Zips: Mining Compressing Sequential Patterns in Streams

Hoang Thanh Lam, Toon Calders, Jie Yang
Fabian Moerchen and Dmitriy Fradkin

Agenda

- The problem
- Pattern explosion issue in frequent pattern mining
- Mining compressing patterns: the state of the art solution to the pattern explosion issue
- Mining Compressing patterns in streams
- Scalability
- Take away messages

Agenda

- The Problem
- Pattern explosion issue in frequent pattern mining
- Mining compressing patterns: a solution to the pattern explosion issue
- Mining Compressing patterns in streams
- Experiments
- Take away messages

The Problem

- Given a stream of sequences, e.g. tweet stream, at any given time point return the most important sequential patterns in the stream
- The set of patterns must be non-redundant and meaningful.

Patterns on Tweets Stream

Zips algorithm

Agenda

- The Problem
- Pattern explosion issue in frequent pattern mining
- Mining compressing patterns: a solution to the pattern explosion issue
- Mining Compressing patterns in streams
- Scalability
- Take away messages

Pattern explosion issue

- Exponential number of frequent patterns, causing redundancy issues
  - Reason: if a set is frequent, all of its subsets are also frequent
- The top most frequent patterns are usually trivial or meaningless.
  - Reason: frequent patterns are combinations of frequent items but unrelated to each other

Pattern explosion issue

Pattern	Support	Pattern	Support
algorithm algorithm	0.376	method method	0.250
learn learn	0.362	algorithm result	0.247
learn algorithm	0.356	Data set	0.244
algorithm learn	0.288	learn learn learn	0.241
data data	0.284	learn problem	0.239
learn data	0.263	learn method	0.229
model model	0.260	algorithm data	0.229
problem problem	0.258	learn set	0.228
learn result	0.255	problem learn	0.227
problem algorithm	0.251	algorithm algorithm algorithm	0.222

The most frequent closed sequential patterns mined from 7000 abstracts of articles from the Journal of Machine Learning Research (JMLR)

Agenda

- The Problem
- Pattern explosion issue in frequent pattern mining
- Mining compressing patterns: a solution to the pattern explosion issue
- Mining Compressing patterns in streams
- Scalability
- Take away messages

Mining compressing sequential patterns

- The key idea is based on the Minimum Description Length Principle (MDL): the model that describes the data in the shortest way is the best model.

Pattern mining using MDL (Krimp algorithm Siebes et al. SDM 2006)

- Model: the set of patterns M.
- Encoding: compress the data D with the help of model M
- Data description length:

$$L_M(D) = L(M) + L(D|M)$$

- Find the model M* that minimizes the data description length:

$$M^* = argmin_M L_M(D)$$

Pattern mining using MDL, how it works?

- Build a dictionary (a set of patterns)
- Encode the data given the dictionary (replace occurrences of patterns in the data by pointers to the dictionary)
- Pointers are represented by binary codewords. Shorter codewords are assigned to pointers with more usage.

Pattern mining using MDL, how it works?

D		
word	Codeword $C(w)$	usage
a		0
b		0
c		0
d		1
e		1
abc		3

Encoded sequence

Codeword length is proportional to *–log(sum/usage)*, i.e
Shorter codeword is assigned to more frequently used patterns

Pattern mining using MDL, how hard?

- Finding an optimal dictionary and an optimal encoding given a dictionary is NP-hard (Lam et al. SDM 2012, SADM 2013)
- The state of the art approaches are based on greedy algorithm: grow the dictionary greedily (step by step add to the dictionary the next pattern that results in the most compression benefit).

It solved the pattern explosion issues

Method	Patterns			
SQS (Vreeken et al. KDD 2012)	**support vector machin** **machin learn** **state art** **data set** **bayesian network**	**larg scale** nearest neighbor decis tree **neural network** cross valid	featur select graphic model **real world** **high dimension** mutual inform	sampl size learn algorithm princip compon analysi logist regress model select
GOKRIMP (Lam et al. SDM 2012, SADM Journal 2013)	**support vector machin** **real world** **machin learn** **data set** **bayesian network**	**state art** **high dimension** reproduc hilbert space **larg scale** independ compon analysi	**neural network** experiment result sampl size supervis learn support vector	well known special case solv problem signific improv object function
Zips (This work)	**support vector machin** **data set** **real world** learn algorithm **state art**	featur select **machine learn** **bayesian network** model select optim problem	**high dimension** paper propose graphic model **larg scale** result show	cross valid decis tree **neutral network** well known hilbert space

Meaningful patterns are extracted by the MDL based pattern mining approaches (SQS by Vreeken et al. KDD 2012, GoKrimp by Lam et al. SDM 2011 and Zips-this work) from the JMLR dataset

Agenda

- The Problem
- Pattern explosion issue in frequent pattern mining
- Mining compressing patterns: a solution to the pattern explosion issue
- Mining Compressing patterns in streams
- Scalability
- Take away messages

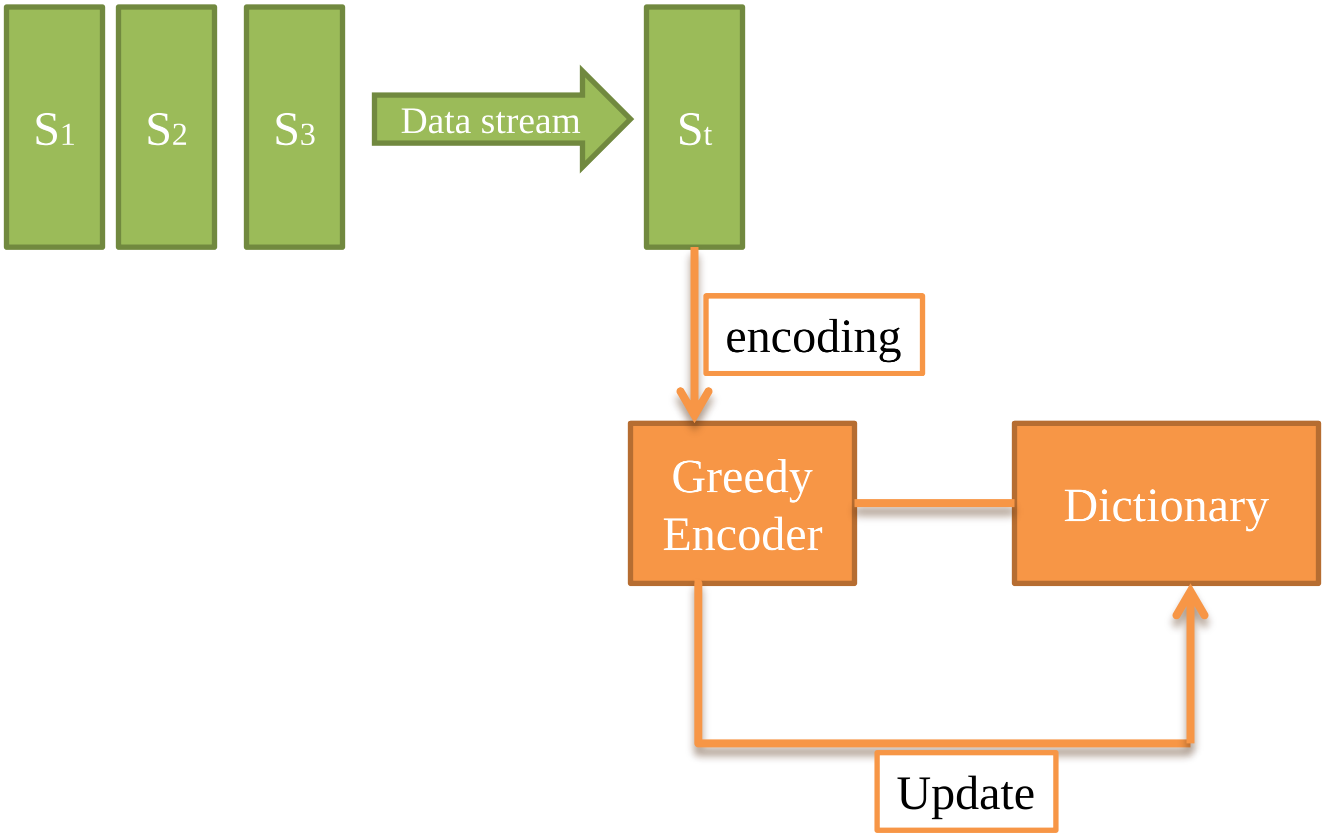
Data streams

- Infinite stream of sequences, for example, tweets stream, search engine query log, machine message log etc.

Compressing sequential patterns mining in Data streams

- Challenges:
  1. Single pass constraint
  2. Memory constraint
  3. High speed updates

Our Algorithm

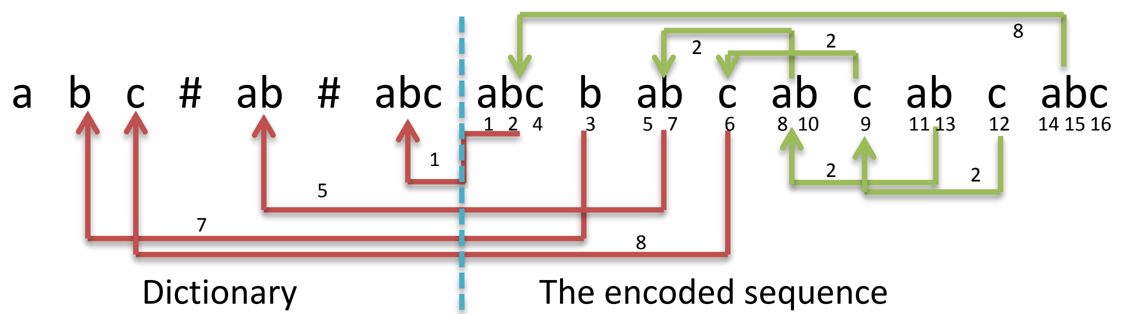


Key technical contributions

- A new encoding called reference encoding for streams, usage is approximately updated in a single pass through data
- Space saving algorithm to keep the size of the dictionary always below a predefined threshold

Reference encoding

- Instead of using pointers to dictionary, use references to the most recently encoded instance of the pattern.



Reference encoding

- Benefits of using reference encoding:
  1. Efficient dictionary update: no need to recalculate usages for all the words in the dictionary per update
  2. Theoretical guarantee: in average, the codeword length of references provably converges to the codeword length assigned based on usage with assumption that the encoded instances of patterns are independent.

Space saving algorithm

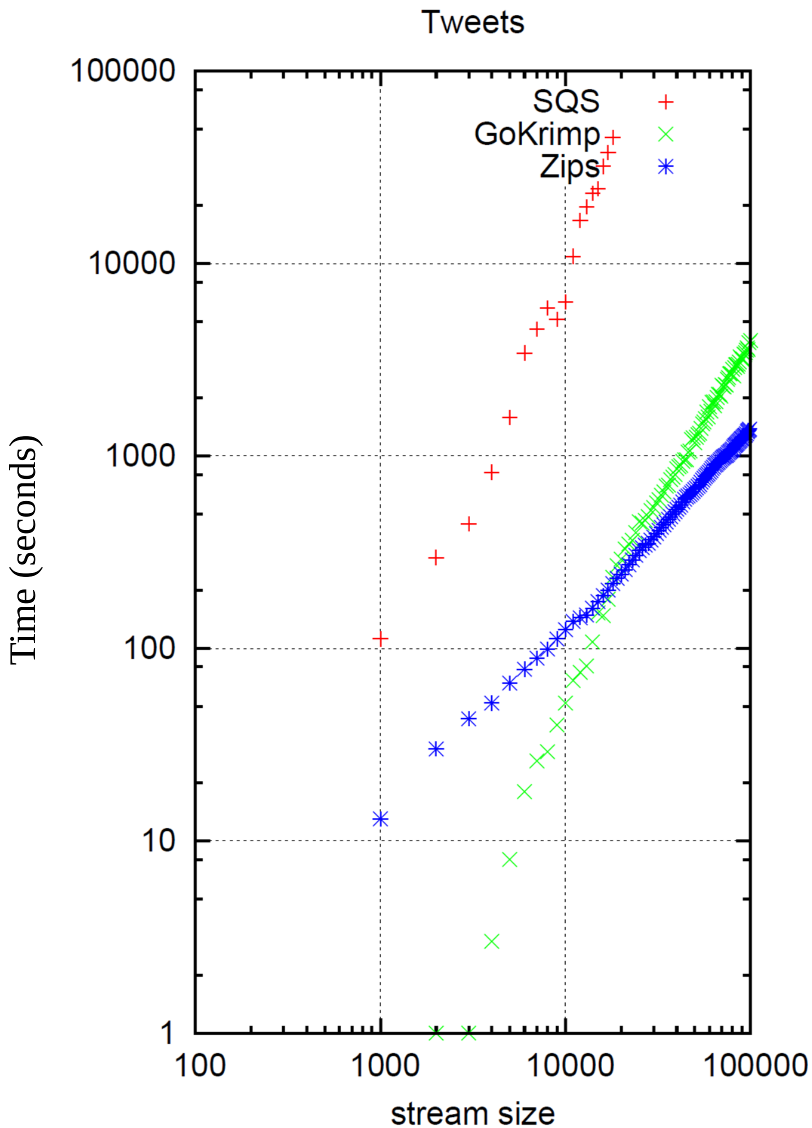
- Dictionary extension: add to the dictionary a new candidate pattern by appending the current pattern with the following item.
- Dictionary is full: replace the pattern with the least overestimated usage by the new candidate patterns.

Agenda

- The Problem
- Pattern explosion issue in frequent pattern mining
- Mining compressing patterns: a solution to the pattern explosion issue
- Mining Compressing patterns in streams
- Scalability
- Take away messages

Scalability

- Zips scales linearly with the size of the tweet stream
- GoKrimp and SQS scale quadradtically



Agenda

- The Problem
- Pattern explosion issue in frequent pattern mining
- Mining compressing patterns: a solution to the pattern explosion issue
- Mining Compressing patterns in streams
- Scalability
- Take away messages

Take away messages

- We solved the mining non-redundant sequential patterns problem in streams
- The quality of the patterns extracted by our solution is similar to patterns extracted by the state of the art algorithms.
- Our solution scales linearly with the size of data while the state of the art algorithms do not

Pattern Visualization on Stream

- We used wordcloud tool to visualize patterns extracted from sliding windows. (demo)
- Wordcloud shows important patterns in each snapshot but doesn't show how importance score of patterns change overtime.
- Need a better visualization tool for stream!