Web Usage Analysis: Mining Frequent Patterns

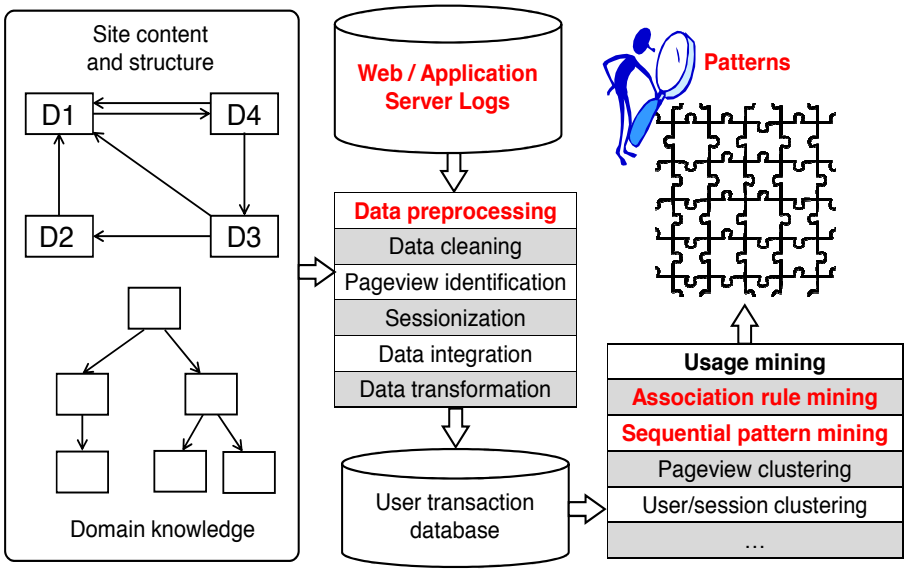
Miłosz Kadziński

Institute of Computing Science
Poznan University of Technology, Poland

www.cs.put.poznan.pl/mkadzinski/wpi

[1] Eksploracja zasobów internetowych polega na odkrywaniu nieoczywistej, potencjalnie przydatnej wiedzy z zawartości, struktury oraz użytkowania sieci. Poprzedni wykład poświęcony był algorytmom, pozwalającym na uszeregowanie stron ze względu na jakość połączeń. Ten wykład jest pierwszym z serii trzech poświęconych analizie użytkowania sieci. Kolejne będą dotyczyły miedzy innymi klasyfikacji, rekomendacji oraz grupowaniu, a dziś naszym głównym przedmiotem zainteresowania będzie odkrywania wzorców częstych, zbiorów lub sekwencji.

Web Usage Analysis (1) - Agenda



[2] Wykład będzie składał się z trzech części poświęconych analizie kliknięć czy też równoważnie wizyt użytkowników na stronach. W pierwszej części omówione zostaną formaty plików logu, które przechowują informację o użytkowaniu danego serwisu. W drugiej skupimy się na etapach wstępnego przetwarzania danych dla potrzeb ich dalszej eksploracji. Odniesiemy się m.in. do czyszczenia danych, identyfikacji użytkowników i sesji oraz uzupełniania ścieżek. Doprowadzi to ostatecznie do powstania bazy danych transakcji zrealizowanych przez użytkowników. W ostatniej części przedstawione zostaną właściwe techniki analizy użytkowania pozwalające na odkrywanie częstych zbiorów i sekwencji oraz indukcję reguł asocjacyjnych. Najważniejszym tematem wykładu będzie algorytm Apriori, który, podobnie jak PageRank, należy do dziesięciu najsławniejszych oraz najbardziej przydatnych algorytmów w dziedzinie analizy danych. W dalszej części wykładu pojawią się też inne zaawansowane zagadnienia, jak choćby wykorzystanie łańcuchów Markova w kontekście okdrywania wzorców nawigacyjnych.

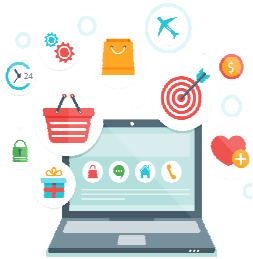
Web Usage Analysis (2) - Sources of Data

Web Usage Mining

- Discovery of meaningful patterns from data generated by user access to resources web/application servers

Typical Sources of Data

- Clickstream data from Web/application server logs or third-party page tagging services
- E-commerce and product-oriented user events (e.g., shopping cart changes, product click-throughs, purchases, etc.)
- User profiles data, user ratings, user contributed data (tags, comments, reviews)
- Product meta-data, page content, site structure



User Transactions

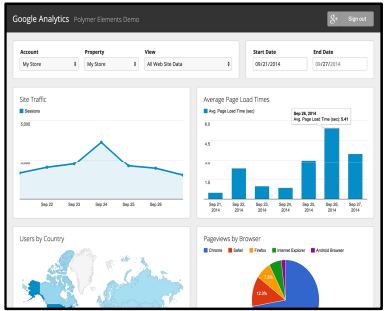
- Sets or sequences of pageviews possibly with associated weights

[3] Omówimy problemy i metody eksploracji danych opisujących korzystanie z sieci Internet. Celem takiej eksploracji jest odkrywanie ogólnych wzorców zachowań na podstawie danych wygenerowanych przez użytkowników w czasie realizacji przez nich dostępu do zasobów internetowych. Źródła pochodzenia takich danych mogą być bardzo różne. Z jednej strony mogą być one wygenerowane w sposób pośredni, nieintencjonalny, przez użytkowników odwiedzających strony lub robiących zakupy w sklepach internetowych. Z drugiej strony mogą być one wygenerowane celowo, jak choćby w przypadku danych dotyczących profili użytkowników, ich ocen, recenzji, tagów lub komentarzy. Dla potrzeb wykładu skupimy się na transakcjach użytkowników rozumianych jako zbiory lub sekwencje odwiedzin stron, z którymi potencjalnie mogą być skojarzone różne wagi.

Web Usage Analysis (3) - Analytics vs. Mining

Web Analytics

- Refers to the measurement, analysis, and reporting of user behavior
- Usually involves descriptive statistics from clickstream and other user behavior data at different levels of aggregations across predetermined dimensions: time, content/ product categories, referring sites, etc.
- Many tools and third party services available (e.g., **Google Analytics**)



Web Usage Mining

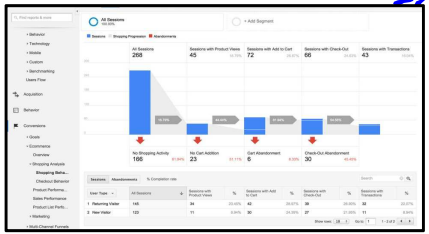
- Goes beyond basic analytics to discover patterns in usage data, identify and characterize important customer segments, find affinities across pages or products, build models to predict future behavior, etc.

[4] W zakresie metod eksploracji danych opisujących korzystanie z sieci wyróżnia się dwa podstawowe pojęcia, analitykę sieci oraz eksplorację użytkowania sieci. Analityka nawiązuje do opisu liczbowego, analizy oraz raportowania zachowania użytkowników. Zwykle stosowane są to stosunkowo proste techniki, bazujące na statystyce odwiedzin lub zachowania użytkowników w kontekście różnych wymiarów takich jak czas, zawartość strony lub strona odsyłająca. Najbardziej popularnym na świecie narzędziem stosowanym w tym celu jest Google Analytics, które dostarcza ponad 80 różnych raportów. Eksploracja użytkowania sieci idzie o krok dalej, skupiając się na odkrywaniu wzorców użytkowania, powiązań między stronami, segmentacji użytkowników lub predykcji ich przyszłych zachowań. Ze zrozumiałych względów eksploracja użytkowania jest więc bardziej interesująca niż analityka.

Web Usage Analysis (4) - Important Questions

- Which web page is the most common **entry** for users?
- Which other web sites **referred** the user to our web site?
- How **many pages** have been viewed in a typical visit?
- How **long** does the typical visitor stay on our web site?
- **When** is our web site the **most popular**?
- Which web page is the most common **departure** point?

- Which pages have been **viewed together**?
- In which **order** have the pages been viewed?



[5] Analityka prowadzi do odpowiedzi na proste pytania, ale nawet one mogą prowadzić do wartościowych wniosków. Przykładowo, najpopularniejszy punkt wejścia do naszego serwisu lub identyfikacja stron odsyłających pozwala zrozumieć kim są nasi użytkownicy, jakie mają zainteresowania albo czy prowadzona kampania reklamowa jest skuteczna. Odwołanie do liczby odwiedzanych stron, spędzanego na nich czasu albo identyfikacja najbardziej popularnych stron mówią nam dużo o strukturze serwisu, potencjalnej konieczności jej poprawy albo zmiany niektórych treści, itd. W przypadku eksploracji użytkowania stosuje się różnorodne, bardziej zaawansowane metody. Przedmiotem naszego zainteresowania będzie analiza stron, który były odwiedzane przez użytkownika w pojedynczej sesji oraz analiza kolejności takich odwiedzin. Kluczowe będzie tu uwzględnienie przymiotnika zzczęste" tak, by odwiedziny i wspólne wystąpienia miały względnie wysokie potwierdzenie w danych.

Log Files (1)

Server Log Files

- Passive data collection
- Data is always available and does not depend on client setup
- Data belongs to the organization (access to full data)

Page Tagging

- Active (client-side) data collection
- Often requires a third party to implement – a vendor supplies page tags, collects data and analyzes it to generate reports
- Usually involves adding code (Javascript) to each page that when loaded, sends back information to vendor

[6] Wyróżniamy dwa podstawowe techniczne sposoby pozyskiwania danych o odwiedzinach użytkowników. Pierwszym jest odwołanie się do wiedzy zgormadzonej w plikach log serwera, które można traktować jako dzienniki lub rejestry zdarzeń. Sposób ten można traktować jako pasywny, dane zbierane są po stronie serwera i nie zależą od ustawień użytkowników. W związku z tym takie dane są zawsze dostępne i w całości należą do twórcy serwisu. Logi serwerów przechowują olbrzymie ilości informacji dotyczące realizowanych dostępów do stron i stanowią potencjalnie ważne źródło opisu zachowań użytkowników serwera. Choć sposób ten ma wiele zalet, to wiążą się z nim też pewne wady związane choćby z niemożnością dokładnego obliczenia czasu spędzonego przez użytkownika na ostatniej stronie w sesji oraz z przechowywanie stron w pamięci podręcznej. Drugi sposób to tagowanie stron, oznaczające aktywne pobieranie danych po stronie klienta. Wiąże się on często z wykorzystaniem takich technologii jak JavaScript czy Ajax, a najłatwiej wyobrazić sobie jego istotę przez odwołanie do prostych liczników, które są widoczne dla użytkowników i pokazują, ile razy dana strona została wyświetlona. Page tagging często wymaga skorzystania z usług zewnętrznych oraz dodanie kodu javascriptowego do każdej strony, która po załadowaniu, odsyłałaby dedykowaną informację, którą następnie można przetworzyć w celu wygenerowania raportu użytkowania.

Log Files (2)

- For each request from a user's browser to a web server, a response is generated automatically
- The response takes the form of a simple single-line transaction record appended to an ASCII text file on the web server
- Text file may be comma-, space-, or tab-delimited
- When loading a particular page, the browser also requests all objects embedded in the page such as .gif or .jpg graphic files

```
141.243.1.172 [01/Jun/2018:03:09:21 -0600]
      "GET /Software.html HTTP/1.0" 200 1497
wpbfl2-45.gate.net [01/Jun/2018:03:10:01 -0600]
      "GET /default.htm HTTP/1.0" 200 4889
wpbfl2-45.gate.net [01/Jun/2018:03:10:02 -0600]
      "GET /icons/circle_logo_small.gif HTTP/1.0" 200 2624
```


[7] W czasie wykładu skupimy się na danych dostępnych w plikach logu serwera. Sposób ich tworzenia sprowadza się do zapisu odpowiedzi wygenerowanej przez serwer na żądanie wygenerowane z wyszukiwarki użytkownika. Odpowiedź jest linią w postaci tekstowej, w której wyróżnia się wiele odseparowanych od siebie pól. Warto zwrócić uwagę, że takie odpowiedzi nie dotyczą tylko żądania strony, bo bezpośrednie żądanie użytkownika pociąga za sobą automatyczne żądania obiektów osadzonych na stronie, takich jak pliki graficzne, dźwiękowe lub stylu. Przykład takiej automatycznej serii żądań zaprezentowano w dolnej części slajdu.

Log Files (3) - What is in a typical server log?

<host_field> <date> [<method> <file> <protocol>] <code> <bytes>

```
141.243.1.172 [01/Jun/2018:03:09:21 -0600]
                "GET /Software.html HTTP/1.0" 200 1497
```

- **Remote host field** – IP address (domain name) of the host making the request
- **Date/time field** – DD/Mon/YYYY:HH:MM:SS offset (w.r.t. Greenwich)
- **HTTP request field**
  - The request metod (GET, HEAD, PUT, POST)
  - The uniform resource identifier (URI)
  - The header and the protocol
- **Status code field** (2 – success, 3 – redirect., 4 – client error, 5 – server error)
  - 200: sucess, 202: accepted, 301: moved permanently
  - 403: forbidden, 404: not found, 500: internal server error
- **Transfer volume (byte) field**

[8] Omówienie formatów plików logu zaczniemy od najprostszego formatu, w którym każda linia składa się tylko z pięciu pól. Pierwsze to pole adresu IP użytkownika generującego żądania. Może ono mieć postać liczbowego adresu IP lub tekstowej nazwy hosta. Drugie pole ma charakter złożony i dotyczy daty. Zawiera informacje o dniu, godzinie oraz przesunięciu względem czasu uniwersalnego. Trzecie pole jest także złożone i dotyczy żądania HTTP. Zawiera metodę (najczęściej GET), identyfikator zasobu, nagłówek i nazwę protokołu. Czwarte pole odwołuje się do kodu odpowiedzi. Przykładowo, kody zaczynające się od 2 (jak 200) oznaczają powodzenie, a od 4 błąd po stronie błąd po stronie klienta (jak 404). Ostatnie pole ujmuje wielkość transferu w bajtach, tj. wielkość pliku przesłanego przez serwer w odpowiedzi na żądanie. Pole to jest wypełniane tylko w przypadku pozytywnej odpowiedzi.

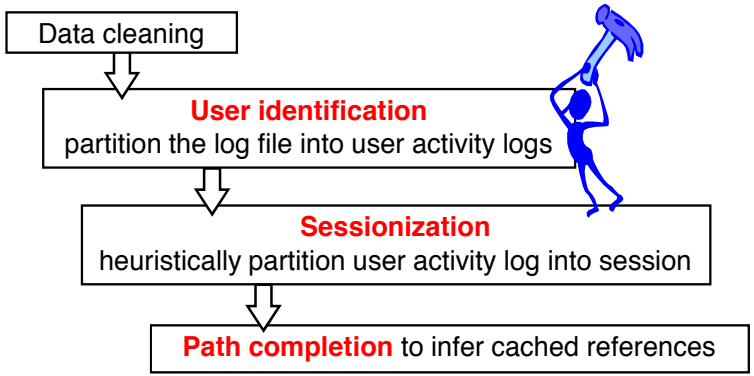
Log Files (3) - Common Log Formats

```
203.30.5.145 -- smith [01/Jun/2018:03:09:21 -0600]
   "GET /Software.html HTTP/1.0" 200 3942
      "http://www.referrer.com/?query=soft" "Mozilla/58 (Win10)"
```

- **Common Log Format (CLF)**
  - Identification Field – stores identity information provided by the client if the web server is performing an identity check (seldom used)
  - Authuser Field – stores the authenticated client user name, if it is required to gain access to passwod protected directories
- **Extended Common Log Format (ECLF)**
  - Referrer Field – the URL of the previous site visited by the client (tracks how people found your site)
  - User Agent Field – the client's browser, the browser version, and the operating system (sometimes information about bots)
- Microsoft IIS Log Format and many others

[9] Najbardziej popularnym formatem logu jest Common Log Format (w skrócie CLF). Nadbudowuje on format omówiony na poprzednim slajdzie o dwa dodatkowe pole związane z identyfikacją użytkownika. Jedno z nich dotyczy informacji identyfikującej w postaci prostego nieszyfrowanego tekstu (rzadko używane), a drugie przechowuje informację o nazwie użytkownika w przypadku, gdy dostęp do jakiegoś zasobu jest chroniony hasłem. W rozszerzonym formacie CLF (w skrócie ECLF) znajduje się w sumie 9 pól, w tym 2 nowe kluczowe dla identyfikacji użytkownika i sesji. Pierwsze z nich wskazuje na stronę, z której wygenerowano żądanie, co pozwala na ustalenie, jak użytkownik znalazł naszą/kolejną stronę. Drugie pole wskazuje na przeglądarkę klienta oraz system operacyjny. W przypadku gdy użytkownikiem jest crawler, można go po tym polu zidentyfikować. Oczywiście istnieje wiele innych formatów logu, ale już analiza tych podstawowych daje wyobrażenie dotyczące zakresu dostępnych w nich informacji.

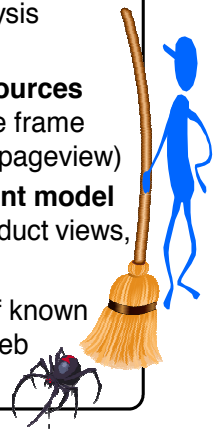
Date Preprocessing - Different Stages



[10] Przed realizacją właściwej eksploracji należy przetworzyć surowy plik logu do postaci adekwatnej dla właściwej analizy. Takie przetwarzanie składa się z czterech podstawowych etapów, które opierają się na wykorzystaniu dedykowanych heurystyk. Pierwszy z nich polega na czyszczeniu danych, tj. eliminacji wpisów zbędnych w kontekście analizy rzeczywistych zachowań użytkowników. Drugi dotyczy identyfikacji użytkowników, tj. podziału wpisów znajdujących się w logu na części związane z unikalnym użytkownikiem. Trzeci sprowadza się do identyfikacji sesji takich użytkowników, a czwarty to uzupełnienie ścieżek w ramach sesji tak, by zaadresować problem przechowywania stron w pamięci podręcznej. Tych etapów może być oczywiście więcej, np. jeden z często realizowanych sprowadza się do integracji wielu plików logów serwera.

Date Cleaning

- Removing **extraneous reference to embedded objects** that may not be important for the purpose of the analysis (styles, graphics, sound files)
- Each pageview is **a collection of web objects or reources representing a specific user event** (only for a single frame site, each file has one-to-one correspondance with a pageview)
- Sometimes requires **a priori specification of an event model** based on which user actions can be categorized (product views, registration, shopping card changes, purchases, etc.)
- Remove reference due to **spider navigation** (a list of known crawlers, ”robot.txt”, typical non-human behavior of web crawlers (many short visits))



[11] Czyszczenie danych jest bardzo rozległym pojęciem, któremu można by poświęcić osobny wykład. My skupimy się na dwóch podstawowych zagadnieniach. Pierwsze z nich dotyczy usunięcia wpisów dotyczących obiektów osadzonych na stronach, które nie są istotne z punktu widzenia właściwej analizy. Krok ten opiera się na analizie statystyk dotyczących rozszerzeń plików i zwykle wymaga określenia z góry, które formaty lub kategorie zdarzeń reprezentują zasoby istotne dla eksploracji. Z jednej strony możliwe jest więc choćby wyeliminowanie plików graficznych, dźwiękowych lub stylu, a z drugiej można też podać, że przedmiotem analizy jest choćby wyświetlenie kart produktów, włożenie ich do koszyka lub dokonanie zakupu. Drugie zagadnienie jest związane z usunięciem wpisów wynikających z działania crawlerów. Ich zachowanie jest bowiem zupełnie inne niż ludzi; odwiedzają one wszystkie strony i spędzają na nich bardzo mało czasu. Identyfikacja taka odwołuje się albo do nazwy użytkownika w polu przeglądarki lub jest realizowana przez wykrycie nietypowego zachowania omówionego powyżej.

User Identification

Method	Description	Privacy concerns	Advantages	Disadvantages
IP + Agent	Each unique IP/Agent pair is a unique user	Low	Always available. No technology needed.	Not guarantee to be unique. Rotating IPs.
Registration	User logs in to the site	Medium	Track individuals not browsers	Many users won't register. Not available before.
Cookies	Save ID on the client's machine	Medium to high	Track repeat visits from same browser	Can be easily turned off
Software agents	Program loaded into browser and sends back data	High	Accurate data for a single user	Likely to be rejected by users

[12] Realizacja etapu identyfikacji użytkownika zależy od formatu eksploatowanego pliku logu oraz sposoby dostępu do zasobów realizowanego przez użytkowników. Przykładowo, jeśli opieramy się tylko na logu to informacja, którą możemy heurystycznie wykorzystać do identyfikacji użytkownika to połączenie adresu IP oraz pola przeglądarki. Jest to sposób zawsze dostępny, bo korzysta z jawnych danych. W przypadku wymogu zalogowania lub rejestracji dokonuje się bezpośredniej identyfikacji użytkowników, a nie wyszukiwarek. Sposób ten jest jednak odrzucany przez wiele osób. W praktyce najczęściej informacja z plików logu jest łączona z informacją zawartą w ciasteczkach, które zapisywane są na maszynie końcowej. Dane osobowe gromadzenie przy użyciu ciasteczek mogą być zbierane wyłącznie w celu wykonywania określonych funkcji na rzecz użytkownika, czyli np. zapamiętania logowania do serwisu. Takie dane są zaszyfrowane w sposób uniemożliwiając dostęp do nich osobom nieuprawnionym.

Sessionization

Time-Oriented Heuristics

- ***h1***: Total session duration may not exceed a threshold θ.
  Given t_0, the timestamp for the first request in a constructed session S, the request with timestamp t is assigned to S, iff t - $t_0 \leq \theta$.
- ***h2***: Total time spent on a page may not exceed a threshold δ.
  Given t_1, the timestamp for request assigned to constructed session S, the next request with timestamp t_2 is assigned to S, iff t_2 - $t_1 \leq \delta$.

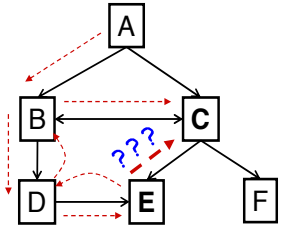
Referrer-Based Heuristic

- ***href***: Given two consecutive requests p and q, q is assigned to S, if the referrer for q was previously invoked in S (in case of a conflict, assign q to the last open session).

Note: in practice, it is often useful to use a combination of time- and navigation-oriented heuristics in session identification

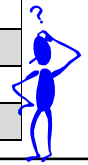
[13] Identyfikacja sesji w oparciu o plik logu jest zadaniem czysto heurystycznym. Istnieją trzy dedykowane do tego procedury: h1, h2 oraz href. h1 zakłada maksymalny czas trwania pojedynczej sesji, stąd kolejne żądania są dodawane do danej sesji o ile od pierwszego żądania nie minął dłuższy czas niż ten predefiniowany próg. h2 także opiera się na wykorzystaniu progu, ale dotyczącego maksymalnej odległości czasowej między kolejnymi żądaniami. Rozległe badania przeprowadzone w wielu ośrodkach na świecie wskazują, że wartość takiego progu powinna wynosić ok. 25-30 minut. Ostatnia heurystyka href zakłada, że dostępna jest informacja o polu strony odsyłającej. Jej analiza dopuszcza dodanie kolejnego żądania do otwartej sesji tylko jeśli strona odsyłająca już się w niej znajduje. W przypadku wielu otwartych sesji, do których takie bieżące żądanie można dodać, konflikt rozwiązywany jest poprzez wybór sesji najświeższej, ostatnio otwartej. W praktyce często stosuje się kombinacje heurystyk opartych na analizie czasu (jak h1 i h2) oraz nawigacji między stronami (jak href).

Path Completion



Server log	
URL	Referrer
A	-
B	A
D	B
E	D
C	B

User's actual navigation path:
A → B → D → **E** → D → B → **C**



- Need knowledge of link structure to complete the navigation path
- There may be multiple candidate for completing the path. For example consider the two paths : **E** → D → B → **C** and **E** → D → B → A → C.
- In this case, the referrer field allows us to partially disambiguate
- One heuristic: always take the path that requires the fewest number of "back" references

[14] Ostatni etap wstępnego przetwarzania polega na uzupełnianiu ścieżek, tj. dodaniu do sesji żądań stron, których nie ma w pliku logu, a które najprawdopodobniej zostały załadowane z pamięci podręcznej po naciśnięciu przycisku Wstecz lub podobnego. Do realizacji tego etapu konieczna jest znajomość struktury połączeń między stronami w serwisie. Przykładowo, analiza sesji pokazanej na slajdzie wskazuje, że nie ma zgodności w stronach odsyłających pomiędzy żądaniami stron E oraz C. Struktura połączeń serwisu wskazuje też, że nie ma linku umożliwiającego przejście z E do C. Zadanie polega na dodanie takich żądań, które uczynią przejście realistycznym. W przykładzie istnieją dwa możliwe uzupełnienie, przez D i B lub dodatkowo jeszcze przez A. Na rozstrzygnięcie pozwala tu analiza stron odsyłających spośród wcześniej wygenerowanych żądań, a heurystyczne podejście zakłada, że uzupełnienie powinno wymagać tak małej liczby żądań, jak to tylko możliwe. Dla analizowanego przykładu założylibyśmy więc, że użytkownik z E cofnął się do D, potem do B (patrz pole odsyłające), a dopiero potem do C.

Date Preprocessing - Example (1)

Time	IP	URL	Ref	Agent
00:01	1.2.3.4	A	-	IE5;Win2k
00:09	1.2.3.4	B	A	IE5;Win2k
00:10	2.3.4.5	C	-	IE4;Win98
00:12	2.3.4.5	B	C	IE4;Win98
00:15	2.3.4.5	E	C	IE4;Win98
00:19	1.2.3.4	C	A	IE5;Win2k
00:22	2.3.4.5	D	B	IE4;Win98
00:22	1.2.3.4	A	-	IE4;Win98
00:25	1.2.3.4	E	C	IE5;Win2k
00:25	1.2.3.4	C	A	IE4;Win98
00:33	1.2.3.4	B	C	IE4;Win98
00:58	1.2.3.4	D	B	IE4;Win98
01:10	1.2.3.4	E	D	IE4;Win98
01:15	1.2.3.4	A	-	IE5;Win2k
01:16	1.2.3.4	C	A	IE5;Win2k
01:17	1.2.3.4	F	C	IE4;Win98
01:26	1.2.3.4	F	C	IE5;Win2k
01:30	1.2.3.4	B	A	IE5;Win2k
01:36	1.2.3.4	D	B	IE5;Win2k

Sort users (based on IP+Agent)

User 1

00:01	1.2.3.4	A	-	IE5;Win2k
00:09	1.2.3.4	B	A	IE5;Win2k
00:19	1.2.3.4	C	A	IE5;Win2k
00:25	1.2.3.4	E	C	IE5;Win2k
01:15	1.2.3.4	A	-	IE5;Win2k
01:16	1.2.3.4	C	A	IE5;Win2k
01:26	1.2.3.4	F	C	IE5;Win2k
01:30	1.2.3.4	B	A	IE5;Win2k
01:36	1.2.3.4	D	B	IE5;Win2k

User 2

00:10	2.3.4.5	C	-	IE4;Win98
00:12	2.3.4.5	B	C	IE4;Win98
00:15	2.3.4.5	E	C	IE4;Win98
00:22	2.3.4.5	D	B	IE4;Win98

User 3

00:22	1.2.3.4	A	-	IE4;Win98
00:25	1.2.3.4	C	A	IE4;Win98
00:33	1.2.3.4	B	C	IE4;Win98
00:58	1.2.3.4	D	B	IE4;Win98
01:10	1.2.3.4	E	D	IE4;Win98
01:17	1.2.3.4	F	C	IE4;Win98

[15] Aby podsumować dotychczas omówione etapy przeanalizujemy plik logu składający się z 19 linii. Jest on przedstawiony w postaci tabelarycznej; każdy wiersz odpowiada żądaniu, a każda kolumna innemu polu. Jeśli do identyfikacji użytkowników wykorzystać by tylko adres IP, to wyróżnilibyśmy dwie osoby. Jeśli do tego dodać jeszcze pole przeglądarki, to unikalnych kombinacji (IP, przeglądarka) są trzy. Wpisy można więc podzielić na trzy grupy przedstawione z prawej strony, składające się z odpowiednio 9, 4 i 6 żądań.

Date Preprocessing - Example (2)

Time	IP	URL	Ref	Agent
00:22	1.2.3.4	A	-	IE4;Win98
00:25	1.2.3.4	C	**A**	IE4;Win98
00:33	1.2.3.4	B	**C**	IE4;Win98
00:58	1.2.3.4	D	**B**	IE4;Win98
01:10	1.2.3.4	E	**D**	IE4;Win98
01:17	1.2.3.4	F	**C**	IE4;Win98

The referrer-based heuristics will result in a single session:

for C – A was previously invoked,

for B – C was previously invoked,

…

The *h*1 heuristic with timeout = 30 minutes will result in two sessions

00:22	1.2.3.4	A	-	IE4;Win98
00:25	1.2.3.4	C	A	IE4;Win98
00:33	1.2.3.4	B	C	IE4;Win98

more than 30 minutes have passed

00:58	1.2.3.4	D	B	IE4;Win98
01:10	1.2.3.4	E	D	IE4;Win98
01:17	1.2.3.4	F	C	IE4;Win98

The *h*2 heuristic with timeout = 10 minutes will result in three sessions

00:22	1.2.3.4	A	-	IE4;Win98
00:25	1.2.3.4	C	A	IE4;Win98
00:33	1.2.3.4	B	C	IE4;Win98

more than 10 minutes have passed

00:58	1.2.3.4	D	B	IE4;Win98

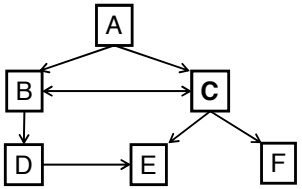
more than 10 minutes have passed

01:10	1.2.3.4	E	D	IE4;Win98
01:17	1.2.3.4	F	C	IE4;Win98

[16] Dla każdego użytkownika z osobna identyfikowane są sesje. Skupmy się na użytkowniku trzecim (User 3). Heurystyka opierająca się na wykorzystaniu strony odsyłającej zakończyłaby się tylko jedną sesją (patrz górna część slajd). Nie jest ważne, czy stroną odsyłającą jest poprzednią strona w sesji. Przykładowo, dla żądania strony F stroną odsyłającą jest C, a poprzednie żądanie dotyczy E. Istotne jest tylko, że C było już żądane w tej sesji (patrz drugi wpis). Heurystyka h1 z progiem 30-minutowym prowadzi do rozróżnienie dwóch sesji. Dla czwartego wpisu czas, który minął od pierwszego żądania jest dłuższy niż 30 minut. To żądanie inicjuje więc nową sesję. Wynik zastosowania heurystyki h2 z progiem 10-minutowym to trzy sesje. Pomiędzy 3 i 4 żądaniem oraz pomiędzy 4 i 5 żądania minęło więcej niż 10 minut, co prowadzi do ustalenia granic między sesjami właśnie w tych miejscach.

Date Preprocessing - Example (3)

Time	IP	URL	Ref	Agent
00:22	1.2.3.4	**A**	-	IE4;Win98
00:25	1.2.3.4	**C**	A	IE4;Win98
00:33	1.2.3.4	**B**	C	IE4;Win98
00:58	1.2.3.4	**D**	B	IE4;Win98
01:10	1.2.3.4	**E**	D	IE4;Win98
01:17	1.2.3.4	**F**	C	IE4;Win98



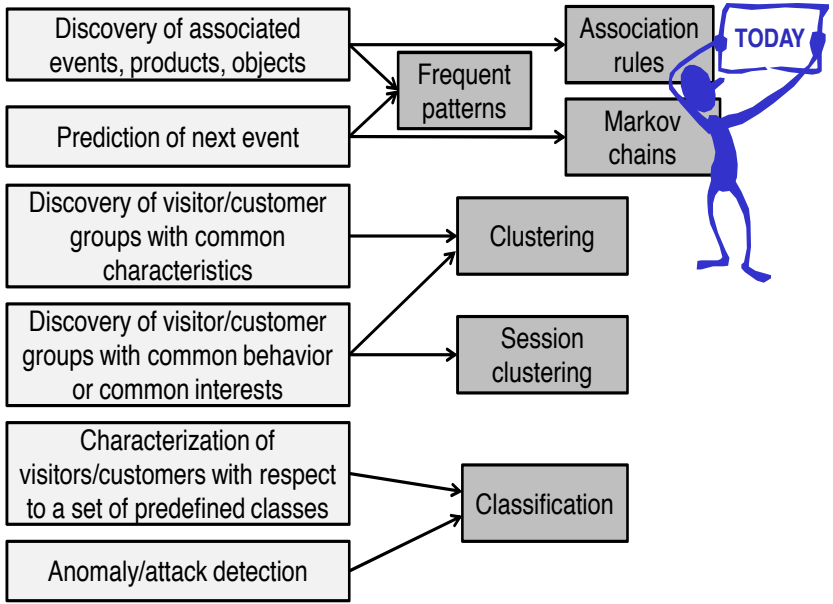
A → C, C → B, B → D, D → E, C → F

Need to look for the shortest backwards path from E to C based on the site topology. The elements of the path need to have occurred in the user trail previously.

E → D, D → B, B → C

[17] Ostatni etap to uzupełnianie ścieżek. Weźmy sesję wykryta przez href dla User 3. Przejście pomiędzy E oraz F jest nierealistyczne. Nie potwierdza go ani adres strony odsyłającej ani struktura sieci (brak linka między E oraz F). Najkrótszą ścieżką składającą się z ruchów wstecz jest przejście od E do D, od D do B i dopiero z B do C. Dodane elementy, czyli D oraz B były uprzednio żądane w ramach analizowanej sesji.

Web Usage Mining



[18] Metody eksploracji danych związane z analizą użytkowania sieci można podzielić na różne grupy. Algorytmy grupowania mogą posłużyć choćby do identyfikacji grup użytkowników o podobnych profilach, zachowaniach lub zainteresowaniach. Klasyfikacja może odnosić się zarówno do użytkowników, jak i wygenerowanych przez nich zdarzeń i zawsze polega na przypisaniu analizowanych do obiektu do zbioru predefiniowanych klas. Klasycznym przykładem jest tu identyfikacja anomalii takich jak spam. Ten wykład poświęcony jest jednak odkrywaniu powiązań między zdarzeniami, produktami czy obiektami, a w szczególnym wypadku także predykcji następnego zdarzenia. Omówimy algorytm Apriori, który służy do odkrywania tzw. zbiorów częstych. Te ostatnie posłużą nam jako podstawa do indukcji tzn. reguł asocjacyjnych. Wspomnimy też krótko zagadnienie odkrywania częstych sekwencji, w których, w przeciwieństwie do zbiorów, kolejność żądań lub odwiedzin ma znaczenie. Wreszcie narzędziem matematycznym, które pozwoli na predykcję kolejnego zdarzenia będzie łańcuch Markova.

Market Basket Analysis (1) - Affinity Analysis

- Data analysis and mining techniques for discovering co-occurence relationships among activities performance by or recorded about specific individuals (groups)
- Goal: find associations among groups of items occuring in a transactional database
- Roots in analysis of point-of-sale data, as in supermarkets

- Input: list of purchases by customers over differet visits
- Output: what items purchased together?

[19] Rozpoczniemy od zagadnienia analizy koszyków (ang. market basket analysis; MBA). Pod tym hasłem rozumie się metody eksploracji danych, służące do wykrywania współwystępowania zdarzeń, produktów lub obiektów na podstawie analizy zachowań użytkowników. Celem wysokiego poziomu jest tu znalezienie powiązań między grupami obiektów na podstawie danych znajdujących się w transakcyjnej bazie danych. Korzenie tego typu analizy znajdują się analizie tradycyjnych danych zakupowych zbieranych w kontekście supermarketów. Dane wejściowe dla takiej analizy stanowią zbiory (koszyki) produktów kupowanych w czasie pojedynczych wizyt w sklepie przez klientów, a spodziewane wyniki powinny odpowiedzieć na pytanie: które produkty są (często) kupowane razem?

Market Basket Analysis (2)

- Provide the retailer with information to understand the purchase behavior of a buyer
- Understand the buyer's needs, rewrite the store's layout accordingly, develop **co-promotional programs**, capture new buyers
- Roots in analysis of point-of-sale data, as in supermarkets

- Customers purchase shampoo and conditioner together
- Male customers buy diapers and beer jointly

[20] Analiza taka jest bardzo wartościowa, bo pomaga sprzedawcom na zrozumienie zachowania klientów oraz ich potrzeb. To z kolei może prowadzić do dedykowanych akcji takich jak zmiana układu sklepu, próba pozyskania nowych klientów na podstawie analizy zachowania tych starych oraz projektowanie promocji. Przykłady oczywistych zbiorów produktów, które często są kupowane razem to szampon i odżywka albo samochodzik i niebieskie ubranka dziecięce. Najsławniejszy przykład, który podaje się w kontekście algorytmu Apriori dotyczy łącznego zakupu pieluch i piwa, który swego czasu został odkryty na podstawie analizy zakupów młodych ojców. Dało to pole do działania sklepom, które mogły obniżyć cenę piwa, a podwyższyć cenę pieluch, zarabiając tym samym więcej na koszyku rozważanym jako całość.

Market Basket Analysis (3) - Frequent Itemset

- Business use of MBA has increased since the introduction of electronic point of sale
- Amazon: cross-selling when recommending products based on the purchase histories

Frequent itemset

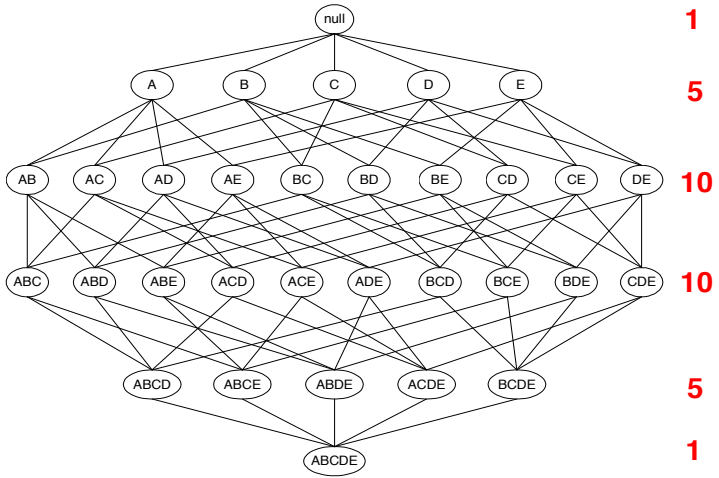
- Itemset: a set of one or more items
- Support: fraction of the baskets in which an itemset appears
- **The sets of items that have the minimum support**
- We focus on the pages that were viewed together in many sessions, but the idea is valid for other products and baskets
- Can be used for generation of **association rules**

[21] Analiza koszykowa znalazła pierwsze zastosowanie w tradycyjnych supermarketach, a potem w sklepach internetowych, gdzie dostępność danych o poszczególnych koszykach jest jeszcze większa. Wiemy, że z wyników tego typu analizy korzystał między innymi Amazon. Centralnym punktem zainteresowania jest tu chęć odkrycia tzw. zbiorów częstych. Wyjaśnijmy części składowe tego pojęcia. W ogólności zbiór to kolekcja jednego lub więcej obiektów. W zastosowaniach internetowych minimalna interesująca liczność to dwa. Aby odnieść się do częstości, musimy potrafić ją mierzyć. Odwołujemy się tu do pojęcia wsparcia (ang. suport), które definiuje się jako odsetek koszyków, w których dany zbiór produktów wystąpił. Aby mówić o zbiorze, że jest częsty, jego wsparcie musi spełniać pewien minimalny predefiniowany próg. W czasie wykładu skupimy się na analizie stron żądanych w wielu sesjach, ale omawiane pomysły są bardziej ogólne i mają zastosowanie w kontekście różnie interpretowanych koszyków i produktów. Co istotne, zbiory częste będą dla nas stanowiły punkt wyjścia dla generacji reguł asocjacyjnych.

Apriori Principle (1)

Given n items, there are 2^n possible itemsets

List all possible itemsets and compute their support



It would not work!

[22] Przedmiotem naszego zainteresowania przez najbliższe kilka slajdów będzie sławny algorytm Apriori, który służy do generacji zbiorów częstych. Rozpocznijmy jego omówienie od analizy wyczerpującego przeszukiwania wszystkich możliwych podzbiorów. Jeśli analizowanych jest n obiektów, to takich podzbiorów jest 2 do n-tej. Już w przypadku n równego 5 taki naiwny algorytm musiałby rozważyć 32 podzbiory. Na slajdzie przedstawiono ich rozpiskę w kontekście sesji odwołujących się do stron A, B, C, D i E. Już ten mały przykład uzmysławia, że analiza naiwna byłaby zbyt kosztowna.

Apriori Principle (2)

Support is "**downward closed**"

- If an itemset is frequent (has enough support), then all of its subsets must also be frequent
- This is due to the ***anti-monotone*** property of support

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

if {AB} is frequent,

⇩

both {A} and {B} are frequent

Corollary: if an itemset doesn't satisfy minimum support, none of its supersets will either

- Essential for pruning search space

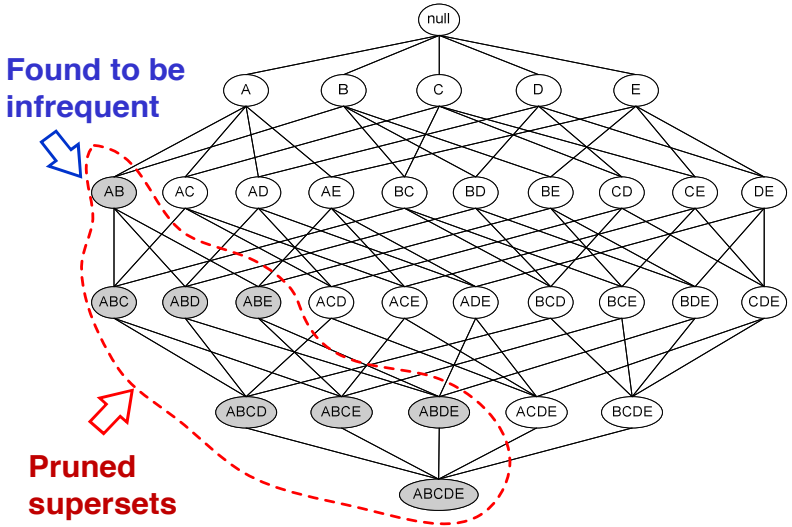
if {A} is not frequent,

⇩

{AB} is not frequent

[23] Szczęśliwie z pomocą w ograniczeniu złożoności przychodzą własności zbiorów częstych. Pierwszą jest własność antymonotoniczności wsparcia. Jeśli zbiór X jest podzbiorem Y, to wsparcie dla X musi być co najmniej takie jak wsparcie dla Y. Konsekwencją tego jest fakt, że jeśli jakiś zbiór nie spełnia minimalnego progu wsparcia, to jego nadzbiory także nie będą go spełniały. Jest to kluczowe w kontekście eliminacji podzbiorów, które trzeba w algorytmie rozważać. Z drugiej strony, wsparcie charakteryzuje się własnością domknięcia w dół. Jeśli jakiś zbiór uznamy za częsty, bo osiągnie minimalny wymagany próg, to wszystkie jego podzbiory też muszą być częste. Podsumowując na przykładach, jeśli zbiór AB jest częsty, to także A oraz B muszą być częste. Jeśli z kolei A nie jest częsty, to także zbiór AB nie będzie.

Apriori Principle (3)

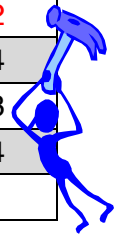


[24] Własności omówione na poprzednim slajdzie są kluczowe dla algorytmu Apriori, a w szczególności dla realizacji odcięć w przestrzeni wszystkich podzbiorów, których wsparcie trzeba sprawdzić. Odwołując się do przykładu zaprezentowanego na slajdzie, jeśli na jakimś etapie trwania algorytmu dowiedzielibyśmy się, że zbiór AB nie jest częsty, to także wszystkie jego podzbiory wynikające z dodania w różnych kombinacjach C, D oraz E, nie byłyby częste. Sumarycznie można by pominąć liczenie wsparcia dla 7 innych podzbiorów. Oszczędność jest zatem spora.

Apriori Principle (4)

Items (1-itemsets)

Page	Count
A	4
B	2
C	4
D	3
E	4
F	1



No need to generate candidates involving B and F

Pairs (2-itemsets)

Pages	Count
{A,C}	3
{A,D}	2
{A,E}	3
{C,D}	2
{C,E}	3
{D,E}	3

No need to generate candidates involving {A,D} and {C,D}

Triplets (3-itemsets)

Pages	Count
{A,C,E}	3

minsup = 3/5

assuming there are 5 sessions

[25] Przed sformułowaniem kroków algorytmu Apriori rozważmy jeszcze prosty przykład, w którym analizowane są wizyty na sześciu stronach od A do F. Chcemy odkryć zbiory częste jedno-, dwu- i trójelementowe, które spełniają minimalny próg wsparcia 3/5 w kontekście 5 sesji. W tabeli po lewej stronie przedstawiono wsparcie dla zbiorów jednoelementowych. Zarówno B, jak i F nie spełniają minimalnego progu, stąd nie ma sensu generować ich nadzbiorów jako kandydatów na zbiory częste. Rozważając A, C, D oraz E, jesteśmy w stanie stworzyć sześciu kandydatów na zbiory częste dwuelementowe. Spośród nich cztery podzbiory spełniają minimalny próg, a dwa inne, AD oraz CD, tego progu nie spełniają. Analiza zbiorów częstych dwuelementowych prowadzi do możliwości utworzenia tylko jednego kandydata trójelementowego ACE. On także okazuje się częsty.

Apriori Algorithm (1)

- Assume all sessions/transactions are internally ordered (e.g., lexicographically)
- L_k : frequent itemset of size k
- C_k : candidate itemset of size k

L_1 = {frequent items}
for (k=1, L_k != empty set, k++) **do**
  begin
    C_{k+1} = **generate candidates from** L_k;
    for each session t do
      increment the count of all candidates in C_{k+1}
      that are contained in t;
    L_{k+1} = candidates in C_{k+1} with **min_support**;
  end
return $\cup_k L_k$

[26] Algorytm Apriori jest jednym z najsławniejszych algorytmów w dziedzinie analizy danych. Załóżmy, że wszystkie sesje lub transakcje są uporządkowane, na przykład leksykograficznie. Przez Ck oznaczymy kandydatów na zbiory częste k-elementowe, a przez Lk - już same zbiory częste o liczności k, a więc kandydatów spełniających minimalny próg. Algorytm rozpoczyna się od generacji zbiorów częstych jednoelementowych. Następnie w pętli, iterując od k równego 1, generujemy kandydatów na zbiory częste k+1 elementowe tylko i wyłącznie na podstawie zbiorów częstych k-elementowych. Dla każdego z kandydatów obliczane jest wsparcie, a jako zbiory częste uznawani są ci kandydaci, którzy spełniają minimalny próg wsparcia. Procedura kontynuowana jest dopóki jakiś zbiór Lk zbiorów częstych k-elementowych okaże się pusty. Jako wynik działania algorytmy zwracane są wszystkie zbiory częste o różnych licznościach.

Apriori Algorithm (2)

```
function C_{k+1} = generate candidates from L_k;
    insert into C_{k+1}
    select  p.item_1, p.item_2, …, p.item_k, q.item_k,
    from L_k p, L_k q,
    where p.item_1 = q.item_1, p.item_2 = q.item_2, …,
          p.item_{k-1} = q.item_{k-1}, p.item_k < q.item_k,

    for all itemsets c in C_{k+1} do
        for all k-subsets s in c do
            if s is not in L_k then delete c from C_{k+1}
endfunction
```

Join step: C_{k+1} is generated by joining L_k with itself

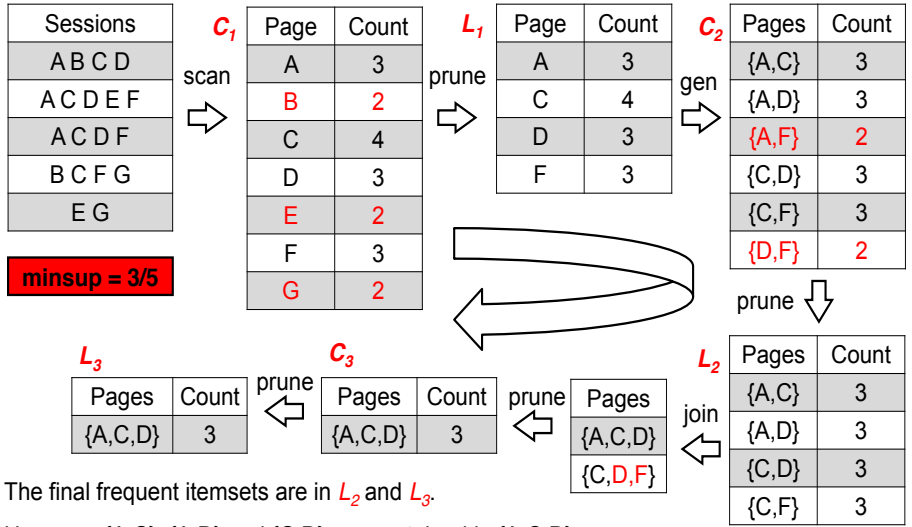
- *abcd* from *abc* and *abd*
- *acde* from *acd* and *ace*

Prune step: any k-itemset that is not frequent cannot be a subset of a frequent k+1-itemset

- *acde* is removed, because *ade* is not frequent

[27] Kluczowy dla działania algorytmu jest sposób generowania kandydatów na zbiory często k+1 elementowe. Procedura ta operuje na uporządkowanych zbiorach częstych k elementowych. W kroku łączenia rozważane są te zbiory, które mają dokładnie takie same k-1 elementów i różnią się tylko elementem ostatnim. W ten sposób powstają kandydaci k+1 elementowi. Przykładowo, jeśli rozważyć zbiory abc oraz abd, to mają one 2 wspólne elementy, ab, i różnią się ostatnim. Z ich połączenia powstaje zbiór abcd. W kroku przycinania rozważane są wszystkie podzbiory k-elementowe takiego roboczego kandydata. Jeśli jakiś z nich nie okaże się częsty, to taki kandydat nie jest rozważany dalej. Przykładowo, jeśli w analizie kandydata acde (powstałego z połączenia acd oraz ace) okaże się, że podzbiór ade nie jest częsty, to acde nie jest dalej rozważane.

Apriori Algorithm (3) - Illustrative Example



The final frequent itemsets are in L_2 and L_3.

However, {A,C}, {A,D} and {C,D} are contained in {A,C,D}.
Thus, the final group of itemsets reported by Apriori are {A,C,D} and {C,F}.

[28] Przykład ilustrujący działanie algorytmu Apriori dotyczy analizy 5 sesji przy minimalnym progu wsparcia 3/5. Zaczynamy od rozważenia zbiorów jednoelementowych. Przykładowo, A pojawia się w 3 sesjach, a B w 2. Ostatecznie tylko 4 z 7 zbiorów jednoelementowych spełniają minimalny próg. W kolejnym kroku są one wykorzystywane do generacji 6 kandydatów. Spośród nich zbiór AC pojawia się w 3 sesjach, a DF tylko w 2. Czterech kandydatów spełnia minimalny próg 3/5. Następny krok polega na połączeniu AC oraz AD, a także CD oraz CF w kandydatów trójelementowych. Kandydat CDF jest od razu odrzucany, bo podzbiór DF sam nie jest częsty. Ostatecznie kandydat ACD okazuje się częsty, ale nie da się już wygenerować kandydatów bardziej licznych. Interesujące z punktu widzenia dalszej analizy są zbiory częste 2- oraz 3-elementowe, a więc AC, AD, CD, CF, a także ACD. Można jednak zauważyć, że AC, AD oraz CD są podzbiorami ACD, stąd można ograniczyć się do rozważenia zbiorów CF oraz ACD.

Apriori Algorithm (4) - Applications

Term Associations

- Find associations among words based on their occurrences in documents
- Words correspond to items and documents correspond to baskets
- Brad and Angelina

	Doc 1	Doc 2	Doc 3	…	Doc n
business	5	5	2	…	1
capital	2	4	3	…	5
…	…	…	…	…	…
Invest	6	0	0	…	3

Document Associations

- Find (content-based) associations among documents in a collection
- Documents correspond to items and words/sentences correspond to baskets
- Frequent itemsets are groups of docs in which many words occur in common (in an extreme case – plagiarism)

[29] Największe piękno algorytmów takich jak PageRank i Apriori wynika z mnogości ich potencjalnych zastosowań. Rozważmy dwa inne przykłady, w których koszyki i produkty będą inaczej interpretowane. Jeśli uznać, że produkty to termy, a koszyki dokumenty, to zbiory częste mogą wskazywać na powiązania między termami, które pojawiają się razem w wielu dokumentach. Analiza taka jest często ograniczana do dokumentów ostatnio dodanych, a najsławniejszym przykładem takiej asocjacji termów byli Brad i Angelina; gdy wiele lat temu zaczęli się mieć ku sobie, różne serwisy szybko zaczęły o tym pisały. Jeśli jednak, w innym przykładzie, uznać, że produkty to dokumenty, a koszyki termy lub zdania, to zbiory częste wskazują na dokumenty powiązane tematycznie, bo używające tych samych pojęć. W szczególnymi wypadku może to oznaczać plagiaty. Przykład ten jest dosyć nietypowy, bo produkty są tu większe niż koszyki, co kłóci się z oryginalną interpretacją analizy koszykowej dla zakupów klientów w supermarketach, gdzie produkt wkłada się do koszyku.

Association Rules (1) - Form

Typical **rule** form:

premise (X) ⇨ conclusion (Y)

- Premise and conclusion can be represented as sets of items
- A **rule** is an implication among **itemsets** ***X*** and ***Y***, of the form **X → Y**, where **X, *Y*** in ***I***, and **X ∩ Y = empty set**

Example: **{A,B} → {C,D,E}**

- if a client purchased products A and B,
  then **it is likely** that (s)he also bought products C, D and E
- if a user viewed pages A and B,
  then **it is likely** that (s)he also viewed pages C, D and E

[30] Zbiory częste będą stanowiły podstawę do dalszej analizy z wykorzystaniem reguł asocjacyjnych. Reguła taka ma postać wyrażenia logicznego, w którym z przesłanki wynika konkluzja. W przypadku reguł asocjacyjnych zarówno w części warunkowej, jak i decyzyjnej pojawiają się zbiory produktów, przy czym muszą być to zbiory rozłączne. Z obecności zbioru X ma wynikać (bardzo prawdopodobna) obecność zbioru Y. Przykładowo, jeśli klient kupił produkty A i B, to prawdopodobne jest też że kupi produkty C, D i E lub z innej perspektywy, jeśli odwiedził strony A i B, to prawdopodobne jest że odwiedzi też strony C, D i E.

Association Rules (2) - Basic Concepts (Support)

Association rule **X → Y**, where **X** and **Y** are non-overlapping itemsets

- Metrics that indicate the strength and importance of rule: support *sup*(**X → Y**) and confidence *conf*(**X → Y**)

Support (*sup*) support(X → Y) = support(X∪Y) = σ(X∪Y) / |D|

- Fraction of transactions/sessions that contain both **X** and **Y**
- Probability that a transaction contains **{X∪Y}** or **Pr(X∧Y)**

Sessions
A B
A C D E
B C D F
A **B C D**
A B C F

Rule **{B,C} → D**

$$sup = \frac{\sigma(B,C,D)}{|sessions|} = \frac{2}{5}$$

2 sessions containing B, C and D

5 sessions overall

[31] Z praktycznego punktu widzenia interesujące są tylko te reguły, które są silne. Miar takiej siły istnieje bardzo wiele, ale my skupimy się na dwóch, wsparciu oraz ufności. Wsparcie dla reguły jeżeli X to Y można obliczyć jako wsparcie dla sumy zbiorów X i Y, a więc częstość występowania X i Y we wszystkich sesjach/transakcjach. Przykładowo, dla reguły: jeśli B i C, to D, wsparcie wynosi 2/5, bo wśród 5 sesji, zbiór BCD występuje 2 razy. Można ten współczynnik interpretować także jako prawdopodobieństwo, że sesja/transakcja zawiera zbiory X i Y.

Association Rules (3) - Basic Concepts (Confidence)

Confidence (*conf*)

- confidence(X → Y) = support(X ∪ Y) / support(X) = σ(X ∪ Y) / σ(X)
- Measures how often items in **Y** appear in transactions that contain **X**
- ***Conditional probability*** that a transaction will contain **Y** given that it contains X **or Pr(Y|X)**

Sessions
A B
A C D E
B C D F
A **B C D**
A B C F

Rule **{B,C} → D**

$$conf = \frac{\sigma(B,C,D)}{\sigma(B,C)} = \frac{2}{3}$$

2 sessions containing B, C and D

3 sessions containing only B and C

[32] Dla reguły: jeżeli X, to Y, ufność odwołuje się do tego, jak często Y pojawia się w transakcjach, w których jest X. Miara ta odzwierciedla więc prawdopodobieństwo warunkowe, które ujmuje na ile często w transakcjach, które zawierają obiekty z części warunkowej pojawią się też obiekty z części decyzyjnej. Obliczenie ufności bazuje na ilorazie wsparcia sumy zbiorów X i Y oraz wsparcia zbioru X (z części warunkowej). Przykładowo, dla reguły: jeśli B i C, to D, ufność wynosi 2/3, bo wsparcie dla zbioru BCD to 2/5, a dla BC 3/5. Wynik ten można interpretować tak, że na 3 razy gdy w sesjach pojawiają się strony B i C, 2 razy występuje tam też D.

Association Rules (4) - Algorithm

- Only strong association rules are interesting
- Strong rule satisfy minimum support (*minsup*) and minimum confidence (*minconf*) thresholds

$sup(X \rightarrow Y) \geq minsup$

$conf(X \rightarrow Y) \geq minconf$

- Frequent itemsets satisfy minimum support threshold *minsup*

for each frequent itemset **F** do
  generate all non-empty subsets of **F**
for each non-empty subset **X** of **F** **do**
  if support(**F**)/support(**X**) ≥ *minconf* **then**
    output rule **X** → **Y**, where Y = F/X
end

$sup(X \rightarrow Y) = sup(F)$

$conf(X \rightarrow Y) = sup(F)/sup(X)$

[33] Silne reguły asocjacyjne to takie, które spełniają minimalny próg wsparcia i minimalny próg ufności. Ich generacji bazuje na zbiorach częstych. Dlaczego? Ano dlatego, że zbiór częsty F z definicji spełnia minimalny próg wsparcia. Jeśli więc podzielić elementy F na część warunkową i decyzyjną, to wsparcie takiej reguły też będzie spełniało minimalny próg. Nazwijmy te elementy przez X (w części warunkowej) oraz Y = F minus X (w części decyzyjnej). Jeśli ufność reguły: jeżeli X to Y, obliczona jako iloraz wsparcia F do wsparcia X, spełnia minimalny próg, to regułę można uznać za interesującą. Takie rozważania powtarzane są dla wszystkich zbiorów częstych i ich wszystkich możliwych podziałów.

Association Rules (5) - Example

Sessions
A B C D
A C D E F
A C D F
B C F G
E G

L_1

Page	Count
A	3
C	4
D	3
F	3

L_2

Pages	Count
{A,C}	3
{A,D}	3
{C,D}	3
{C,F}	3

L_3

Pages	Count
{A,C,D}	3

minsup = 3/5

minconf = 1

Candidate **rules** for {C,F}		Candidate **rules** for {A,C,D}			
Rule	Conf.	Rule	Conf.	Rule	Conf.
{C}→{F}	3/4	**{A,C}→{D}**	**3/3**	**{A}→{C}**	**3/3**
{F}→{C}	**3/3**	**{A,D}→{C}**	**3/3**	**{A}→{D}**	**3/3**
		{C,D}→{A}	**3/3**	{C}→{A}	3/4
		{A}→{C,D}	**3/3**	{C}→{D}	3/4
		{C}→{A,D}	3/4	**{D}→{A}**	**3/3**
		{D}→{A,C}	**3/3**	**{D}→{C}**	**3/3**

[34] Rozważmy wcześniej analizowany przykład. Celem jest teraz generacja reguł o minimalnym wsparciu 3/5 oraz minimalnej ufności 1. Aby mieć gwarancję minimalnego wsparcia reguły wygenerujemy ze zbiorów częstych. Wcześniej uzyskaliśmy dwa takie zbiory, CF oraz ACD. Ze zbioru CF, jesteśmy w stanie utworzyć dwie reguły: jeżeli C to F oraz jeżeli F to C. Tylko ta druga spełnia minimalny próg ufności. Ze zbiory ACD możliwych reguł jest dużo więcej (12). Co ważne, generujemy je ze zbiorów 2- i 3-elementowych. Reguły spełniające minimalny próg ufności pogrubiono na slajdzie. Oczywiście postać reguł może być też narzucona z góry. Przykładowo, jeśli interesujące reguły miałyby mieć formę: jeżeli dwa obiekty to jeden obiekt, to rozważalibyśmy tylko zbiory częste trójelementowe i każdy ich podzbiór dwuelementowy znalazłby się w części warunkowej kandydackiej reguły asocjacyjnej.

Association Rules (6) - Applications

Examples

- 60% of clients who accessed **/products/**, also accessed **/products/software/webminer.htm**
- 30% of clients who accessed **/special-offer.html**, placed an online order in **/products/software/**
- Actual example from IBM official Olympics Site: {Badminton, Diving} → {Table Tennis}

Applications

- Use rules to **serve dynamic, customized contents to users**
- **Prefetch files** that are most likely to be accessed
- Determine the best way to structure the web site (**site optimization**)
- Targeted **electronic advertising** and increasing cross sales

[35] Wykorzystanie reguł asocjacyjnych w praktyce jest bardzo rozległe. Podstawowy przykład dotyczy użytkowników serwisów internetowych lub klientów sklepów online, dla których jesteśmy w stanie stwierdzić, że jeśli odwiedzili jakieś strony lub kupili jakieś produkty, to są szanse, że będą też zainteresowani konkretnymi innymi stronami i produktami. Taką wiedzę można z kolei wykorzystać do dynamicznej prezentacji potencjalnych interesujących treści, stron lub produktów, szybszej obsługi kolejnych ruchów użytkownika, optymalizacji struktury serwisu lub sklepu i wreszcie do wypracowania dedykowanych reklam, które pomogą nam zwiększyć sprzedaż.

Association Rules (7) - Applications

Discover affinities among sets of web page references across user sessions

[36] Na slajdzie przedstawiono najsławniejszy przykład wykorzystania reguł asocjacyjnych w serwisie Amazon. Dla danego produktu, filmu czy książki, na dole strony przedstawione są produkty, które kupowane były łącznie z nim przez wielu klientów, którzy wcześniej dokonali transakcji. Takie rekomendacje produktów można spotkać w wielu serwisach sprzedażowych. Wystarczy wejść do sklepu online adidas lub showroom, by się o tym przekonać.

Sequential Pattern Mining (1)

- Association rule mining **does not consider the order of transactions**
- In many applications such orderings are significant:
  - in market basket analysis, it is interesting to know whether people buy some items in sequence
  - e.g., buying bed first and then bed sheets some time later

- In web usage mining, it is useful to find **navigational patterns** of users in a web site from **sequences of page visits of users**

[37] Zbiory częste i reguły asocjacyjne, które na nich bazują nie biorą pod uwagę kolejności odwiedzania stron w ramach sesji. W wielu zastosowaniach porządek ma jednak znaczenie. Nawet w tradycyjnej analizie koszykowej kolejność zakupów często odgrywa istotną rolę. Przykładowo, dobrze jest wiedzieć, że najpierw kupowane jest łóżko, a potem pościel, najpierw samochód, a potem ubezpieczenie, najpierw telefon, a potem jego pancerna obudowa. W eksploracji użytkowania sieci, przydatne jest więc odkrywanie wzorców nawigacyjnych dla użytkowników korzystających z serwisu na podstawie sekwencji odwiedzania przez nich stron, uwzględniających kolejność, a nie tylko fakt współwystępowania.

Sequential Pattern Mining (2)

- Sequential patterns add **an extra dimension** to frequent itemsets and association rules - **time**
- Items can appear before, after, or at the same time as each other
- General form: “x% of the time, when A appears in a transaction, B appears (within *z* transactions)”
- Other items may appear between A and B, so sequential patterns do not necessarily imply consecutive appearances of items (in terms of time)

Examples

- Renting “Star Wars”, then “Empire Strikes Back”, then “Return of the Jedi” in that order
- Collection of ordered events within an interval

[38] W porównaniu do zbiorów częstych, sekwencje uwzględniają więc dodatkowo wymiar czasu. Istotne jest, że niektóre obiekty pojawiają się przed, po lub w tym samym czasie co inne. Interpretacja takich wzorców sekwencyjnych jest następująca: x% razy, gdy w transakcji pojawia się najpierw A, później pojawi się w niej także B. To następstwo nie musi być wcale bezpośrednie; pomiędzy A i B może pojawić się wiele innych obiektów. Intuicyjny przykład może dotyczyć dowolnej kolekcji zdarzeń zachodzących w określonych przedziale czasowym takich jak obejrzenie poszczególnych części Gwiezdnych Wojen w określonej kolejności (podanej na slajdzie).

Sequential Pattern Mining (3)

- Given a set *S* of input data sequences (or sequence database), the problem of mining sequential patterns is to **find all the sequences that have a user-specified minimum support**
- Most sequential pattern discovery algorithms are based on **extensions of the Apriori algorithm** for discovering itemsets
- Each such sequence is called a **frequent sequence**, or a **sequential pattern**

A *sequence database*

ID	Sequence
1	<A(**AB**C)(A**C**)D(CF)>
2	<(AD)C(BC)(AE)>
3	<(EF)(**AB**)(DF)**C**B>
4	<EG(AF)CBC>

first A, second (ABC), third (AC), fourth D, fifth (CF)

if *support threshold* ***minsup* = 2/4**, **<(AB)C>** is a *sequential pattern*

[39] Zagadnienie eksploracji wzorców sekwencyjnych zostanie w czasie tego wykładu tylko wspomniane. Niech będzie danych S sekwencji wejściowych. Na slajdzie jest ich 4; każdą z nich można interpretować jako kolejne sesje pojedynczego użytkownika. Zadanie polega tu na znalezieniu wzorców sekwencji lub równoważnie częstych sekwencji, które spełniają minimalny próg wsparcia. Przykładowo, wzorzec sekwencji zaprezentowany na slajdzie wskazuje, że jeśli pojawi się najpierw A i B, to potem pojawia się też C. Jest to wzorzec ze wsparciem 2/4, bo 2 z 4 sekwencji w bazie danych potwierdzają takie niekoniecznie bezpośrednie następstwo. Większość algorytmów służących do generacji takich częstych sekwencji, jak GSP (Generalized Sequential Pattern) opiera się na rozszerzeniach algorytmu Apriori.

Sequential Pattern Mining (4) - Navigational Patterns

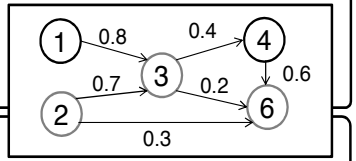
- Can be viewed as a special form of sequential patterns which capture navigational patterns among users of a site
  - A **session** is treated as a **consecutive sequence of pageview references** for a user over a specified period of time
  - Each session induces a user trail through the site
- A trail is a sequence of web pages followed by a user during a session, ordered by time of access
- A **sequential pattern** in this context is a **frequent trail**

- Sequential pattern mining can **help identify common navigational sequences** which in turn helps in understanding common user behavioral patterns
- Underlying machinery for **link prediction** and for **web prefetching**

[40] Wzorce nawigacyjne są specjalnym rodzaje wzorców sekwencyjnych, które ujmują pewne schematy zachowań grupy użytkowników. Wzorce takie rozważa się więc w bardzo konkretnym zastosowaniu, gdzie sesja użytkownika jest traktowano jako sekwencja stron odwiedzonych w stosunkowo krótkim czasie, de facto wyznaczając szlak lub ślad, po którym porusza się użytkownik. Wzorzec sekwencyjny ma tu więc postać często powtarzającego się śladu. Sugeruje to podstawowe zastosowanie wzorców nawigacyjnych do znalezienia ścieżek pokonywanych przez wielu użytkowników i zrozumienia często powtarzającego się zachowania. To z kolei jest przydatne choćby przy predykcji kolejnych ruchów użytkowników na podstawie prawdopodobieństwa ich wystąpienia pod warunkiem, że wcześniej zrobili oni to i to, oraz bardziej efektywnej obsłudze takich potencjalnych zachowań.

Navigational Patterns (1) - Markov Chain

- If the goal is to make predictions about future user actions based on past behaviour, approaches such as *Markov models* (e.g., **Markov Chains**) can be used
- Model the navigational sequences through the site as a **state-transition diagram**



- **Markov Chain** consists of:
  - a set of states (pages or pageviews in the site): $S = \{s_1, s_2, \ldots, s_n\}$
  - a set of transition probabilities (likelihood that a user will navigate from one state to another)

    $P = \{p_{1,1}, \ldots, p_{1,n}, p_{2,1}, \ldots, p_{2,n}, \ldots, p_{n,1}, \ldots, p_{n,n}\}$
  - "memorylessness" - the probability of each event depends only on the state attained in the previous event

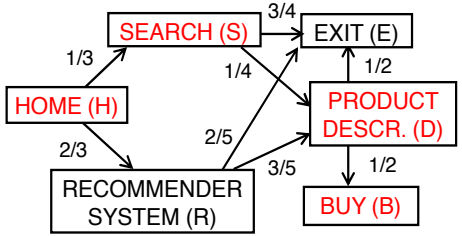
[41] W kontekstach, gdy celem jest wykorzystanie przeszłych zachowań do predykcji tych przyszłych bardzo przydatne okazują się modele Markova. My wykorzystamy łańcuch Markova do reprezentacji sekwencji nawigacyjnych w postaci diagramu przejść między stanami. Taki łańcuch składa się ze zbioru stanów, które w naszym przypadku reprezentują strony, a właściwie ich wyświetlenia oraz zbioru prawdopodobieństw przejść, które dla danej pary stanów określają prawdopodobieństwo przejścia od stanu poprzedniego do następnego. Dany stan jest osiągalny z innego, jeśli prawdopodobieństwo takiego przejście jest dodatnie. Z kolei stany nazywamy skomunikowanymi, jeżeli są wzajemnie osiągalne. Łańcuch Markova jest układem bez pamięci, co oznacza, że prawdopodobieństwo każdego zdarzenia zależy jedynie od stanu poprzedniego. Jest to uproszczenie, ale w kontekście analizy sesji użytkowników w dużej mierze uzasadnione, bo użytkownik znajdujący się na danej stronie ma dostęp do wszystkich opcji, która ona oferuje niezależnie od tego, czy jest to dla niego pierwsza czy dziesiąta strona w sesji.

Navigational Patterns (2) - Markov Chain

- A path r from a state s_i to a state s_j, is a sequence states where the transition probabilities for all consecutive states are greater than 0
- The probability of reaching a state s_j from a state s_i via a path r is the product of all probabilities along the path: $p(r) = \prod_{K} p_{k,k+1}$
- The probability of reaching s_j from s_i is the sum over all paths:

$$p(s_j|s_i) = \sum_{r \in R} p(r)$$

What is the probability that a user who visits Home purchases a product?



p(H → S → D → B) =
= 1/3 · 1/4 · 1/2 = 1/24 = 0.042

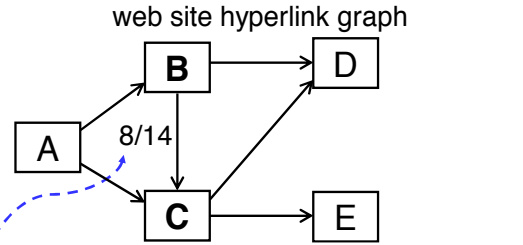
p(H → R → D → B) =
= 2/3 · 3/5 · 1/2 = 1/20 = 0.050

p(B|H) = 0.042 + 0.050 = 0.092

[42] Ścieżką ze stanu i do stanu j nazywamy sekwencje stanów, które prowadzą od i do j poprzez przejścia z niezerowymi prawdopodobieństwami. Prawdopodobieństwo pokonania takiej ścieżki oblicza się jako iloczyn prawdopodobieństw wszystkich ruchów/przejść składowych. Wreszcie prawdopodobieństwo osiągnięcia stanu j ze stanu i jest obliczane jako suma prawdopodobieństw pokonania wszystkich możliwych ścieżek prowadzących od i do j. Taka analiza jest najbardziej interesująca, gdy strony mają ściśle przypisane role jak na przykładzie przedstawionym na slajdzie, gdzie mamy choćby stronę domową, stronę związaną z rekomendacją, włożeniem produktu do koszyka, zakupem lub wyjściem z serwisu. Aby obliczyć prawdopodobieństwo realizacji zakupu (strona B) pod warunkiem, że użytkownik znalazł się wcześniej na stronie domowej H, trzeba rozważyć dwie sekwencje: HSDB oraz HRDB, obliczyć ich prawdopodobieństwa jako iloczyny prawdopodobieństw realizacji poszczególnych ruchów składowych, a na samym końcu zsumować.

Navigational Patterns (3) - Markov Chain

Sessions:
A, B
A, B
A, B, C
A, B, C
A, B, C, D
A, B, C, E
A, C, E
A, C, E
A, B, D
A, B, D
A, B, D, E
B, C
B, C
B, C, D
B, C, E
B, D, E



the transition probabilities are obtained from counting click-throughs

Transition B → C:
Total occurrences of B: **14**
Total occurrence of BC: **8**
p(C|B) = **8/14**

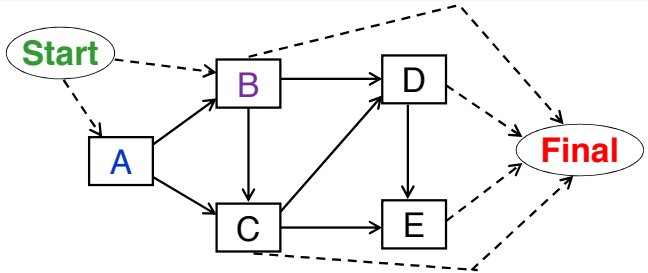
[43] Przykład odnosi się do konstrukcji łańcucha Markova w kontekście wzorców nawigacyjnych. Analizowany jest zbiór 16 sesji, w czasie których łącznie odwiedzonych jest 5 stron, A-E. Strony odpowiadają stanom łańcucha, a prawdopodobieństwa oblicza się na podstawie analizy bezpośredniego następstwa stron w sesjach. Przykładowo, prawdopodobieństwo przejścia od B do C to 8/14, bo B pojawia się w 14 sesjach, ale tylko w 8 jego bezpośrednim następnikiem jest C. Na podobnej zasadzie prawdopodobieństwo przejścia od A do B wynosi 9/11, bo A jest w 11 sesjach, w których B pojawia się zaraz po A 9 razy.

Navigational Patterns (4) - Markov Chain

Sessions:
A, B
A, B
A, B, C
A, B, C
A, B, C, D
A, B, C, E
A, C, E
A, C, E
A, B, D
A, B, D
A, B, D, E
B, C
B, C
B, C, D
B, C, E
B, D, E

Add a **unique start state**

- the start state has a transition to the first page in each session (representing the start of a session)



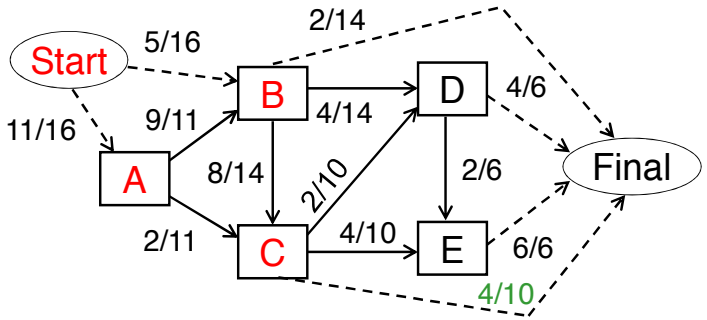
Add a **unique final state**

- the last page in each trail has a transition to the final state (representing the end of the session)
- the Markov chain built is called *absorbing* since we always end up in the final state

[44] Należy odróżnić łańcuch Markova od struktury połączeń między stronami w serwisie. W łańcuchu reprezentowane są tylko przejścia, dla których prawdopodobieństwa są dodatnie. Link może więc istnieć, ale jeśli nie był wykorzystany przez żadnego użytkownika, to takiego przejścia w łańcuchu nie będzie (patrz B do E). Na podobnej zasadzie linku może nie być (patrz D do E), a przejście będzie reprezentowane w łańcuchu o ile zaobserwowany jest w co najmniej jednej sesji. Dodatkowo, w łańcuchu reprezentuje się wirtualne stany, początkowy i finalny. Ten pierwszy umożliwia zamodelowanie przejść do stron inicjujących sesje, jak A czy B. Ten drugi pozwala na reprezentowanie stanów kończących sesje jak B, C, D czy E. Skonstruowany łańcuch Markova nazywamy absorbujących, gdyż zawsze kończymy w stanie finalnym. Na slajdzie przedstawiono stany, ale zupełna jego definicja wymaga też podania prawdopodobieństw.

Navigational Patterns (5) - Example

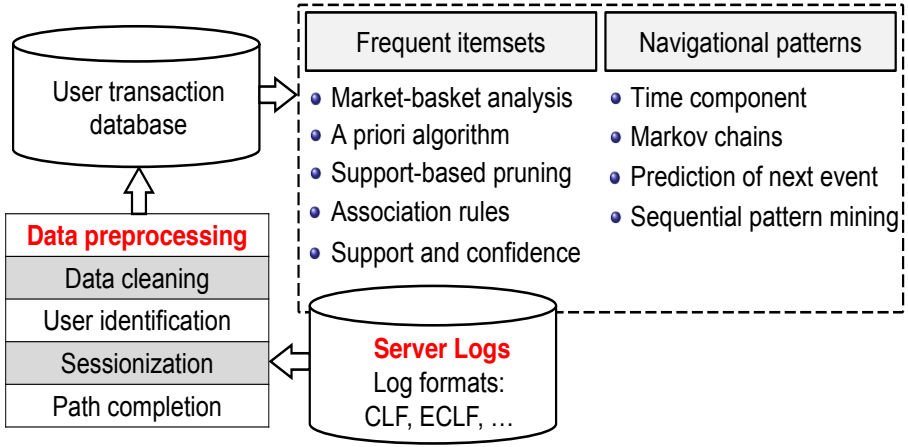
Sessions:
A, B
A, B
A, B, C
A, B, C
A, B, C, D
A, B, C, E
A, C, E
A, C, E
A, B, D
A, B, D
A, B, D, E
B, C
B, C
B, C, D
B, C, E
B, D, E



- Probability that someone will visit page C?
  S→B→C + S→A→C + S→A→B→C
  (5/16 * 8/14) + (11/16 * 2/11) + (11/16 * 9/11 * 8/14) = 0.503
- Probability that someone who has visited B will visit E?
  B→D→E + B→C→E + B→C→D→E
  (4/14 * 2/6) + (8/14 * 4/10) + (8/14 * 2/10 * 2/6) = 0.335
- Probability that someone visiting page C will leave the site? 4/10

[45] Pełny łańcuch Markova przedstawiono na slajdzie. Pozwala on na uzyskanie w łatwy sposób odpowiedzi na wiele interesujących pytań. Przykładowo, aby obliczyć prawdopodobieństwo odwiedzin strony C, należy rozważyć wszystkie ścieżki od stanu początkowego S do C, obliczyć ich prawdopodobieństwa i posumować. Na slajdzie zaprezentowano też sposób obliczenia prawdopodobieństwa odwiedzin strony E, o ile ktoś był wcześniej na stronie B oraz prawdopodobieństwo opuszczenia serwisu po wizycie na stronie C. Gdy mówimy tylko o stronach oznaczonych symbolami A-E, to ciężko sobie wyobrazić przydatność takiej analizy. Jeśli natomiast pomyśleć, że te strony pełnią różne role choćby w sklepie internetowym (domowa, promocja, koszyk, zakup, itd.), to praktyczna użyteczność jest już oczywista.

Summary (1)



[46] Wykład stanowił wstęp do analizy użytkowania sieci. W pierwszej kolejności skupiliśmy na źródłach danych, które można by wykorzystać do eksploracji użytkowania. Szczególną uwagę przyłożyliśmy do plików log serwera oraz ich formatów. Omówiliśmy też etapy wstępnego przetwarzania danych, w tym czyszczenie, identyfikację użytkowników i sesji oraz uzupełnianie ścieżek. Kroki te prowadzą do uzyskania bazy transakcyjnej, które stanowi podstawę dla właściwej analizy użytkowania. Głównym przedmiotem zainteresowania w czasie wykładu była generacja zbiorów częstych. Wiecie już na czym polega analiza koszykowa, algorytm Apriori oraz generacja reguł asocjacyjnych. Znacie też własności zbiorów z punktu widzenia ich wsparcia i potraficie interpretować współczynniki wsparcia oraz ufności w kontekście opisu siły reguł. Do naszej analizy wprowadziliśmy też komponent czasowy. W tym względzie szczególnie interesowała nas generacja wzorców nawigacyjnych oraz wykorzystanie w ich kontekście łańcuchów Markova. Na trzech kolejnych slajdach znajdują się zadania przydatne do zrozumienia materiału przedstawionego na wykładzie oraz przygotowania do kolokwium zaliczeniowego.

Summary (2)

Given the following log file:

I. 1.1.1.1 [30:00:22:38] "GET /A.html HTTP/1.0" 200 156
II. 1.1.1.2 [30:00:29:47] "GET /B.html HTTP/1.0" 200 1788
III. 1.1.1.2 [30:00:41:47] "GET /C.htm HTTP/1.0" 200 1788
IV. 1.1.1.1 [30:00:41:55] "GET /D.html HTTP/1.0" 200 457
V. 1.1.1.2 [30:01:00:02] "GET /E.html HTTP/1.0" 200 1588
VI. 1.1.1.1 [30:01:15:47] "GET /F.html HTTP/1.0" 200 1788
VII. 1.1.1.2 [30:01:22:38] "GET /G.html HTTP/1.0" 200 1588

I) Identify the users using IP address and their sessions using H1 with timeout=10min or H2 with timeout=30min?

II) For each session, provide the number of request, compute its length, and an average pageview time.

[47] Zadania do samodzielnej realizacji jako powtórka:
I)
II)

Summary (3)

Given the following sessions/transactions:

{D1 D2 D3}, {D1 D3}, {D1 D3}, {D2 D3},

generate the frequent items while assuming that the minimal support is 3/4.

C1	support	L1
{D1}	3/4	√
{D2}	2/4	X
{D3}	**?**	**?**

candidates *frequent?*

I) Is $\{D_3\}$ frequent?

II) Can we generate $\{D_1, D_2\}$ as a candidate?

III) Can we generate $\{D_1, D_3\}$ as a candidate?

IV) Is $\{D_1, D_3\}$ frequent?

Assuming a minimal support equal to 3/4 and a minimal confidence – 0.8, **generate all association rules** $\{item_1\} \rightarrow \{item_2\}$.

V) Can we generate $D_1 \rightarrow D_2$ as a candidate? Why not?

VI) Which frequent itemset should be used as a basis for rule generation?

VII) What is the confidence of $D_1 \rightarrow D_3$ and $D_3 \rightarrow D_1$?

[48] Zadania do samodzielnej realizacji jako powtórka:

I)

II)

III)

IV)

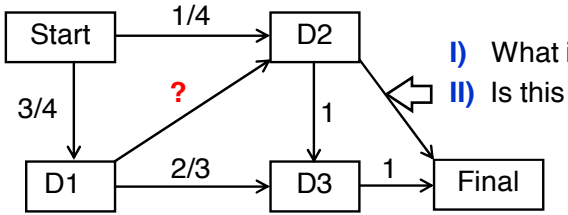
V)

VI)

VII)

Summary (4)

Given the following sessions: {D1 D2 D3}, {D1 D3}, {D1 D3}, {D2 D3}, **draw the Markov chain**.



I) What is P(D1→D2)?

II) Is this arrow making any sense?

III) What is the probability of starting/terminating a session in D2?

IV) What is the probability of P(Start→D1→D3)?

V) What is the probability of P(D3|D1)?

[49] Zadania do samodzielnej realizacji jako powtórka:

I)

II)

III)

IV)

V)