Web Usage Mining

Overview

— Session 1 —

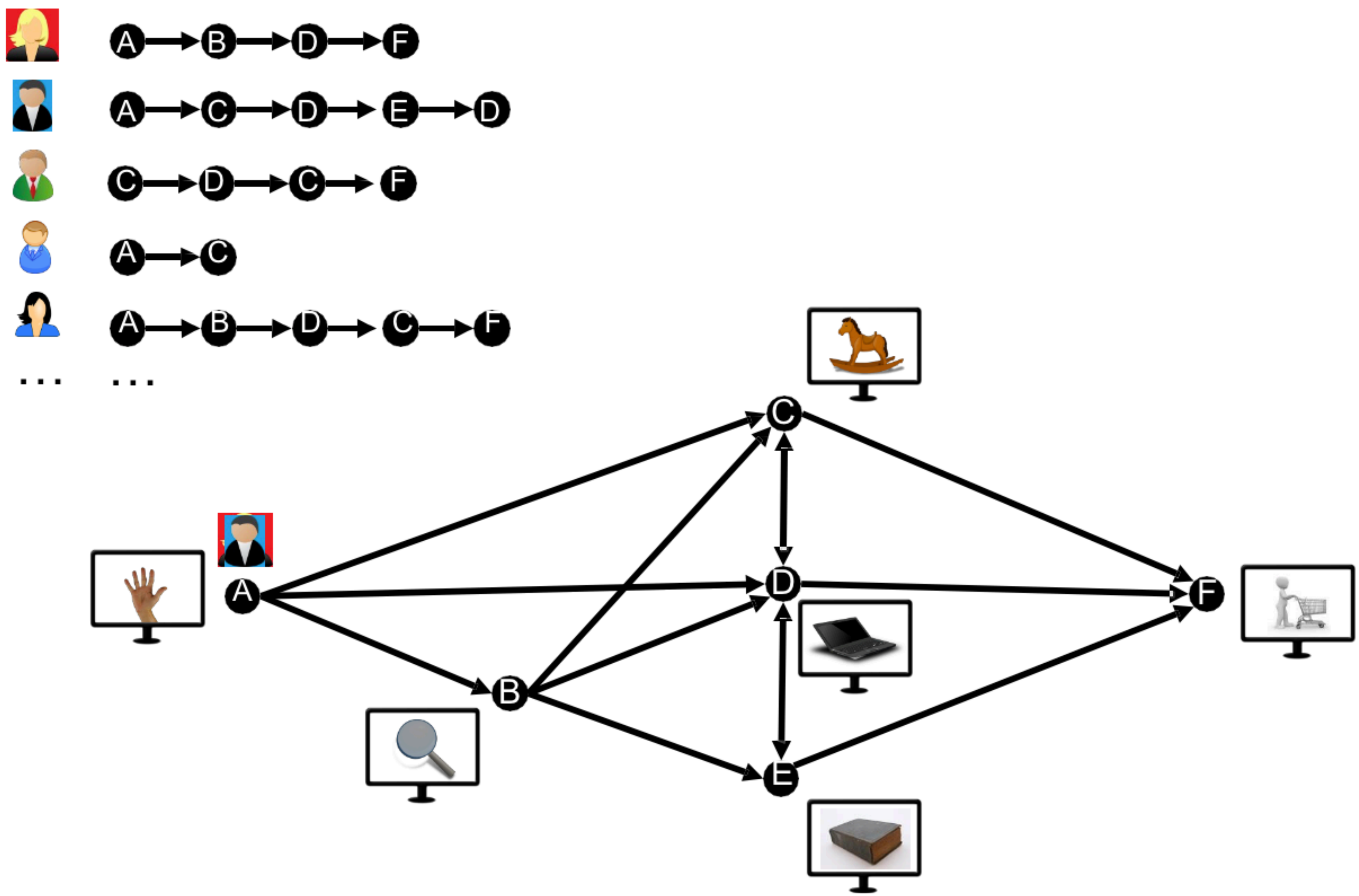
This material is inspired from the WWW'16 tutorial entitled ''Analyzing Sequential User Behavior on the Web''

Outline

1. Introduction
2. Preprocessing
3. Analysis

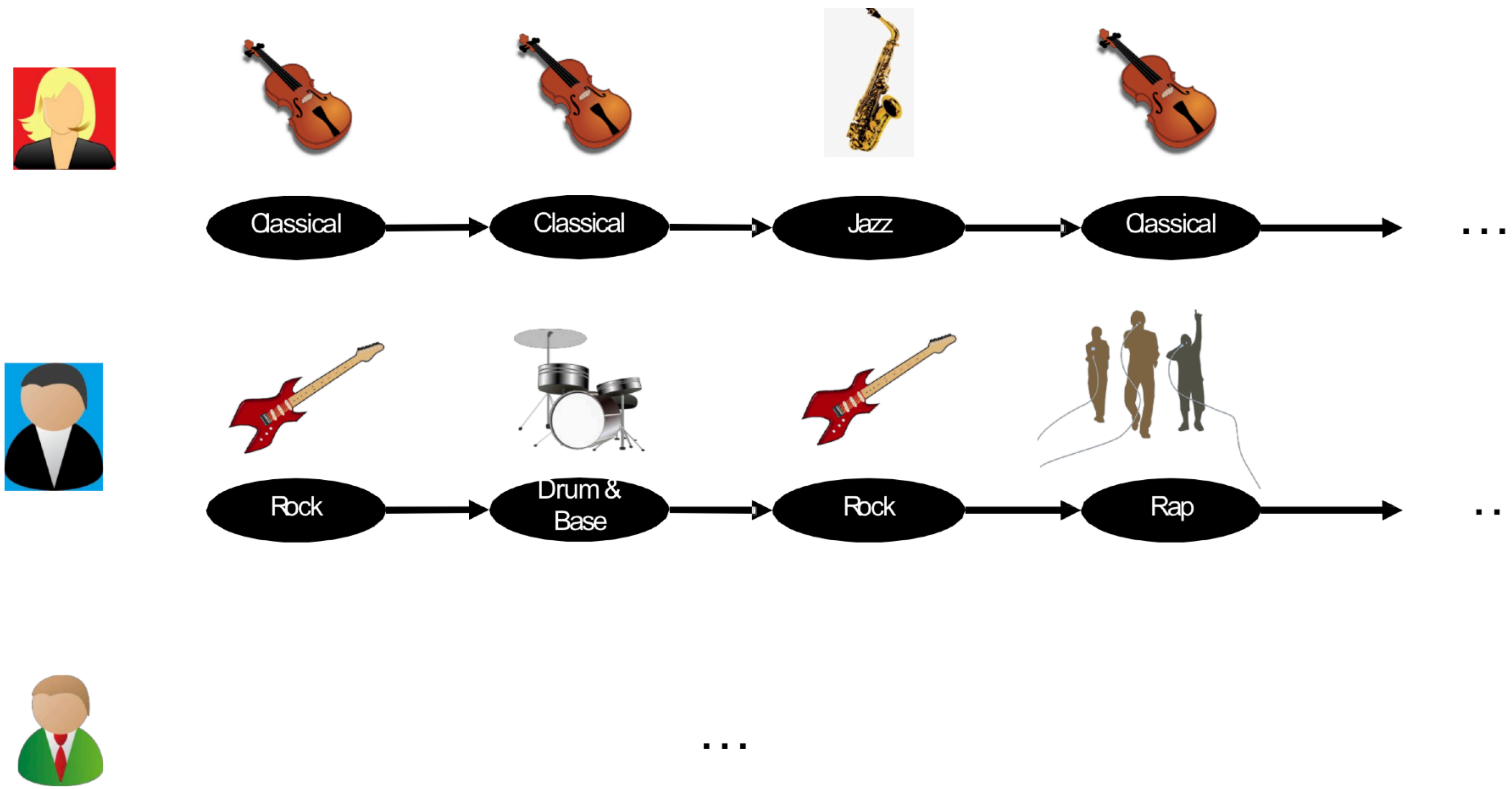
Example 1

Navigation through the Web



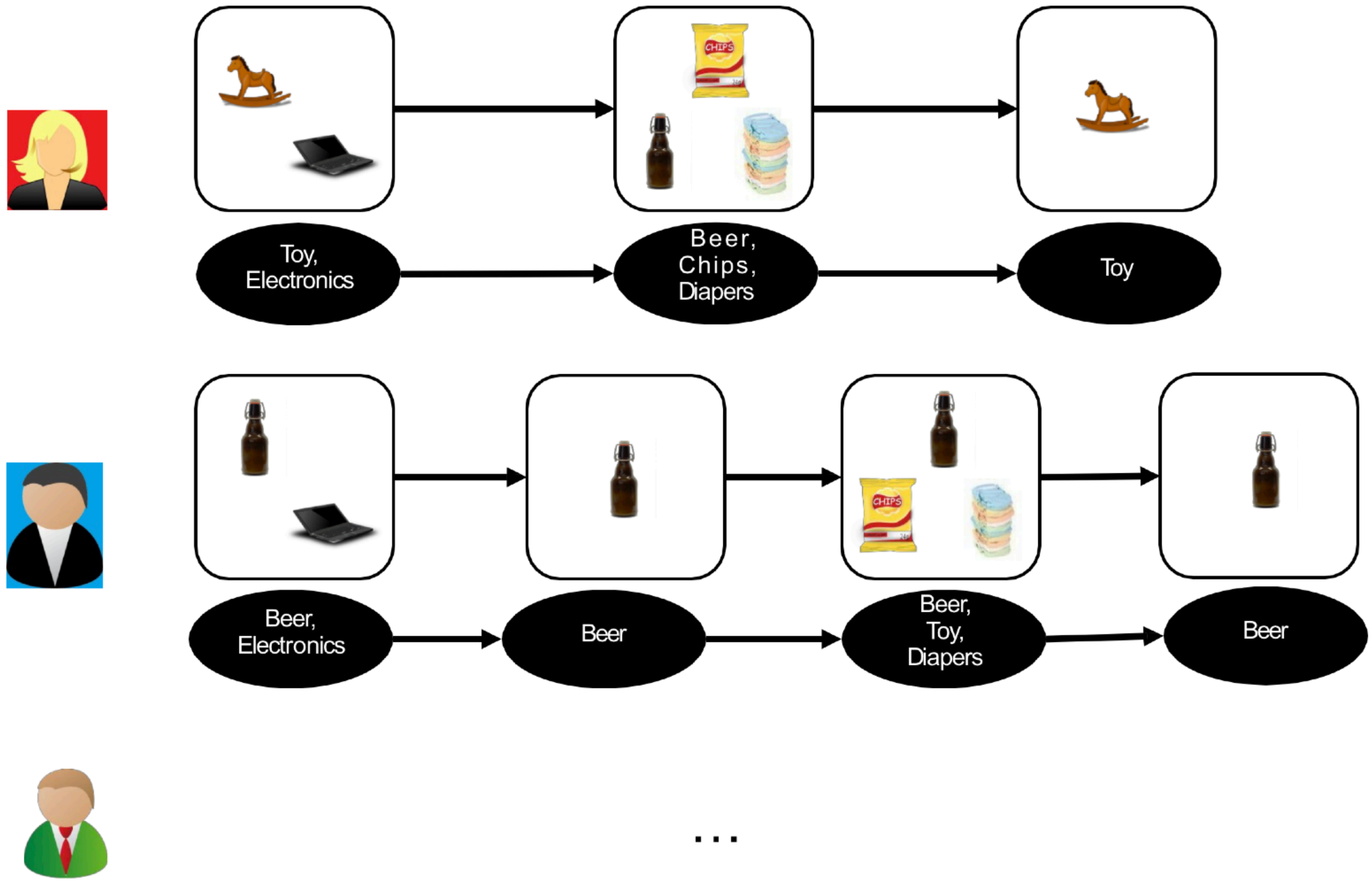
Example 2

Listening history



Example 3

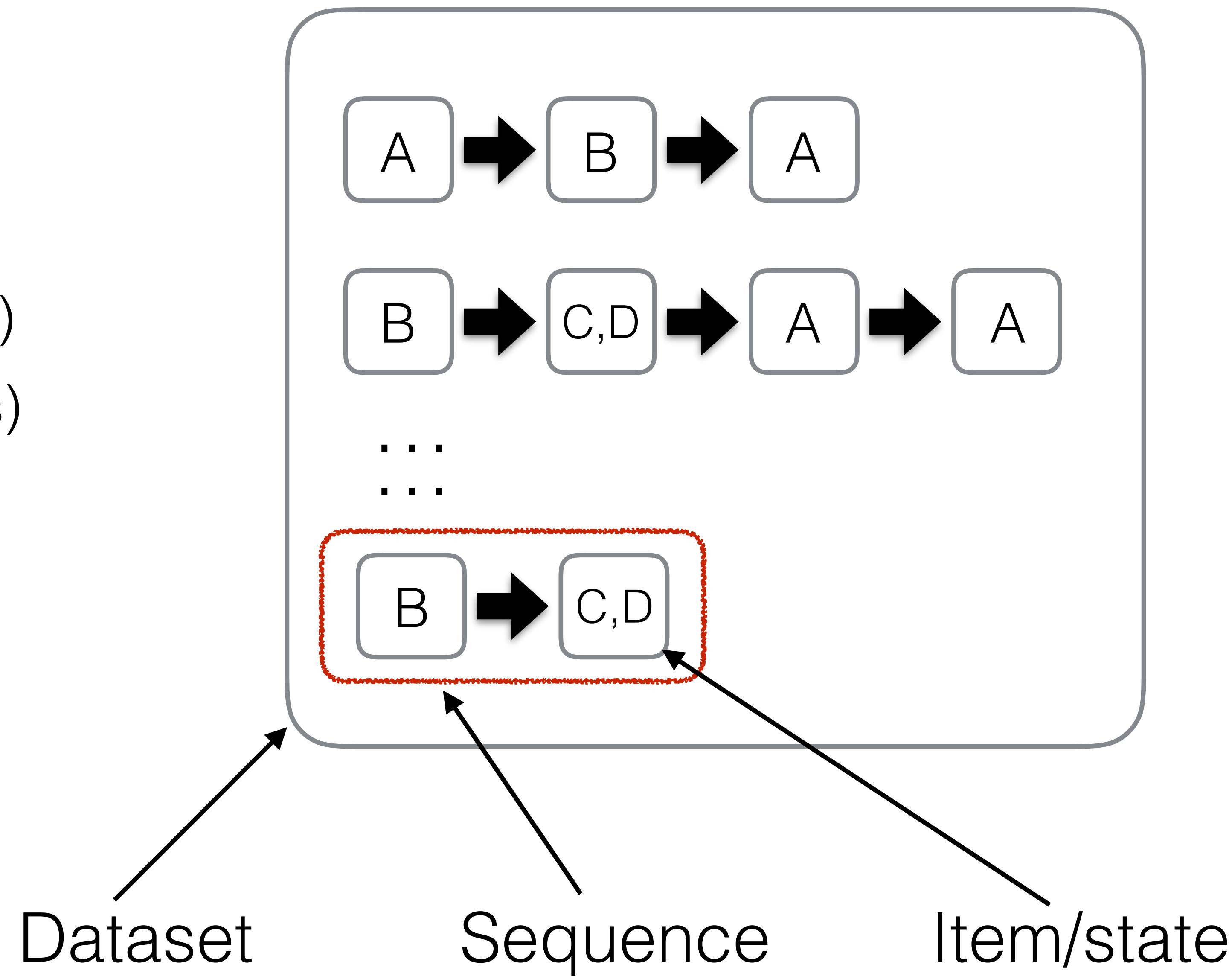
Shopping history



Manipulated data

- Set of sequences
- Sequence = several events
- An event has :
  - One categorical variable (state)
  - Multiple binary variables (items)

- Data are not numeric
- No timestamps
- No text data



Data sources

- Web logs

```
cragateway.cra.com.au [30:00:23:07] "GET /OSWRCRA/non-hw/muncpl/criteria HTTP/1.0" 302 -
ebaca.icsi.net [30:00:23:26] "GET /docs/TechInitiative HTTP/1.0" 302 -
ebaca.icsi.net [30:00:23:29] "GET /TechInitiative/ HTTP/1.0" 200 1994
ebaca.icsi.net [30:00:23:33] "GET /icons/circle_logo_small.gif HTTP/1.0" 200 2624
ebaca.icsi.net [30:00:23:45] "GET /docs/CSI HTTP/1.0" 302 -
ebaca.icsi.net [30:00:23:47] "GET /CSI/ HTTP/1.0" 200 493
ebaca.icsi.net [30:00:23:53] "GET /docs/CSI/CSI HTTP/1.0" 302 -
ebaca.icsi.net [30:00:23:55] "GET /CSI/CSI/ HTTP/1.0" 200 801
ebaca.icsi.net [30:00:24:01] "GET /docs/CSI/CSI/background HTTP/1.0" 302 -
systems61.fisher.su.oz.au [30:00:24:03] "GET / HTTP/1.0" 200 4889
ebaca.icsi.net [30:00:24:04] "GET /CSI/CSI/background/ HTTP/1.0" 200 871
ebaca.icsi.net [30:00:24:16] "GET /docs/CSI/CSI/background/facts.html HTTP/1.0" 200 101
cragateway.cra.com.au [30:00:24:17] "POST /cgi-bin/waisgate/134.67.99.11=earth1=210=/usr1/comwais/indexes/HTDOCS=gopher@earth1=0.00=:free HTTP/1.0" 200 2374
```

- Cookies
- Client-side tracking (e.g., mobile apps, eye-tracking)
- Web APIS
  - Reddit / Wikipedia
- Scrapping

Use cases

Human navigation

- User navigation from Web logs
- Strong regularities in WWW surfing
- Mining longest repeating subsequences for prediction
- Navigation on Wikipedia
  - Human way finding in information networks
  - Automatic vs. Human navigation
  - Memory and Structure

Use cases

Others

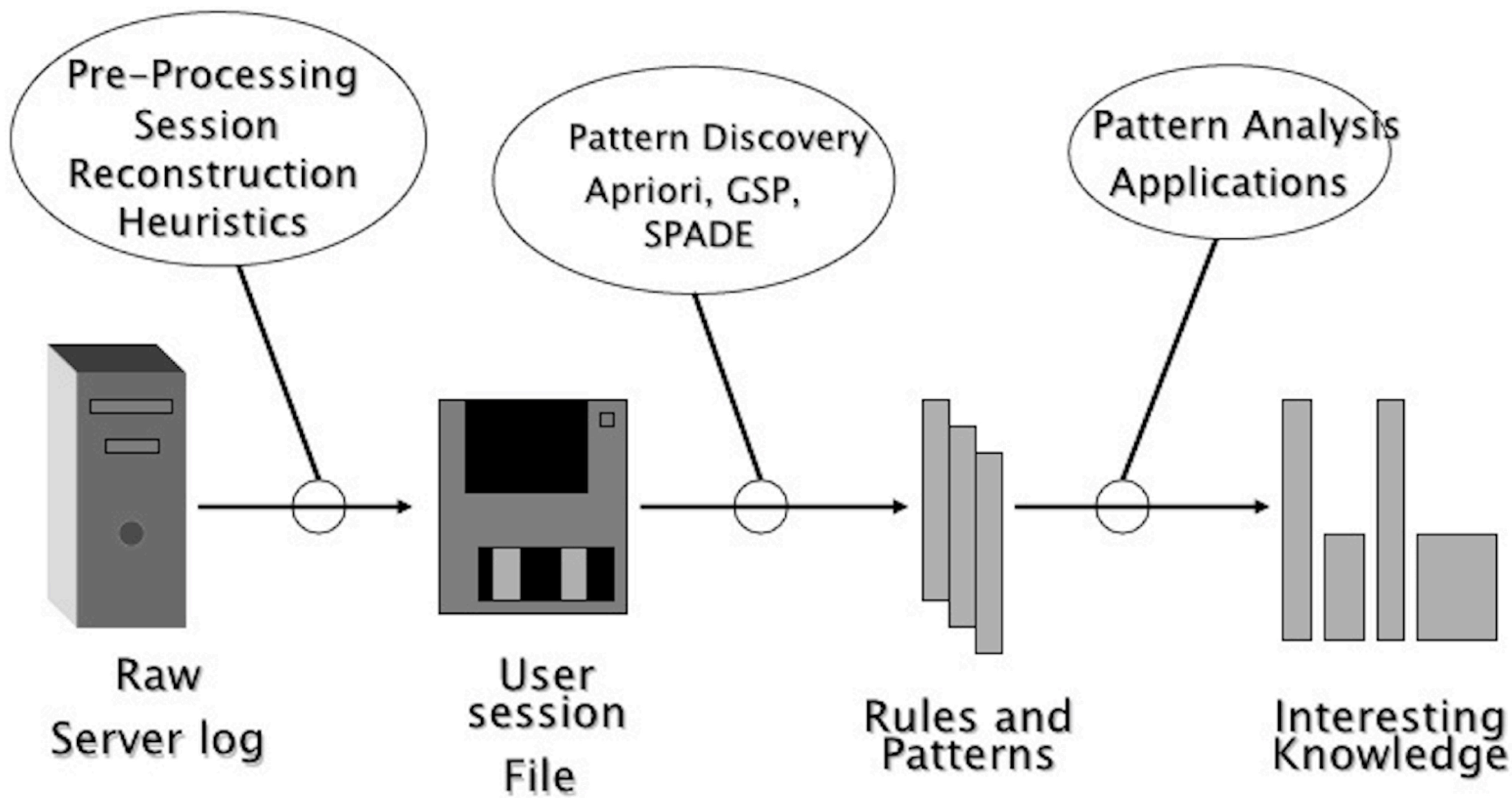
- Detecting (a-)typical user behavior
- Fake identity
- Improve web site design
- Personalization of Web content
  - Link recommendation
  - Personalized site maps
- Pre-fetching and caching
- E-commerce (CRM)
- Identification of relevant websites
- …

Privacy

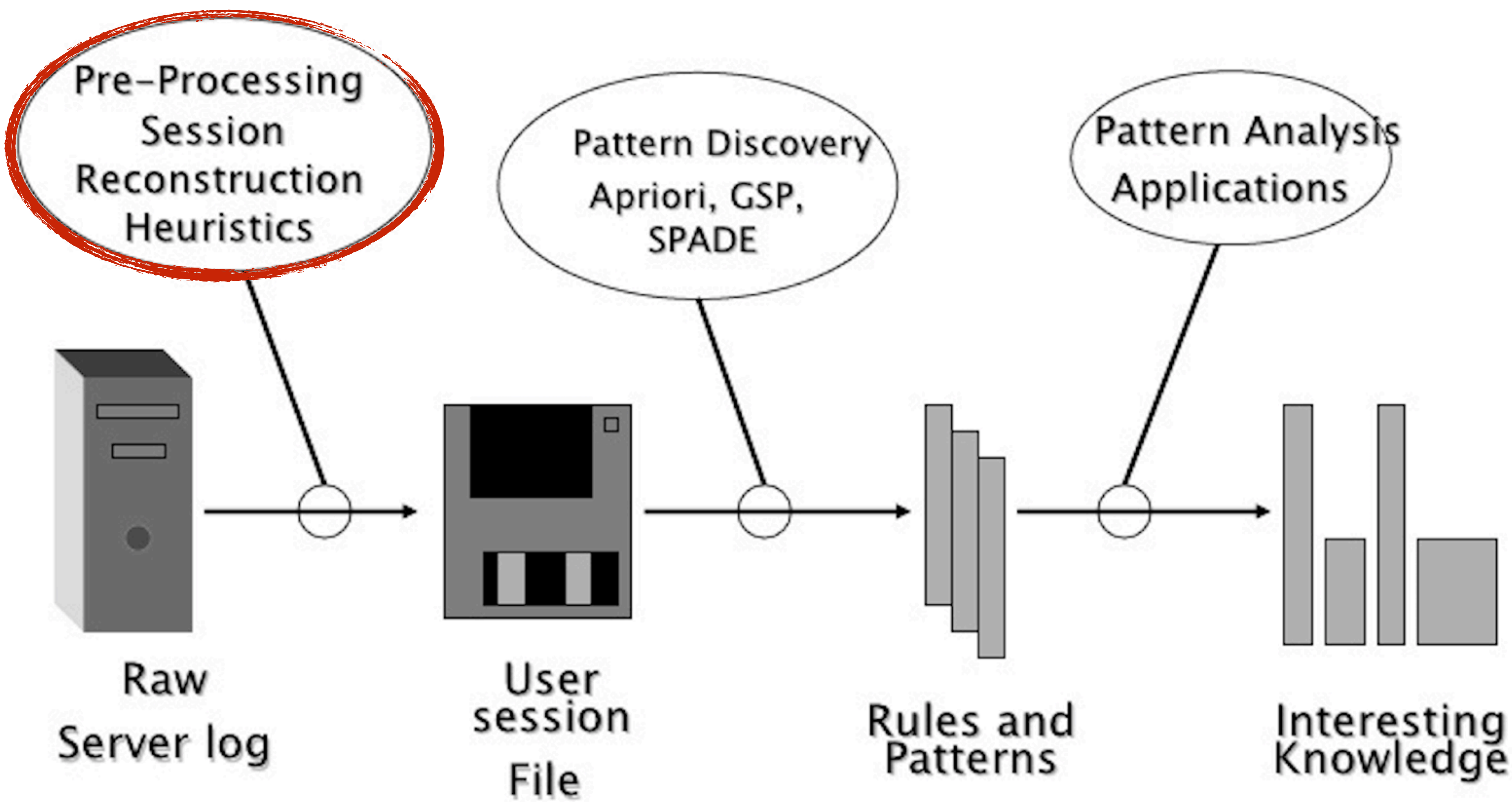
Ethical and Legal issues

- Ethical
  - WUM exploits user data (often no — conscious — agreement)
  - Users are judged on group features instead of individual merit
- Legal
  - Country dependent
  - Key question: IP = personal data ?
    - In France, yes (ask forCNIL agreement)
    - In EU, yes as well
  - Difference between academic and commercial use

The big picture

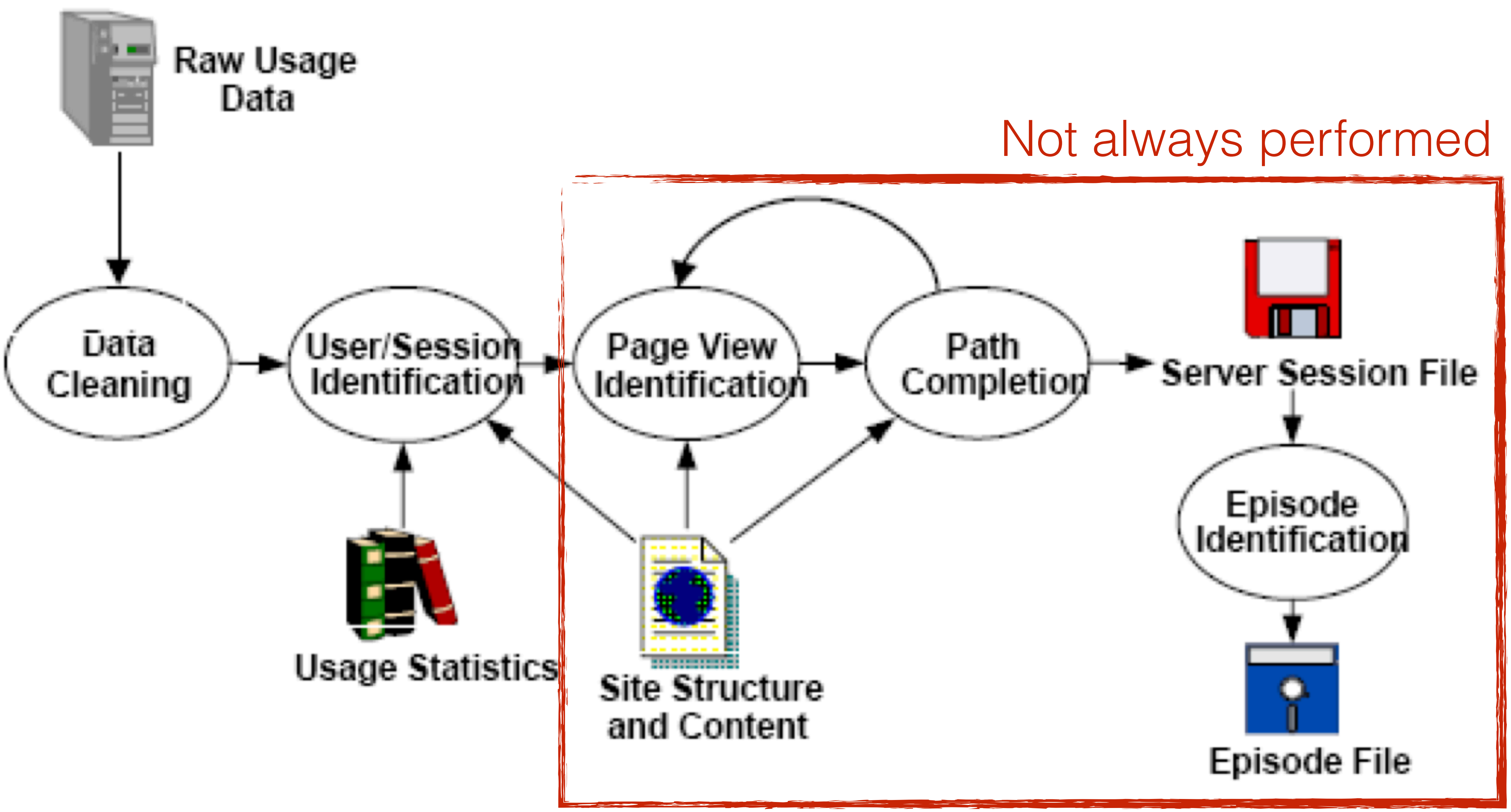


Step 1 — Preprocessing



Step 1 — Preprocessing

Details



Data cleaning

- Remove irrelevant references and fields in server logs
  - E.g., size of the response
- Remove references (entries) due to web spider (crawler) navigation
- Remove irrelevant content
  - E.g., scripts, images
- Remove erroneous references
- Add missing references due to caching (done after sessionization

Sessionization

Identifying sessions

- In Web usage analysis, these data are the sessions of the site visitors: the activities performed by a user from the moment he/she enters the site until the moment he/she leaves it.

- Difficult to obtain reliable usage data due to proxy servers and anonymizers, dynamic IP addresses, missing references due to caching, and the inability of servers to distinguish among different visits.

Sessionization

Strategies

Session reconstruction = correct mapping of activities to different individuals + correct separation of activities belonging to different visit of the same individual

While users navigate the site. Identify …		**In the analysis of log files. Identify …**		**Resulting partitionning of the log file**
Users by	**Sessions by**	**Users by**	**Sessions by**	
-	-	IP & Agent	Heuristics	Constructed sessions
Cookies	-		Heuristics	Constructed session
Cookies	Embedded session IDs			Real sessions

Sessionization

Heuristics

- Time-oriented heuristics
  - H1: total session duration must not exceed a maximum (user-defined)
  - H2: page stay times must not exceed a maximum (user-defined)
- Navigation-oriented heuristic
  - H3: A page must have been reached from a previous page in the same session

Sessionization

Example

User 1

Time	IP	URL	Ref
0:01	1.2.3.4	A	-
0:09	1.2.3.4	B	A
0:19	1.2.3.4	C	A
0:25	1.2.3.4	E	C
1:15	1.2.3.4	A	-
1:26	1.2.3.4	F	C
1:30	1.2.3.4	B	A
1:36	1.2.3.4	D	B

Session 1

0:01	1.2.3.4	A	-
0:09	1.2.3.4	B	A
0:19	1.2.3.4	C	A
0:25	1.2.3.4	E	C

Session 2

1:15	1.2.3.4	A	-
1:26	1.2.3.4	F	C
1:30	1.2.3.4	B	A
1:36	1.2.3.4	D	B

H1 = 1 hour / H2 = 15 min

User identification

Method	Description	Privacy concerns	Pro	Cons
IP + Agent	Pair (IP+agent) is a unique user	Low	Always available. No additional technology resquired	Not guaranteed to be unique (rotating IPs)
Embedded session Ids	Use dynamically generated pages to associate ID with every hyperlinks	Low to medium	Always available. Independent of IP	Cannot capture repeated visitors Additional overhead for dynamic pages
Registration	Use log in to the site	Medium	Can track individuals not just browsers	Few users will register. Not available before registration
Cookies	Save ID on the client machine	Medium to high	Can track repeated visit from same browser	Can be turned off by users
Software agents	Program loaded into a browser	High	Accurate usage data for a single site	Likely to be rejected by users

User identification

Example

Time	IP	URL	Ref	Agent
0:01	1.2.3.4	A	-	IE5;Win2k
0:09	1.2.3.4	B	A	IE5;Win2k
0:10	2.3.4.5	C	-	IE6;WinXP;SP1
0:12	2.3.4.5	B	C	IE6;WinXP;SP1
0:15	2.3.4.5	E	C	IE6;WinXP;SP1
0:19	1.2.3.4	C	A	IE5;Win2k
0:22	2.3.4.5	D	B	IE6;WinXP;SP1
0:22	1.2.3.4	A	-	IE6;WinXP;SP2
0:25	1.2.3.4	E	C	IE5;Win2k
0:25	1.2.3.4	C	A	IE6;WinXP;SP2
0:33	1.2.3.4	B	C	IE6;WinXP;SP2
0:58	1.2.3.4	D	B	IE6;WinXP;SP2
1:10	1.2.3.4	E	D	IE6;WinXP;SP2
1:15	1.2.3.4	A	-	IE5;Win2k
1:16	1.2.3.4	C	A	IE5;Win2k
1:17	1.2.3.4	F	C	IE6;WinXP;SP2
1:26	1.2.3.4	F	C	IE5;Win2k
1:30	1.2.3.4	B	A	IE5;Win2k
1:36	1.2.3.4	D	B	IE5;Win2k

User 1

0:01	1.2.3.4	A	-
0:09	1.2.3.4	B	A
0:19	1.2.3.4	C	A
0:25	1.2.3.4	E	C
1:15	1.2.3.4	A	-
1:26	1.2.3.4	F	C
1:30	1.2.3.4	B	A
1:36	1.2.3.4	D	B

User 2

0:10	2.3.4.5	C	-
0:12	2.3.4.5	B	C
0:15	2.3.4.5	E	C
0:22	2.3.4.5	D	B

User 3

0:22	1.2.3.4	A	-
0:25	1.2.3.4	C	A
0:33	1.2.3.4	B	C
0:58	1.2.3.4	D	B
1:10	1.2.3.4	E	D
1:17	1.2.3.4	F	C

Pageview

- A pageview is an aggregate representation of a collection of Web objects contributing to the display on a user's browser resulting from a single user action (such as a click-through).

- Conceptually, each pageview can be viewed as a collection of Web objects or resources representing a specific "user event," e.g., reading an article, viewing a product page, or adding a product to the shopping cart.

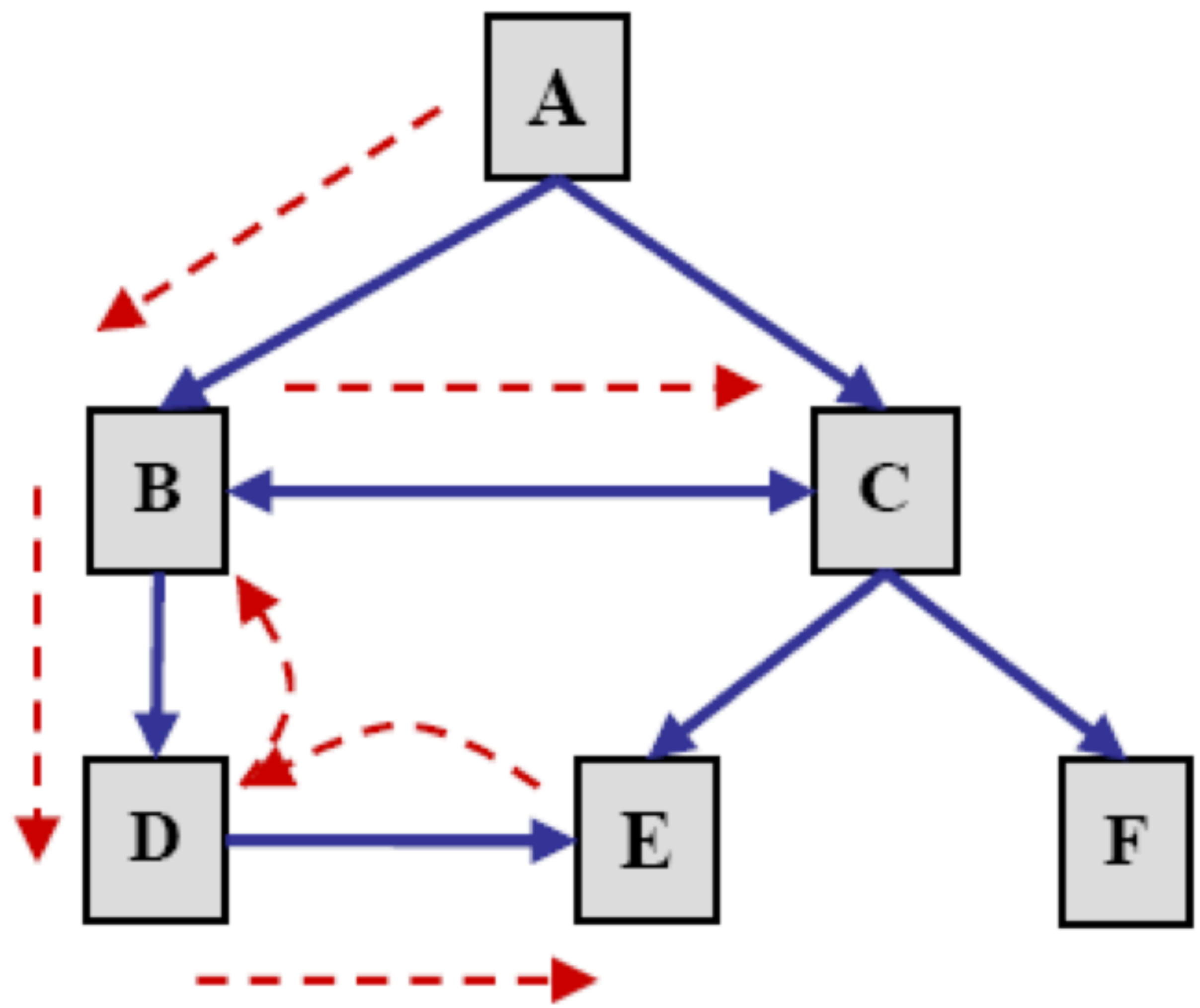
Path completion

Problem defintion

- Client- or proxy-side caching can often result in missing access references to those pages or objects that have been cached.
- For instance,
  - if a user returns to a page A during the same session, the second access to A will likely result in viewing the previously downloaded version of A that was cached on the client-side, and therefore, no request is made to the server.
  - This results in the second reference to A not being recorded on the server logs.

Path completion

Illustration



User's actual navigation path:

A →B → D → E → D → B → C

What the server log shows:

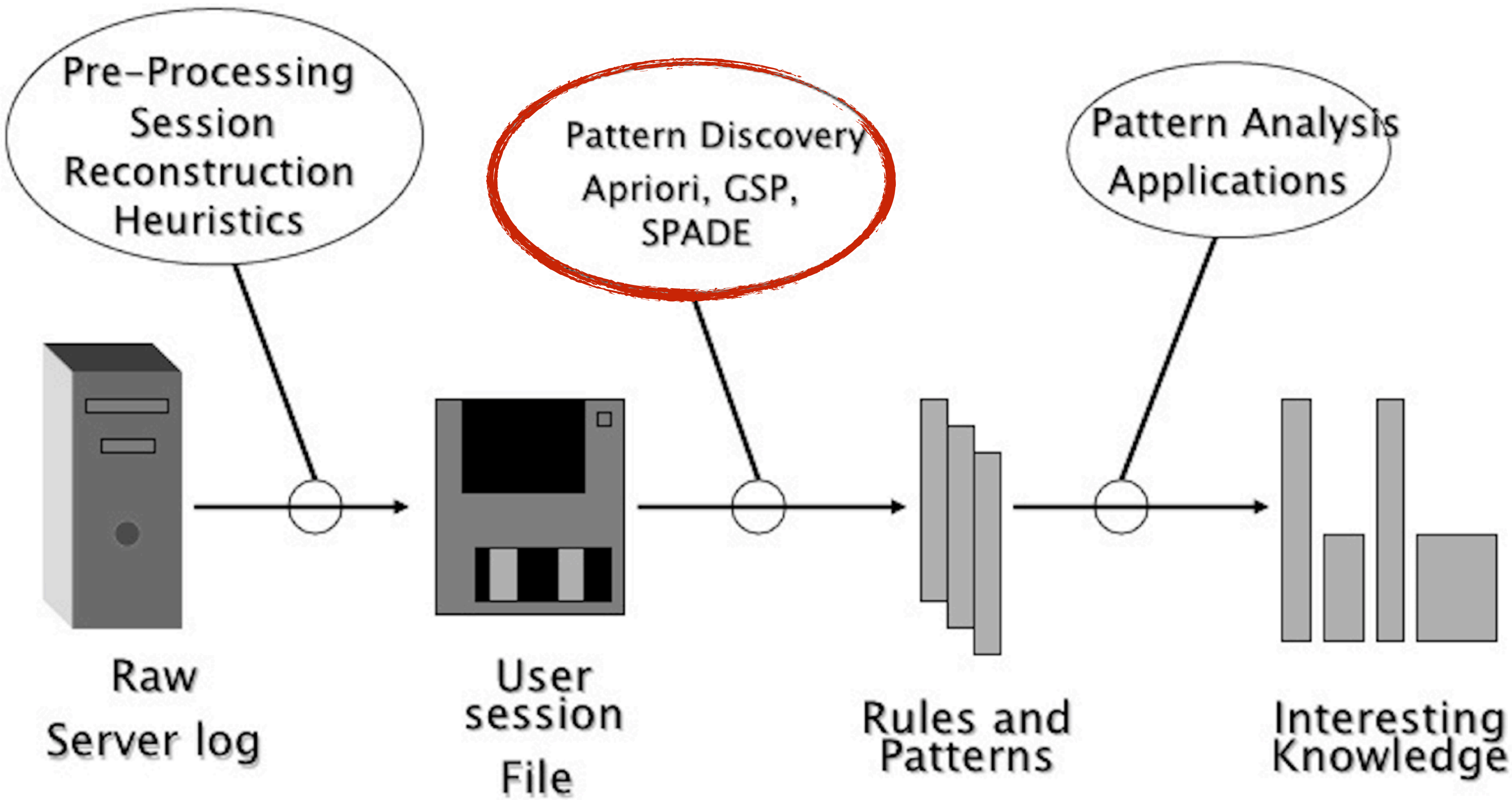
URL	Referrer
A	--
B	A
D	B
E	D
C	B

Path completion

Discussion

- The problem of inferring missing user references due to caching.
- Effective path completion requires extensive knowledge of the link structure within the site
- Referrer information in server logs can also be used in disambiguating the inferred paths.
- Problem gets much more complicated in frame-based sites.

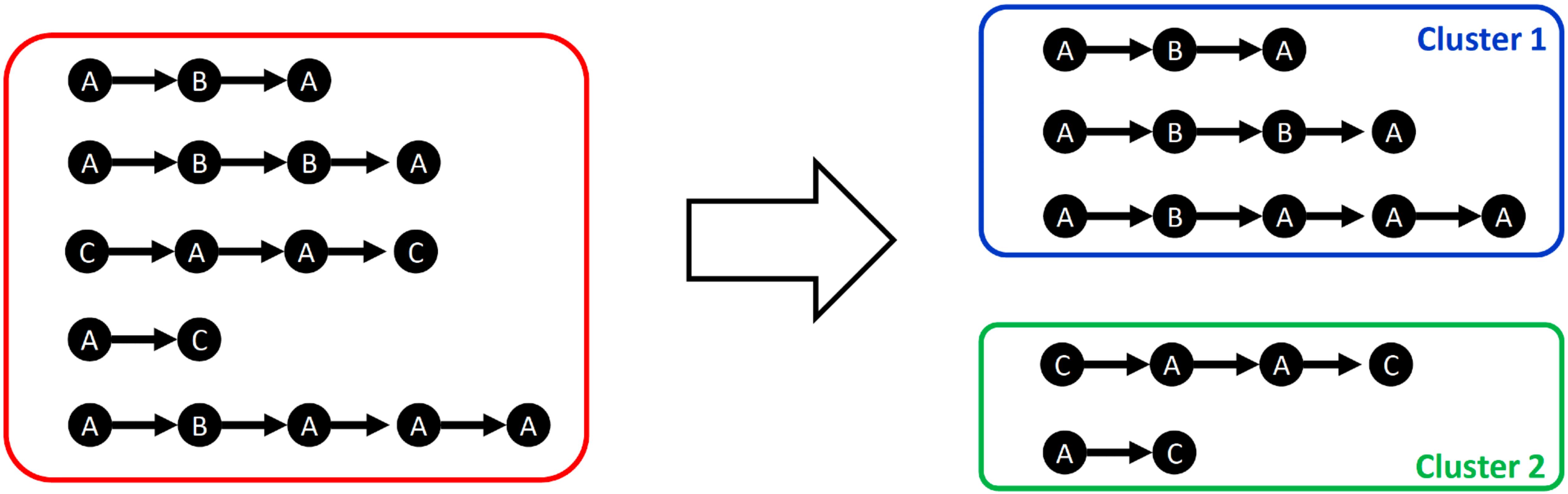
Step 2 — Analysis



Some tasks on sequence data

- Sequence clustering
- Sequence classification
- Sequence prediction
- Sequence segmentation
- Sequential pattern mining
- Sequence modeling

Sequence clustering

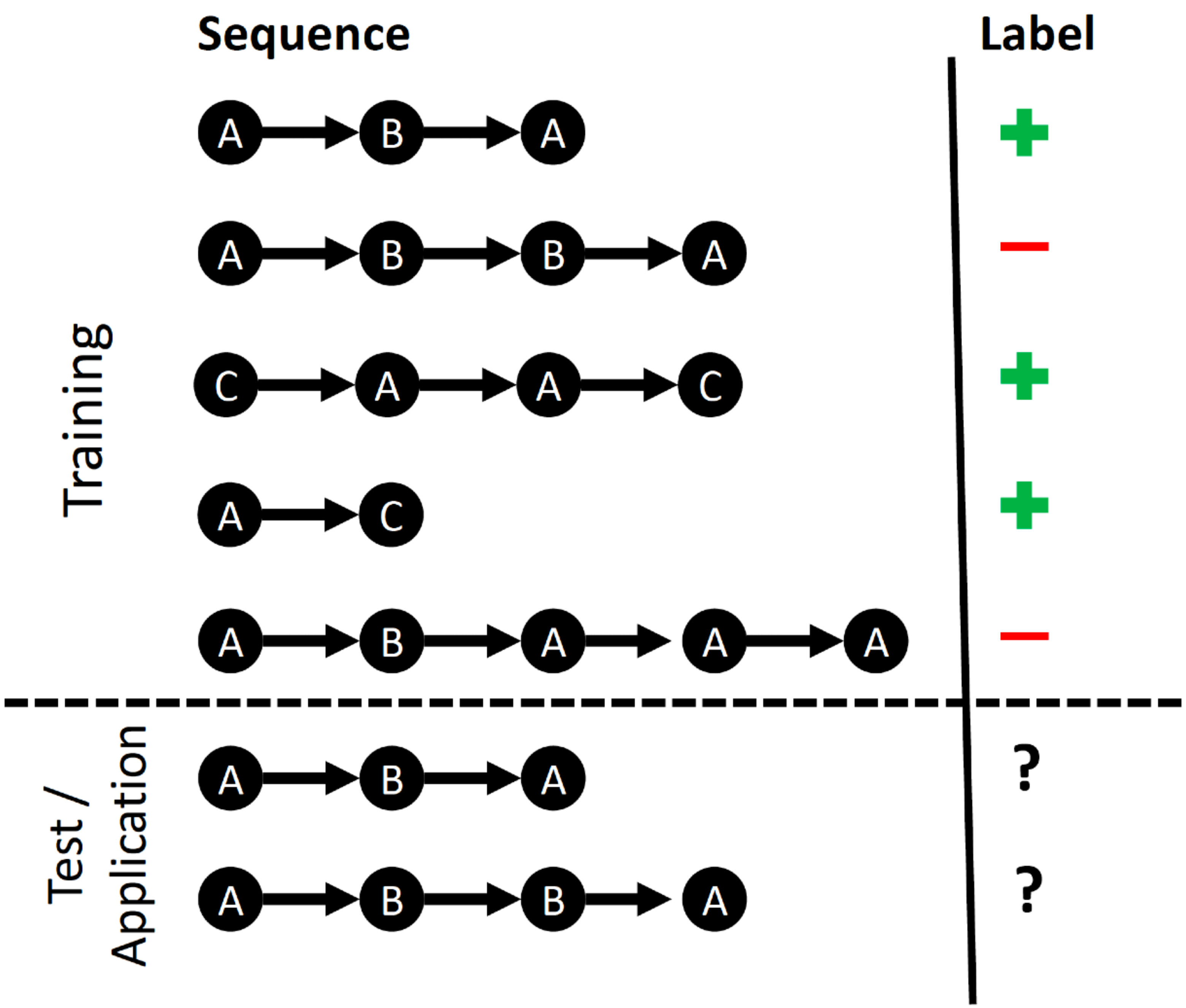


Sequence clustering

Overview

- Based on similarity sequence
  - For instance, edit distance (number of modifications)
  - All types of clustering algorithms can be applied
- Indirect clustering
  - Features : n-grams, sequential patterns
  - Use of standard clustering vector space algorithms on these features
- Statistical sequence clustering / model based clustering
  - Use set of Hidden Markov Models
  - Each model « generates » the sequence of one cluster
  - EM algorithm optimizes clusters and sequence-cluster mapping

Sequence classification

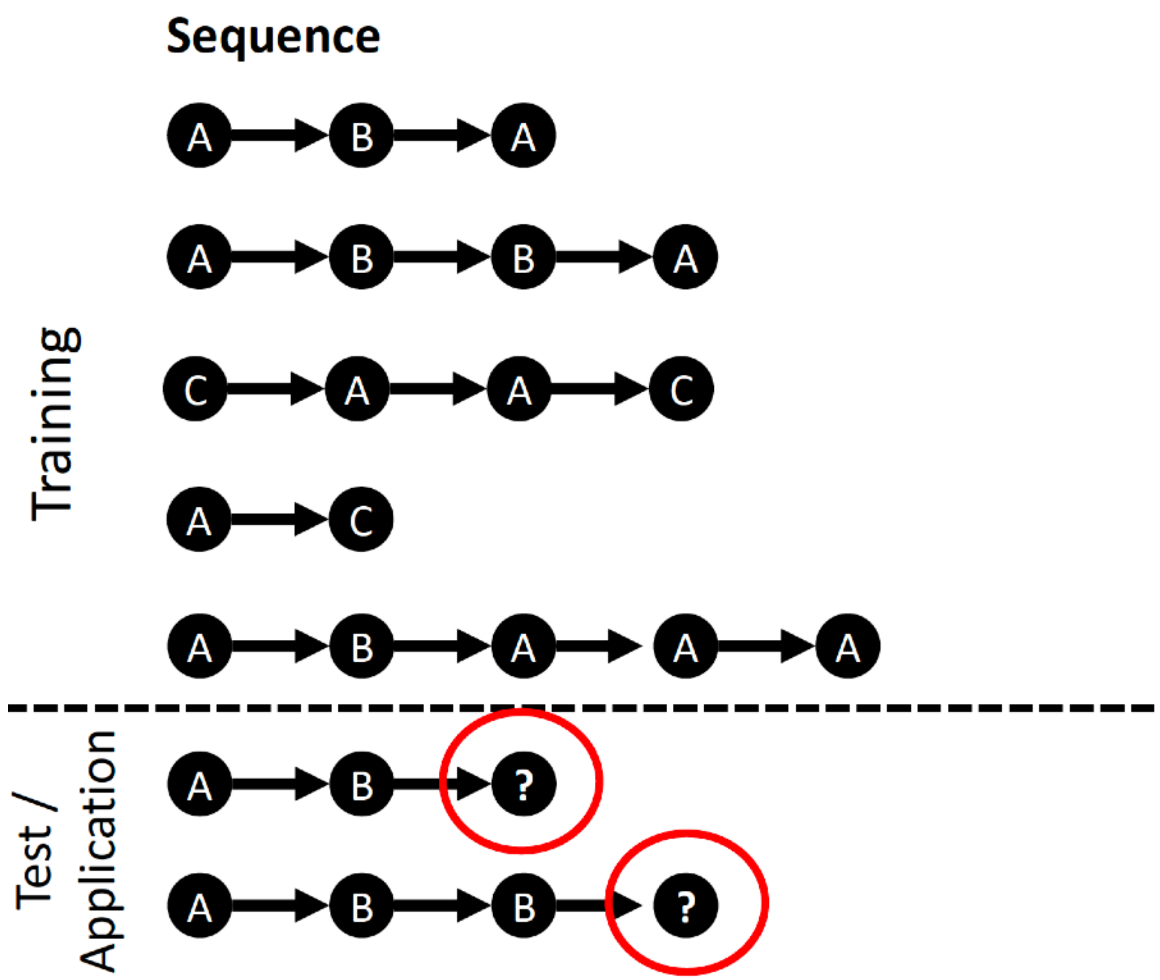


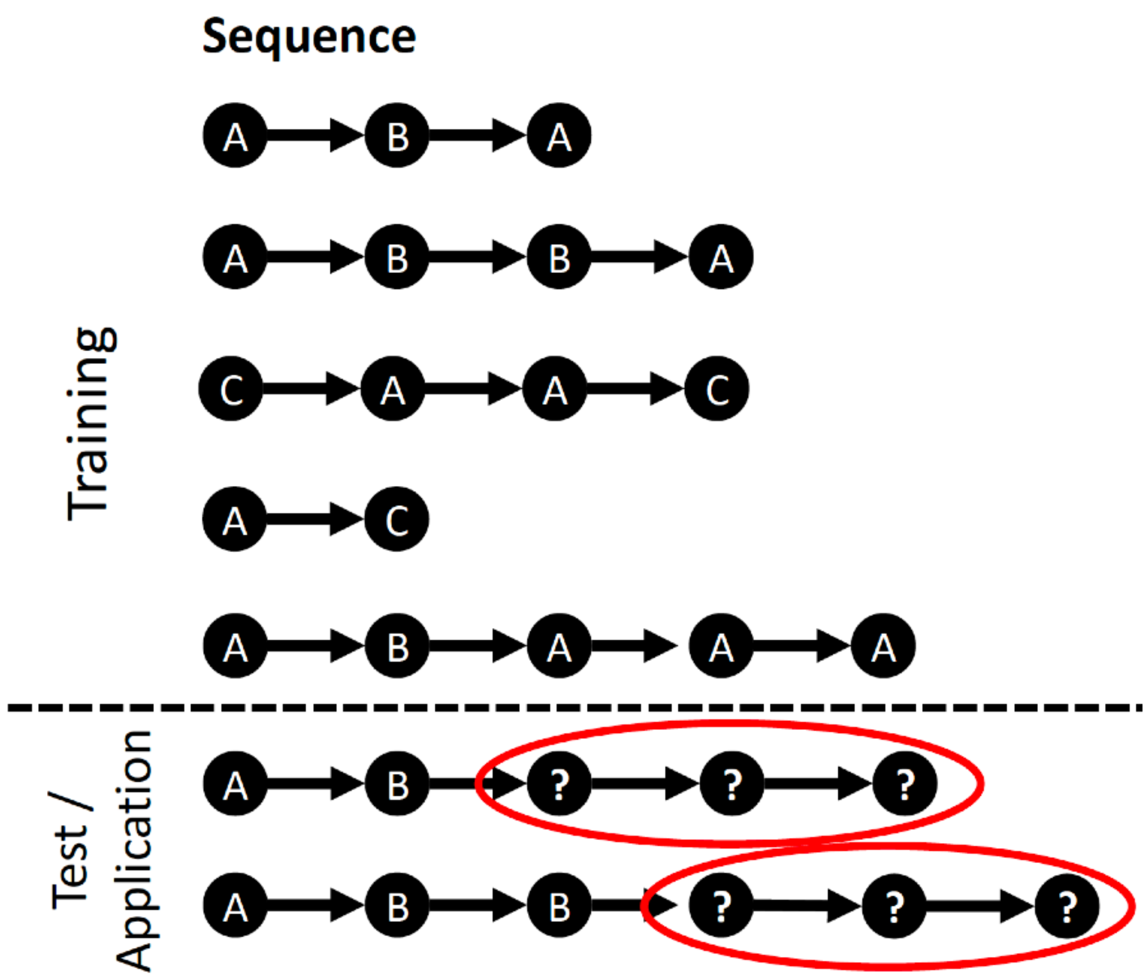
Sequence classification

Overview

- Using similarity metrics
  - See sequence clustering
  - Use of KNN
- Indirect classification
  - See sequence clustering
  - Use of any classification algorithm
  - SVM + string kernel
- Model-based classification
  - Discriminatively trained Markov Models
    - Different variations of Hidden Markov Models

Sequence prediction/generation



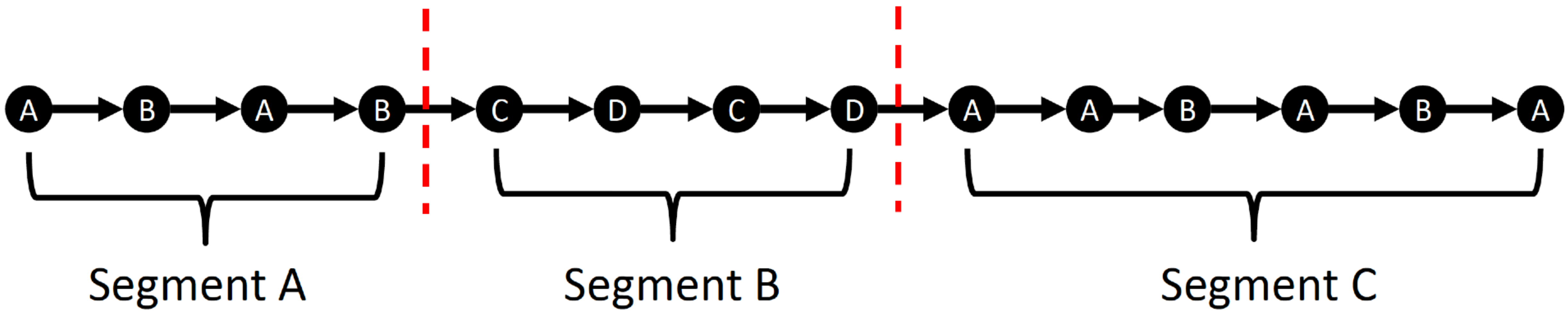


Sequence prediction

Overview

- (Hidden) Markov Models
- (Partially ordered) Sequential rules (based on sequential patterns)
- Recurrent Neural Networks (RNNs)

Sequence segmentation



Sequence segmentation

Overview

- Applications
  - **Behavioral stages of web users**
  - DNA segmentation
  - Text segmentation
- Methods
  - If timestamp is available: similar to discretization
  - Models + MDL
  - Set of models, optimizes (log-) likelihood

Some tasks on sequence data

- Sequence clustering
- Sequence classification
- Sequence prediction
- Sequence segmentation
- **Sequential pattern mining**
- Sequence modeling

Sequential pattern mining

Definition

"Given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and given a user-specified min_support threshold, sequential pattern mining is to find all of the frequent subsequences, i.e., the subsequences whose occurrence frequency in the set of sequences is no less than min_support."

[Agrawal & Srikant, 1995]

"Given a set of data sequences, the problem is to discover sub-sequences that are frequent, i.e., the percentage of data sequences containing them exceeds a user-specified minimum support."

[Garofalakis, 1999]

Sequential pattern mining

Notations and Preliminary Concepts

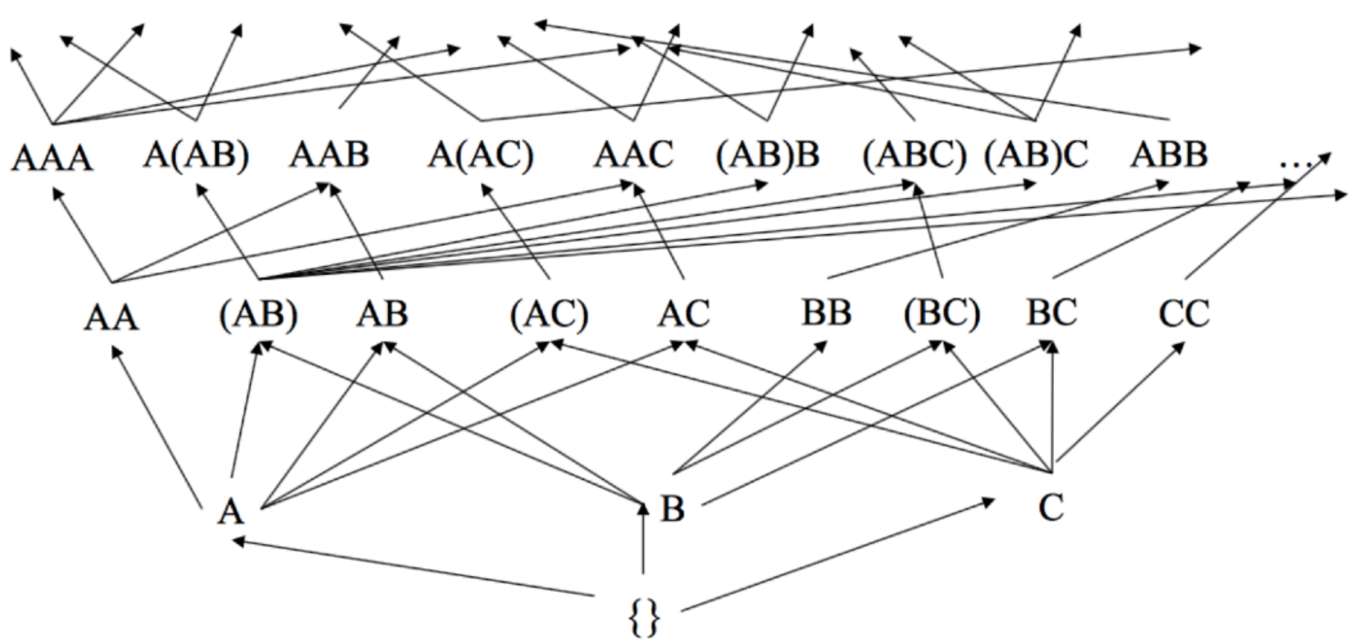
- Data
  - Dataset: set of sequences
  - Sequence: ordered list of itemsets (events) $<e_1,\ldots,e_n>$
  - Itemset: unordered set of items $e_i = \{i_{i1},\ldots,i_{iz}\}$
- Subsequence
  - $S_{sub} = <s_1,\ldots,s_n>$ is a subsequence of $S_{ref} = <r_1,\ldots,r_m>$ iff:

$$\exists i_1 < ... < i_n : s_k \subseteq r_{i_k}$$

    - <a,(a,b),c> is a subsequence of <d,(a,c),e,(a,b),(c,d)>
  - Length of a sequence: number of items used in the sequence (not unique)
    - Length(<a,(a,b),c> = 4

Frequent sequential pattern

- The support — sup(S) — of a sequence S is the number of sequences in the dataset that have S as a subsequence
- Given a user-defined threshold, minSupp, S is said to be frequent if sup(S) $\geq$ minSupp
- Objective: find all frequent sequences in the dataset
- Huge pattern space… brute force algorithm does not scale



E.g., 100 items
Around 10^{30} candidates for sequences of length 3…

Monotonicity

- If S is a subsequence of R then sup(S) $\geq$ sup(R)
- Monotonicity
  - If S is not frequent, R cannot be frequent!
- Pruning
  - If we know S is not frequent, it is useless to evaluate any supersequence of S!

A wide variety of algorithms

- Apriori and variants
  - A *levelwise* approach
    - Find frequent patterns with length 1
    - Use them to find frequent patterns with length 2
    - …
- SPaDE and variants
  - Vertical data representation
  - Use of equivalence classes

SID	Time	Items
1	10	a, b, d
1	15	b, d
1	20	c
2	15	a
2	20	b, c, d
3	10	b, d

(Original) Horizontal database layout

a

SID	Time
1	10
2	15

b

SID	Time
1	10
1	15
2	20
3	10

c

SID	Time
1	20
2	20

d

SID	Time
1	10
1	15
2	20
3	10

Vertical database layout

- PrefixSpan and variants
  - Projected databases
  - Recursive mining

Given Sequence	Projection to a
< b, (c,d), a, (b d), e >	<a, (b,d), e>
<c, (a,d), b, (d,e)>	<(a,d), b, (d,e)>
<b, (de), c>	[will be removed]

Redundancy problem

- If <a, (bc), d> is frequent so are <a>, <b>, <c>, <d>, <a,b>… (i.e., all the subsequences of <a, (bc), d>)
- Presenting such frequent subsequences is of little interest to the users
- Need for some compressed representations
  - Lossless compression: closed sequential patterns
  - Lossy compression: maximal sequential patterns

Closed and Maximal Sequential Patterns

- Frequent closed sequences
  - All the super-sequences have a smaller support
- Frequent maximal sequences
  - All super-sequences are not frequent

Dataset
<a, b, c, d, e, f>
<a, c, d>
<c, b, a>
<b, a, (de)>
<b, a, c, d, e>

sup (<a,c>) = 3 → frequent
sup (<a,c,d>) = 3 → frequent, closed
sup (<a,c,d,e>) = 2 → frequent, closed, max.
sup (<a,c,d,e,f>) = 1 → not frequent

Mining closed and maximal patterns

- Specialized algorithms exist
  - Much faster!!
- Well-known examples
  - Closed patterns
    - CloSpan [Yan et al., 2003]
    - BIDE [Wang et Han, 2007]
  - Maximal
    - VMSP [Fournier-Viger et al. 2014]

Which one should you use?

- All give the same results
- Only a matter of memory usage and runtime
- No clear conclusions from existing studies
- In practice:
  - Dense dataset: SPaDE and variations
  - Sparse dataset: PrefixSpan and variations

- But that strongly depends on the implementation…

Beyond the frequency

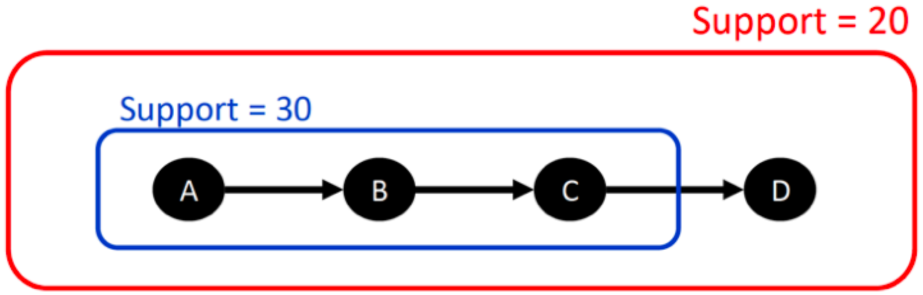
- Frequent patterns are not always the most interesting ones
- E.g., in web logs of a e-commerce website < home, paiement> has good chances to be frequent but is not really of interest…
- Two alternatives:
  - Adding constraints
  - Use of interestingness measures

Adding constraints

- ***Item constraints****:* e.g., high-utility items: Sum all items in the sequence > 1000$
- ***Length constraint****:* Minimum/maximum number of events/transactions
- ***Model-based constraints****:* Sub-/supersequences of a given sequence
- ***Gap constraints****:* Maximum gap between events of a sequence
- ***Time constraints****:* Given timestamps, maximum time between events of a sequence
- Computation:
    1. Mine all frequent patterns and then filter
    2. Push constraints into the algorithm

Interestingness measures

- Many measures have been proposed
  - See *Interestingness measures for data mining: A survey [Genq et Hamilton, 2006]* for a non-exhaustive though long list
  - One of the most well-known measure: the confidence
    - Split the pattern into two components, the head (last itemset) and the tail (the rest)
    - Calculate the probability P(head | tail)
    - Should not be considered as a causality measure



Confidence (< A, B, D > ➔ F) = $\frac{20}{30}$

Available software libraries

- Java:
  - SPMF (most extensive library) http://www.philippe-fournier-viger.com/spmf/
  - Basic support in RapidMiner, KNIME
- R
  - arulesSequences package
  - TraMiner package
- Python
  - Multiple basic implementations
- Spark: PrefixSpan available