] := 0     //dodanie zerowej kolumny
        Dołącz stronę i do macierzy RAM
    End If
End For
For i = 1 To Liczba Stron w macierzy RAM
For j = 1 To Liczba Stron w macierzy RAM
    If (link(i,j)) Then RAM[i,j]: = 1
End For
End For
```

Przykład 1.

Na rys. 1 przedstawiono skierowany graf powiązań, w którym spośród sześciu stron WWW jedynie cztery strony o numerach 1, 3, 5, 6 mają odnośniki do plików muzycznych. Na rys. 1 węzły te zostały oznaczone literą M.

Macierz sąsiedztwa AM dla tego grafu ma postać:

$$AM = \begin{array}{c} \\ s1 \\ s2 \\ s3 \\ s4 \\ s5 \\ s6 \end{array} \begin{array}{c} \begin{array}{cccccc} s1 & s2 & s3 & s4 & s5 & s6 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array} \tag{1}$$

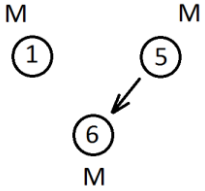


Rys. 1. Skierowany graf powiązań stron WWW
Fig. 1. Directed graph of connections between Websites

Załóżmy, że na stronie *s1* jest umieszczonych 5 plików muzycznych, na stronie *s3* – 2 pliki, na stronie *s6* – 50 plików, a na *s5* – 30 plików. Przyjmując za dolny próg tolerancji *TL* = 3, do budowy zredukowanej macierzy sąsiedztwa RAM będą brane jedynie strony *s1*, *s5* i *s6*, ponieważ strona *s3* zawiera tylko 2 pliki muzyczne (*TL* >2), a strony s2 i s4 nie zawierają ich wcale. Stąd otrzymujemy macierz *RAM* postaci (2).

$$RAM = \begin{array}{c} \\ s1 \\ s5 \\ s6 \end{array} \begin{array}{c} \begin{array}{ccc} s1 & s5 & s6 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \tag{2}$$

W przypadku poszukiwania plików muzycznych przez użytkownika, wykorzystując klasyczne algorytmy (HITS, PageRank, SALSA), macierz RAM zdecydowanie lepiej oddaje rzeczywistą sytuację. Koszty, poniesione na analizę powiązań stron WWW w trakcie weryfikacji tematyki muzycznej, zwrócą się szybko w postaci znacznego zredukowania rozmiarów macierzy sąsiedztwa stron WWW. Graf przedstawiony na rys. 2 odpowiada macierzy (2), utworzonej dla stron muzycznych z przykładu 1.



Rys. 2. Zredukowany skierowany graf powiązań stron WWW
Fig. 2. Reduced directed graph of connections between Websites

Przedstawione w artykule modyfikacje z powodzeniem można zastosować nie tylko dla stron zawierających odnośniki do plików dźwiękowych, ale także przy pozycjonowaniu stron zawierających pliki graficzne lub video.

4. MusicPageRank

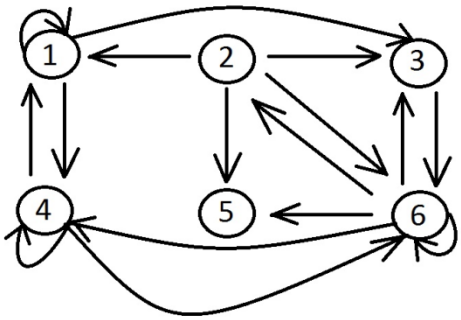
Zredukowaną postać macierzy sąsiedztwa (2) można zastosować bezpośrednio z klasyczną metodą potęgową wyznaczania wektora wynikowego według algorytmu PageRank, jednakże traci się kompletnie informacje, do ilu plików muzycznych dana strona się odwołuje oraz, czy jednocześnie inne strony nie wskazują na ten sam plik. Można wtedy jedynie stwierdzić, że dana strona zawiera nie mniej plików niż ustalony wcześniej poziom tolerancji *TL*.

W zaproponowanym algorytmie MusicPageRank (MPR) każdy element macierzy RAM zostaje zastąpiony przez prawdopodobieństwo wystąpienia pliku muzycznego na stronie:

$$MRAM_{i,j} = \frac{card(O_j)}{\sum_{k=1}^{n} card(O_k)} \cdot RAM_{i,j},$$

gdzie $card(O_j)$ jest liczbą linków do różnych plików muzycznych, wychodzących z *j*-tej strony WWW (kilka odnośników do tego samego pliku traktuje się jako jeden odnośnik). Liczbę tę można potraktować jako liczność zbioru odnośników wychodzących z *j*-tej strony. $\sum_{k=1}^{n} card(O_k)$ oznacza liczbę odnośników do różnych plików, znajdujących się na wszystkich stronach, dla których została utworzona macierz RAM. Innymi słowy jest to suma liczności zbiorów wszystkich odnośników do plików muzycznych, znajdujących się na stronach WWW z macierzy RAM. Zmienna *n* oznacza wymiar macierzy RAM. Element macierzy MRAM, dotyczący strony w *j*-tej kolumnie jest równy prawdopodobieństwu znalezienia odnośnika do pliku muzycznego przy przejściu z *i*-tej na *j*-tą stronę. Ponieważ macierz RAM_{3x3} z przykładu 1 jest zbyt prosta do zilustrowania przekształceń w ramach algorytmu MusicPageRank, dlatego w kolejnym przykładzie do ilustracji działania algorytmu pozycjonowania MPR wykorzystamy macierz RAM_{6x6}.

Przykład 2.



Rys. 3. Przykładowy zredukowany skierowany graf powiązań stron WWW
Fig. 3. An example of reduced directed graph of connections between Websites

$$RAM = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array} \tag{3}$$

Liczba różnych odnośników do plików muzycznych dla stron WWW w macierzy (3) jest następująca: $card(O_1)$=52, $card(O_2)$=38, $card(O_3)$=69, $card(O_4)$=66, $card(O_5)$=95, $card(O_6)$=91. Stąd obliczamy sumę wszystkich, różnych odnośników do plików muzycznych, znajdujących się na stronach w macierzy RAM

$$\sum_{k=1}^{6} card(O_k) = card(O_1) + card(O_2) + card(O_3) + card(O_4) + card(O_5) + card(O_6)$$
$$= 52 + 38 + 69 + 66 + 95 + 91 = 411$$

Niezerowe elementy macierzy RAM zastępujemy następującymi elementami:

$$MRAM_{1,1} = \frac{card(O_1)}{\sum_{k=1}^{6} card(O_k)} \cdot RAM_{1,1} = \frac{52}{411}$$
$$MRAM_{2,1} = \frac{card(O_1)}{\sum_{k=1}^{6} card(O_k)} \cdot RAM_{2,1} = \frac{52}{411}$$
$$MRAM_{2,3} = \frac{card(O_3)}{\sum_{k=1}^{6} card(O_k)} \cdot RAM_{2,3} = \frac{69}{411}$$

Wykonując analogiczne do powyższych przekształcenia wszystkich niezerowych elementów otrzymujemy macierz MRAM:

$$MRAM = \begin{pmatrix} \frac{52}{411} & 0 & \frac{69}{411} & \frac{66}{411} & 0 & 0 \\ \frac{52}{411} & 0 & \frac{69}{411} & 0 & \frac{95}{411} & \frac{91}{411} \\ 0 & 0 & 0 & 0 & 0 & \frac{91}{411} \\ \frac{52}{411} & 0 & 0 & \frac{66}{411} & 0 & \frac{91}{411} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{38}{411} & \frac{69}{411} & \frac{66}{411} & \frac{95}{411} & \frac{91}{411} \end{pmatrix}$$

Dla każdego wiersza macierzy MRAM obliczamy prawdopodobieństwo wystąpienia pliku muzycznego na stronach, do których nie prowadzą odnośniki z innych stron:

$$z_{i=} \frac{1 - \sum_{k=1}^{n} MRAM_{i,k}}{n}$$

Wyrażenie $\sum_{k=1}^{n} MRAM_{i,k}$ oznacza sumę wszystkich elementów dla *i*-tego wiersza macierzy MRAM, skąd otrzymujemy:

$$z_1 = \frac{1 - (\frac{52}{411} + \frac{69}{411} + \frac{66}{411})}{6} = \frac{112}{1233}$$
$$z_2 = \frac{1 - (\frac{52}{411} + \frac{69}{411} + \frac{95}{411} + \frac{91}{411})}{6} = \frac{52}{1233}$$
$$z_3 = \frac{1 - \frac{91}{411}}{6} = \frac{160}{1233}$$
$$z_4 = \frac{1 - (\frac{52}{411} + \frac{66}{411} + \frac{91}{411})}{6} = \frac{101}{1233}$$
$$z_5 = \frac{1 - 0}{6} = \frac{1}{6}$$
$$z_6 = \frac{1 - (\frac{38}{411} + \frac{69}{411} + \frac{66}{411} + \frac{95}{411} + \frac{91}{411})}{6} = \frac{26}{1233}$$

Jak widać (przykładowo dla z_5) wszystkie wiersze zerowe zastępujemy elementami

$$r_{zero} = \frac{\frac{\sum_{k=1}^{n} card(O_k)}{n}}{\sum_{k=1}^{n} card(O_k)} = \frac{1}{n},$$

gdzie *n* jest rozmiarem macierzy MRAM (liczbą wszystkich stron, dla których została utworzona macierz MRAM).

Ostatecznie macierz MPR zostaje utworzona za pomocą dodania prawdopodobieństw z_i do każdego elementu *i*-tego wiersza macierzy MRAM:

$$MPR_{i,j} = MRAM_{i,j} + z_i$$

Macierz potrzebna do obliczenia algorytmu MPR dla przykładowego grafu ma postać:

$$MPR = \begin{pmatrix} \frac{268}{1233} & \frac{112}{1233} & \frac{319}{1233} & \frac{310}{1233} & \frac{112}{1233} & \frac{112}{1233} \\ \frac{208}{1233} & \frac{52}{1233} & \frac{259}{1233} & \frac{52}{1233} & \frac{337}{1233} & \frac{325}{1233} \\ \frac{160}{1233} & \frac{160}{1233} & \frac{160}{1233} & \frac{160}{1233} & \frac{160}{1233} & \frac{433}{1233} \\ \frac{257}{1233} & \frac{101}{1233} & \frac{101}{1233} & \frac{299}{1233} & \frac{101}{1233} & \frac{374}{1233} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{26}{1233} & \frac{140}{1233} & \frac{233}{1233} & \frac{224}{1233} & \frac{311}{1233} & \frac{299}{1233} \end{pmatrix}$$

Tak więc wykorzystując z_i można wyznaczyć dokładną wartość współczynnika teleportacji α znanego z klasycznej wersji algorytmu dla dokumentów tekstowych. Wykorzystując informacje o liczbie niepowtarzających się odnośników do plików muzycznych prowadzących z poszczególnych stron, w prosty sposób zdefiniowano współczynnik α indywidualnie dla każdego niezerowego wiersza macierzy MPR i nie trzeba ustalać tego współczynnika eksperymentalnie.

Aby obliczyć wektor wynikowy MPRV przyjmujemy wektor początkowy o wymiarze *1* x *n*, o dowolnych elementach z przedziału [0,1], przy czym suma jego wszystkich współrzędnych musi być równa 1. Następnie, aż do uzyskania zbieżności wektora, dla zadanej dokładności wykonywane są iteracje:

1. $LastMPRV = MPRV$
2. $NextMPRV = MPRV \cdot MPR$
3. $MPRV = \frac{NextMPRV}{\|NextMPRV\|_1}$
4. *Sprawdź, czy dla zadanej dokładności wektor MPRV jest równy wektorowi otrzymanemu w wyniku poprzedniej iteracji LastMPRV, jeśli tak – koniec (osiągnięto zbieżność), jeśli nie – kontynuuj.*

Dla macierzy MPR z przykładu 2 obliczony numerycznie wektor MPRV ma postać:

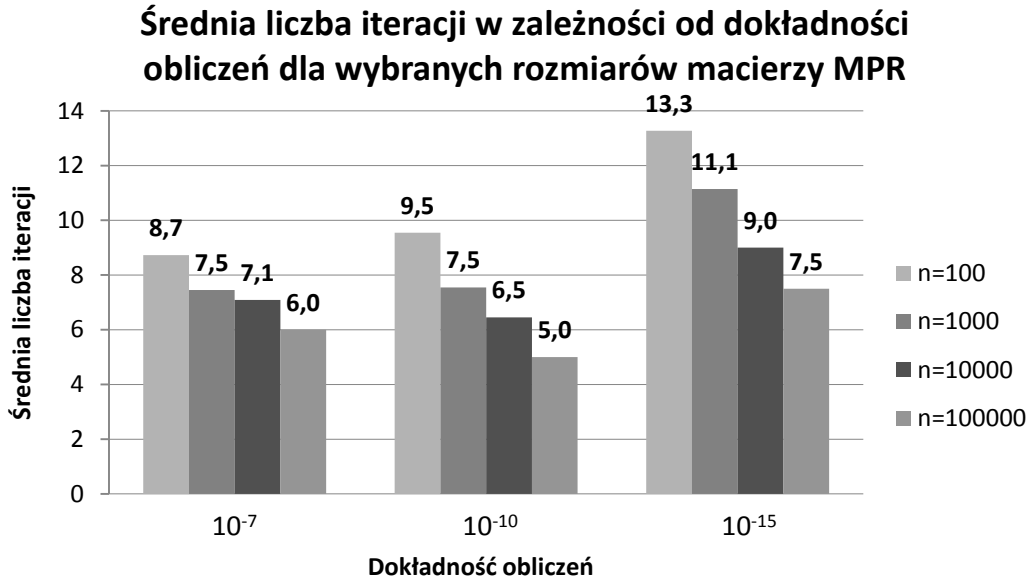
$MPRV = (0{,}14019490;\ 0{,}10865438;\ 0{,}16852394;\ 0{,}17572757;\ 0{,}16703498;\ 0{,}23986421)$

co oznacza, że w wynikach wyszukiwania powinny się pojawić kolejno strony 6, 4, 3, 5, 1, 2.

Główną zaletą algorytmu MPR jest brak przekształcania macierzy MRAM z wykorzystaniem z góry ustalonego współczynnika α, a przede wszystkim wykorzystanie informacji o poszukiwanej przez użytkownika zawartości stron WWW – w tym przypadku plików muzycznych. Ponieważ istotna jest też struktura połączeń pomiędzy stronami WWW, algorytm MPR daje lepsze rezultaty niż wektor prawdopodobieństw wystąpień pliku muzycznego na stronach WWW. Możliwe jest wykorzystanie algorytmu MPR także do niezredukowanej macierzy sąsiedztwa stron AM – wtedy wszystkie odnośniki do stron, które nie zawierają plików muzycznych zostaną zastąpione zerami. Plusem wykorzystania wszystkich stron WWW będzie oryginalna struktura połączeń pomiędzy stronami. Minusem takiego rozwiązania będzie przypisanie niezerowego prawdopodobieństwa wystąpienia odnośników do plików muzycznych także tym stronom, które nie zawierają żadnego pliku muzycznego – zmniejszając tym samym prawdopodobieństwo dla tych stron, które rzeczywiście te odnośniki zawierają. Ze względu na większe rozmiary macierzy AM niż macierzy RAM, trzeba liczyć się ze wzrostem złożoności obliczeniowej i pamięciowej, w stosunku do wersji MPR dla zredukowanej macierzy sąsiedztwa RAM. Zaproponowany algorytm MPR może być wykorzystany także dla innych źródeł danych niż pliki muzyczne, dźwiękowe czy tekstowe, na przykład dla plików graficznych, video.

Dla zaproponowanego algorytmu MPR zostały wykonane testy liczby iteracji koniecznych do wykonania w metodzie potęgowej, w celu uzyskania odpowiedniej dokładności obliczeń wektora MPRV. Testy zostały wykonane na komputerze z pamięcią operacyjną wielkości 4GB.

Wykonane testy dowodzą zadowalającej szybkości zbieżności wektora wynikowego MPRV. Najważniejszy jest tu fakt, że nie zastosowano żadnej metody przyspieszającej zbieżność wektora wynikowego w metodzie potęgowej, jak np. ekstrapolacja kwadratowa. Dzięki otrzymanym wynikom można stwierdzić, że algorytm MPR nadaje się do praktycznego zastosowania. W porównaniu z dotychczasowymi wynikami badań implementacji klasycznej wersji algorytmu PageRank, na przykład dla macierzy o rozmiarze 1000 x 1000 (gdzie konieczne było wykonanie średnio 250 iteracji, aby osiągnąć wektor rankingowy obliczony z dokładnością 10^{-15} [1]), przy zastosowaniu algorytmu MPR otrzymano prawie trzydziestokrotne zmniejszenie liczby iteracji.



Rys. 4. Wyniki testów dla algorytmu MPR
Fig. 4. Tests results for MPR algorithm

5. Podsumowanie

W artykule zaproponowano algorytm, dostarczający informacje kontekstowe o plikach muzycznych, na podstawie sieci połączeń pomiędzy stronami WWW. Otrzymane wyniki dowodzą, że istniejący i dobrze zbadany algorytm pozycjonujący dokumenty tekstowe PageRank może być odpowiednio przystosowany też do innych źródeł informacji niż pliki tekstowe. Wykonane testy dowodzą, że zaproponowany algorytm MusicPageRank nadaje się do praktycznych zastosowań. Algorytm ten, w przeciwieństwie do algorytmu PageRank, nie wymaga stałej wartości współczynnika α. Wykorzystanie informacji o liczbie różnych odnośników do plików muzycznych pozwala lepiej odwzorować rzeczywistość panującą w sieci Web.

BIBLIOGRAFIA

1. Czyżowicz M.: Badanie porównawcze metod inteligentnej nawigacji i adaptacja algorytmów optymalizacyjnych do zadań nawigacji w zbiorach dokumentów tekstowych. Praca magisterska, Politechnika Warszawska, Warszawa 2003.
2. Langville A., Meyer C.: A survey of eigenvector methods for Web Information Retrieval. SIAM (Society for Industrial and Applied Mathematics) Review, Vol. 47, No. 1, March 2005, s. 135÷161.

3. Langville A., Meyer D.: Deeper Inside Page Rank. Internet Mathematics Vol. 1, No. 3, 2004, s. 335÷380.
4. Langville A., Meyer C.: Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press, 2006.
5. Manning Ch., Raghavan P., Schutze H.: Introduction to Information Retrieval. Cambridge University Press, 2008.
6. Mazur Z.: Modele i modyfikacje rozproszonych systemów wyszukiwania informacji opartych na tezaurusach z wagami. Prace Naukowe Centrum Obliczeniowego Politechniki Wrocławskiej, Wrocław 1989.
7. Mazur Z., Wiklak K.: Music Information Retrieval on The Internet. Chapter 22 in Advances in Multimedia and Network Information System Technologies, Springer 2010.
8. Meyer C.: Matrix Analysis and Applied Linear Algebra. SIAM Society for Industrial and Applied Mathematics, 2004.
9. Prystowsky J., Gill L.: Calculating Web Page Authority Using the Page Rank Algorithm. Math 45, Fall 2005.
10. Wiklak K.: Internetowe systemy wyszukiwania informacji muzycznej. Praca magisterska, Politechnika Wrocławska, Wrocław 2010.

Recenzent: Dr hab. inż. Marcin Gorawski, prof. Pol. Wrocławskiej

Wpłynęło do Redakcji 31 stycznia 2011 r.

Abstract

Although the music information retrieval is already well described topic it still remains as a research subject of many scientists. Recently popular MP3 search engines are currently replaced with more advanced music information retrieval systems, that provide new ways of creating queries by user – like *query by humming* or *query by singing*. The files returned as a result system response are ordered with selected ranking algorithm. Because audio files are different data source than text, it give a new way to calculate final rank of music file, returned as a result. In fact there are query dependent (dynamic) and query independent (static) measures. Query dependent measures are based on relevance between information given by user as a query input data and the meta data stored in a file. Meta data can also be used in calculation of static measures, like relevance between file meta data and its original values. Although meta data are very useful, it do not provide any information about music file context - for example: if the audio file is referenced from the popular Website. That causes necessity of

applying Web structure algorithms, that use connections (links) between sites as a primary knowledge source about Websites. Such algorithms are HITS, PageRank, SALSA. After short analysis of pros and cons of each, the PageRank have been chosen as a base of developing new algorithm, that uses also music file links – MusicPageRank. Section 1 is a short introduction. Section 2 describes result music files ranking problem, including query dependent and independent measures. This section provides a general formula of calculating static FileRank measure and describes role of meta data, contextual file information and individual preferences in calculating FileRank. Section 3 describes methods of modification link based algorithms to work with music information retrieval systems. The Web is described as a Directed graph of connection between Websites (Fig 1) and as a site Adjacency Matrix (1). A transformation from Adjacency Matrix to Reduced Adjacency Matrix (2), which is build using Websites, that contain links to music files is also described. Section 4 describes a proposed MusicPageRank algorithm (MPR). The MPR is discussed using example graph on (Fig. 2), which corresponds to Reduced Adjacency Matrix (3). The calculation of result rank vector is described with using a iterative power method. This section ends with MPR test results, presented on (Fig 3). The tests results have prooved that MPR is fast convergent algorithm, that can be applied in real music search engines.

Adresy

Zygmunt MAZUR: Politechnika Wrocławska, Instytut Informatyki, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Polska, zygmunt.mazur@pwr.wroc.pl.
Konrad WIKLAK: Politechnika Wrocławska, Instytut Informatyki, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Polska, konrad.wiklak@pwr.wroc.pl.