LASH: Large-Scale Sequence Mining with Hierarchies

Kaustubh Beedkar and Rainer Gemulla

Data and Web Science Group
University of Mannheim

June 2nd, 2015
SIGMOD 2015

Syntactic Explorer (Verb to Verb Noun)

SyntEx

-[VB]/@lemma to -[VB]/@lemma -[NN]/@lemma

Displaying top-20 sequences

want to do something	2152
have to do something	2103
authorize to seek contribution	1103
want to be part	1082
be to take place	1027
decline to comment yesterday	1011
try to do something	932
want to go home	675
try to take advantage	634
want to do anything	632
have to take care	623
refuse to answer question	618
expect to announce today	597
go to do something	594
adjust to represent sale	590
weight to represent sale	563
go to do anything	552

Sequence Mining

- Goal: Discover subsequences as patterns in sequence data
- Input: Collection of sequences of items, e.g.,
  - Text collection (sequence of words)
  - Customer transactions (sequence of products)

Sequence Mining

- Goal: Discover subsequences as patterns in sequence data

- Input: Collection of sequences of items, e.g.,
  - Text collection (sequence of words)
  - Customer transactions (sequence of products)

- Output: subsequences that
  - occur in σ input sequences (frequency threshold)
  - have length at most λ (length threshold)
  - have gap γ (contiguous subsequences or non-contiguous subsequences)

Sequence Mining

- Goal: Discover subsequences as patterns in sequence data
- Input: Collection of sequences of items, e.g.,
  - Text collection (sequence of words)
  - Customer transactions (sequence of products)
- Output: subsequences that
  - occur in σ input sequences (frequency threshold)
  - have length at most λ (length threshold)
  - have gap γ (contiguous subsequences or non-contiguous subsequences)
- Example:

  S_1: Anna lives in Melbourne
  S_2: Bob lives in the city of Berlin
  S_3: Charlie likes London

Sequence Mining

- Goal: Discover subsequences as patterns in sequence data

- Input: Collection of sequences of items, e.g.,
  - Text collection (sequence of words)
  - Customer transactions (sequence of products)

- Output: subsequences that
  - occur in σ input sequences (frequency threshold)
  - have length at most λ (length threshold)
  - have gap γ (contiguous subsequences or non-contiguous subsequences)

- Example:

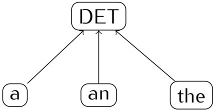
  S_1: Anna lives in Melbourne
  S_2: Bob lives in the city of Berlin
  S_3: Charlie likes London
  - Subsequence: lives in
    $\sigma = 2, \lambda = 2, \gamma = 0$

Hierarchies

Items can be naturally arranged in a hierarchy, e.g.,

Hierarchies

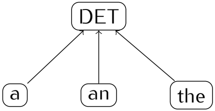
Items can be naturally arranged in a hierarchy, e.g.,



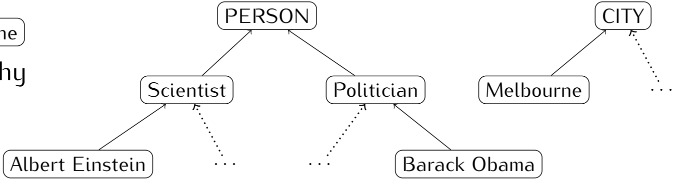
Syntactic hierarchy

Hierarchies

Items can be naturally arranged in a hierarchy, e.g.,



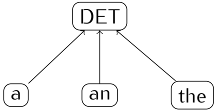
Syntactic hierarchy



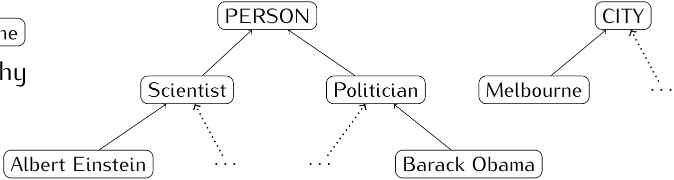
Semantic hierarchy

Hierarchies

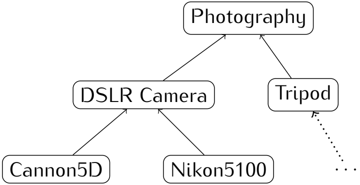
Items can be naturally arranged in a hierarchy, e.g.,



Syntactic hierarchy



Semantic hierarchy



Product hierarchy

Sequence Mining with Hierarchies

- Item hierarchies are specifically taken into account
- Discover non-trivial patterns

Sequence Mining with Hierarchies

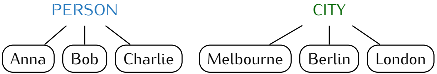
- Item hierarchies are specifically taken into account
- Discover non-trivial patterns
- Example

  S_1: Anna lives in Melbourne
  S_2: Bob lives in the city of Berlin
  S_3: Charlie likes London

Sequence Mining with Hierarchies

- Item hierarchies are specifically taken into account
- Discover non-trivial patterns
- Example

  S_1: Anna lives in Melbourne
  S_2: Bob lives in the city of Berlin
  S_3: Charlie likes London

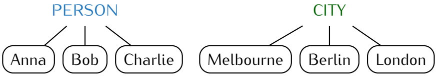


Semantic hierarchy

Sequence Mining with Hierarchies

- Item hierarchies are specifically taken into account
- Discover non-trivial patterns
- Example

  S_1: Anna lives in Melbourne
  S_2: Bob lives in the city of Berlin
  S_3: Charlie likes London



Semantic hierarchy

  - Generalized subsequence:
    PERSON lives in CITY
    $\sigma = 2, \lambda = 4, \gamma = 3$

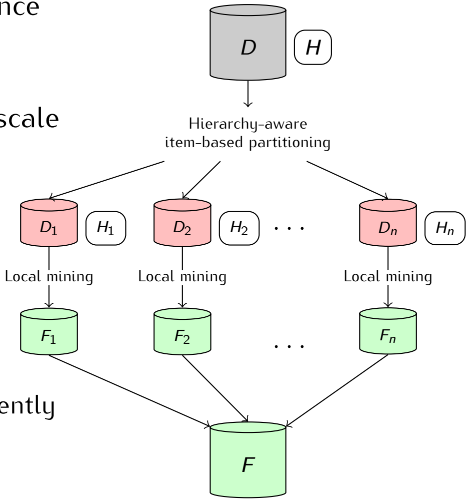
Sequence Mining with Hierarchies

Applications

- Linguistic patterns, e.g.,
  - read DET book
  - NNP lives in NNP
- Information extraction, e.g.,
  - PERSON lives in CITY
- Market-basket analysis, e.g,
  - buy DSLR camera → photography book → flash
- Web-usage mining
- ...

LASH

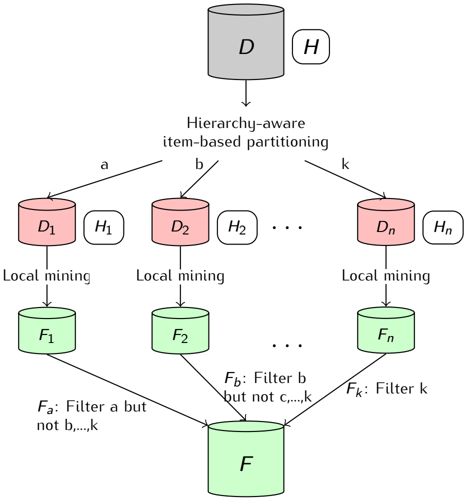
- Distributed framework for sequence mining with hierarchies
- Built over MapReduce for large-scale data processing
- Map (Partitioning)
  - Divide data into potentially overlapping partitions
- Reduce (mining)
  - Partitions are mined independently
- No global post-processing



Outline

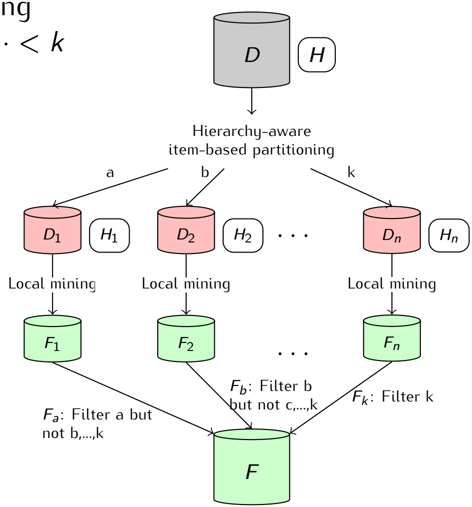
1. Introduction
2. Partitioning
3. Local Mining
4. Evaluation
5. Conclusion

Item-based Partitioning



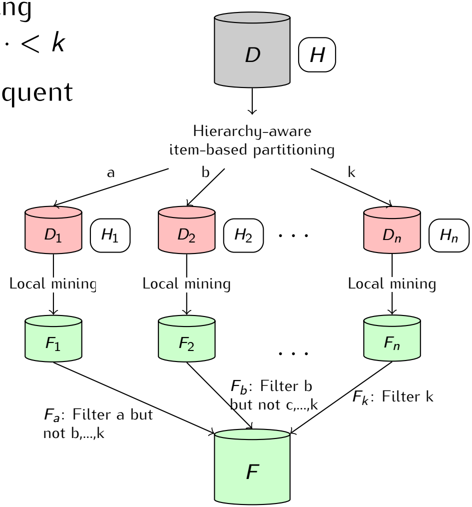
Item-based Partitioning

- Items are ordered by decreasing frequency, e.g., $a < b < c < \cdots < k$



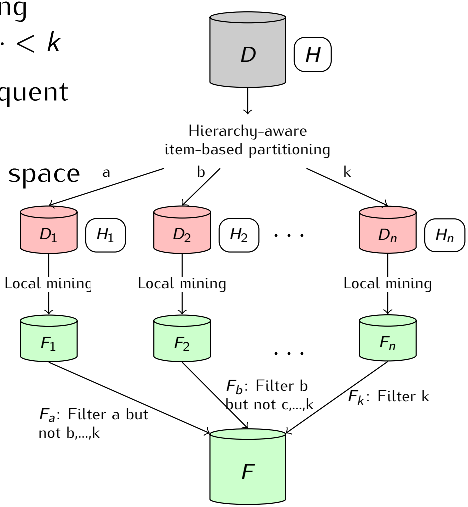
Item-based Partitioning

- Items are ordered by decreasing frequency, e.g., $a < b < c < \cdots < k$
- Create a partition for each frequent item called **pivot item**



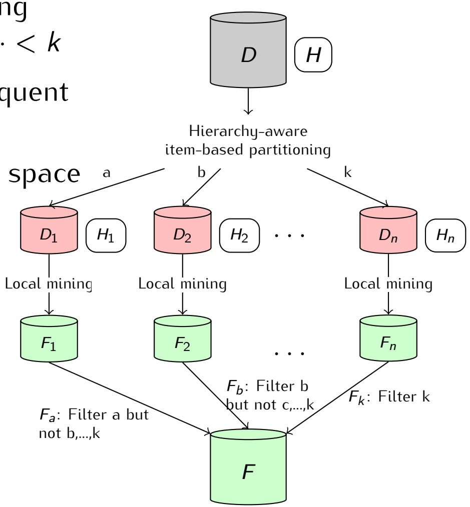
Item-based Partitioning

- Items are ordered by decreasing frequency, e.g., $a < b < c < \cdots < k$
- Create a partition for each frequent item called **pivot item**
- Key idea: partition the output space
  - $$\underbrace{\underbrace{\underbrace{\underbrace{a}_{F_a} < b}_{F_b} < c}_{F_c} < \cdots < k}_{F_k}$$



Item-based Partitioning

- Items are ordered by decreasing frequency, e.g., $a < b < c < \cdots < k$
- Create a partition for each frequent item called **pivot item**
- Key idea: partition the output space
  - $$\underbrace{\underbrace{\underbrace{\underbrace{a}_{F_a} < b}_{F_b} < c}_{F_c} < \cdots < k}_{F_k}$$
- Rewrite D for each pivot item
  - Reduces communication
  - Reduces computation
  - Reduces skew



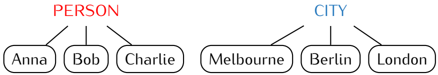
Item-based Partitioning

Example ($\sigma = 2, \gamma = 3, \lambda = 4$)

S_1: Anna lives in Melbourne

S_2: Bob lives in the city of Berlin

S_3: Charlie likes London



Semantic hierarchy

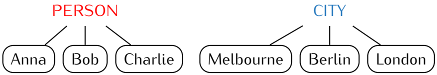
Item-based Partitioning

Example ($\sigma = 2, \gamma = 3, \lambda = 4$)

S_1: Anna lives in Melbourne

S_2: Bob lives in the city of Berlin

S_3: Charlie likes London



Semantic hierarchy

- PERSON < CITY < in < lives

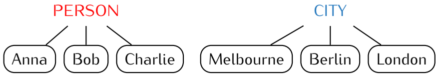
Item-based Partitioning

Example ($\sigma = 2, \gamma = 3, \lambda = 4$)

S_1: Anna lives in Melbourne

S_2: Bob lives in the city of Berlin

S_3: Charlie likes London



Semantic hierarchy

- PERSON < CITY < in < lives

PERSON : 3

PERSON

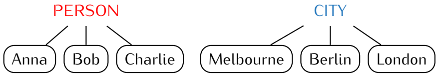
Item-based Partitioning

Example ($\sigma = 2, \gamma = 3, \lambda = 4$)

S_1: Anna lives in Melbourne

S_2: Bob lives in the city of Berlin

S_3: Charlie likes London



Semantic hierarchy

- PERSON < CITY < in < lives

PERSON	: 3

PERSON

PERSON $\_^2$ CITY	: 1
PERSON _ CITY	: 1

CITY

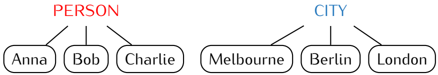
Item-based Partitioning

Example ($\sigma = 2, \gamma = 3, \lambda = 4$)

S_1: Anna lives in Melbourne

S_2: Bob lives in the city of Berlin

S_3: Charlie likes London



Semantic hierarchy

PERSON : 3

PERSON

PERSON $\_^{2}$ CITY : 1
PERSON _ CITY : 1

CITY

PERSON _ in CITY : 1
PERSON _ in $\_^{3}$ CITY : 1

in

- PERSON < CITY < in < lives

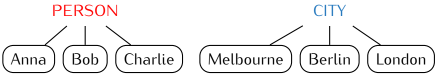
Item-based Partitioning

Example ($\sigma = 2, \gamma = 3, \lambda = 4$)

S_1: Anna lives in Melbourne

S_2: Bob lives in the city of Berlin

S_3: Charlie likes London



Semantic hierarchy

- PERSON < CITY < in < lives

PERSON : 3

PERSON

PERSON $\_^{2}$ CITY : 1
PERSON _ CITY : 1

CITY

PERSON _ in CITY : 1
PERSON _ in $\_^{3}$ CITY : 1

in

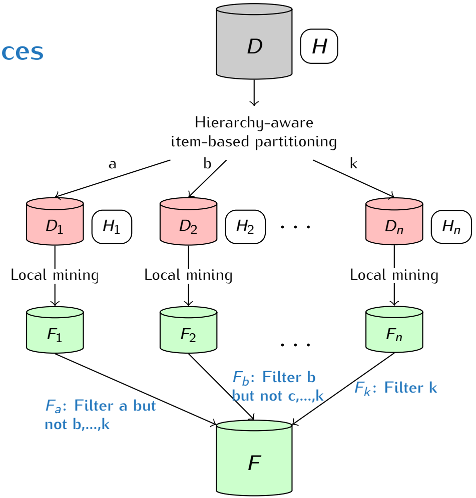
PERSON lives in CITY : 1
PERSON lives in $\_^{3}$ CITY : 1

lives

Outline

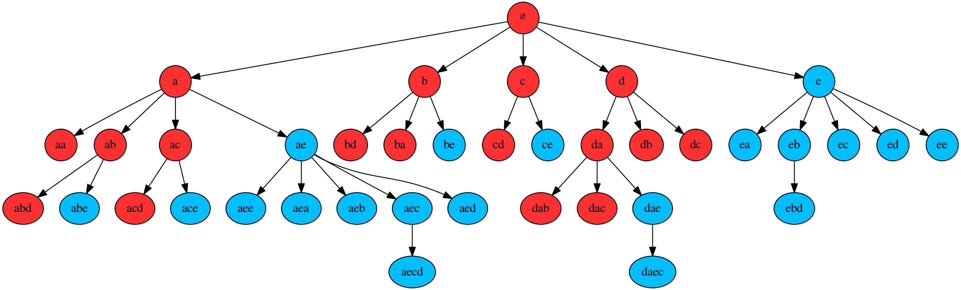
Local Mining

- Goal: Compute **pivot sequences**
- $\underbrace{\underbrace{\underbrace{\underbrace{a}_{F_a} < b}_{F_b} < c}_{F_c} < \cdots < k}_{F_k}$



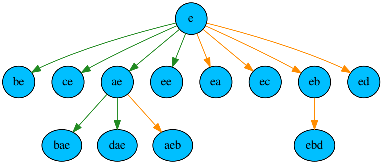
Local Mining

- Traditional approach
  - Use any mining algorithm
    (based on depth-first or breadth-first search)
  - Filter out non-pivot sequences
- Example: depth-first search
  - Pivot item: *e*



Local Mining

- Pivot sequence miner (PSM)
  - Mines only pivot sequences
    - Start with the pivot item
    - **Right expansions**
    - **Left expansions**
  - Optimized search space exploration
- Example: PSM search space
  - Pivot item: *e*

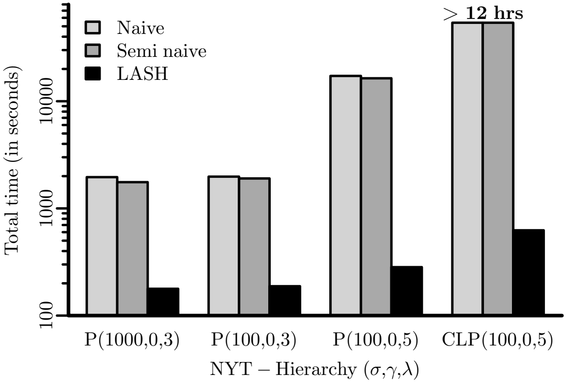


Outline

Overall Runtime

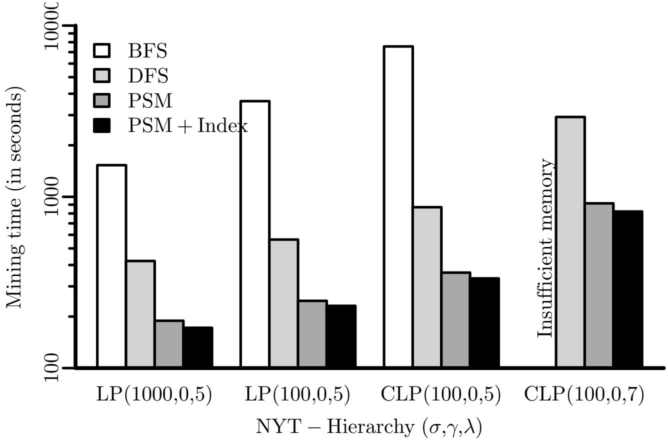
The New York Times Corpus

- ∼50M sequences, ∼1B items of which ∼2.7M distinct
- Syntactic hierarchy (word → lowercase → lemma → POS tag)
- 10 node hadoop cluster



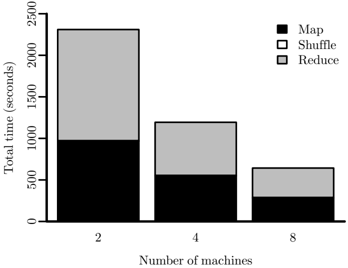
LASH is multiple orders of magnitude faster

Local Mining

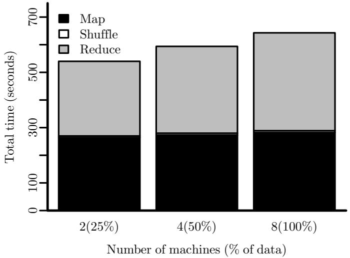


PSM is effective, more than 3× faster

Scalability



(a) Strong Scalability



(b) Weak Scalability

Good strong and weak scalability

Outline

Summary and Contributions

- Sequence mining with hierarchies is an important problem
  - Enables mining non-trivial patterns

Summary and Contributions

- Sequence mining with hierarchies is an important problem
  - Enables mining non-trivial patterns
- LASH: LArge-scale Sequence mining with Hierarchies
  - Novel hierarchy-aware form of item-based partitioning
  - Efficient special-purpose algorithm for mining each partition

Summary and Contributions

- Sequence mining with hierarchies is an important problem
  - Enables mining non-trivial patterns
- LASH: LArge-scale Sequence mining with Hierarchies
  - Novel hierarchy-aware form of item-based partitioning
  - Efficient special-purpose algorithm for mining each partition
- First distributed, scalable algorithm to mine such sequences

Summary and Contributions

- Sequence mining with hierarchies is an important problem
  - Enables mining non-trivial patterns
- LASH: LArge-scale Sequence mining with Hierarchies
  - Novel hierarchy-aware form of item-based partitioning
  - Efficient special-purpose algorithm for mining each partition
- First distributed, scalable algorithm to mine such sequences

Thank you!
Questions? / Comments