Jihad Mohamad Alja'am
Abdulmotaleb El Saddik
Abdul Hamid Sadka *Editors*

Recent Trends in Computer Applications

Best Studies from the 2017 International Conference on Computer and Applications, Dubai, UAE

Springer

Recent Trends in Computer Applications

Jihad Mohamad Alja'am •
Abdulmotaleb El Saddik • Abdul Hamid Sadka
Editors

Recent Trends in Computer Applications

Best Studies from the 2017 International Conference on Computer and Applications, Dubai, UAE

Springer

Editors
Jihad Mohamad Alja'am
Computer Science and Engineering
Qatar University
Doha, Qatar

Abdulmotaleb El Saddik
Faculty of Engineering
University of Ottawa
Ottawa
Ontario, Canada

Abdul Hamid Sadka
Electronic and Computer Engineering
Brunel University London
Uxbridge, United Kingdom

ISBN 978-3-319-89913-8 ISBN 978-3-319-89914-5 (eBook)
https://doi.org/10.1007/978-3-319-89914-5

Library of Congress Control Number: 2018958782

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword

By the turn of the second millennium, it became clear that computers (and more broadly intelligent machines) are becoming the focus of science and technology for the next few decades to come. This book introduces the reader to the realm of the most recent trends in the area of computer applications, with a special focus on sustainable development, marking this important trend during the first decades of the third millennium. The broad scope of the book is by design as the editors and authors introduce a wide scope of application fields where modern computing brought about several paradigm shifts in the way data is analysed, managed and visualised.

Already a decade ago, the notion of Big Data was introduced, and since then new scientific and technical challenges were formulated and efficient solutions have been proposed. Big Data was later cast in the framework of decision-making environments, in which theories, algorithms, methods and systems have been developed to efficiently map data into decisions. Data-driven and data-intensive computer applications have since been developed in a number of areas, including, but not limited to, media (both audio and visual), healthcare, robotics, security, web applications and web interfaces. Conceptually, data handling strategy can conveniently be presented as a three-layered scheme, in which the first layer interfaces with raw data (computer-generated, time series, sensor data, etc.) and offers various ways to represent, clean, abstract and possibly augment the source data. The second layer hosts methods and algorithms for analytics, management and visualisation of the processed data. And finally, the third layer links the results of the second layer to a specific application, that is, it interfaces with the real-world application domain. Most of the contributions in this book cover one or more of these layers targeting a specific application domain. As the amount of data keeps increasing exponentially and the demand of split (real-time) decision is becoming more imminent, the key challenges we are facing today include scalability, efficiency and real-time performance.

The book is recommended to the tech-savvy managers as well as engineers, technicians and researchers in various fields of computer applications. It is rather seldom we come across a reference where such an overwhelming amount of information regarding diverse application fields has been gathered in a single volume. The book requires common mathematical and computer science knowledge acquired by most university college degrees and, therefore, it is easy to read and grasp the main concepts which are well illustrated throughout the chapters of the book.

NSF I/UCRC Center for Visual
and Decision Informatics, TUT-Site
Laboratory of Signal Processing
Tampere University of Technology Tampere, Finland

Moncef Gabbouj

Preface

This book consists of an agglomeration of know-how and recent research findings imparted by a broad range of international scientists within the field of information and communication technologies. The book recognises computers potentially as data-generating machines and computer applications as platforms for the acquisition, analysis, processing, management and visualisation of data in its multiple forms, scales (i.e. volumes), complexities and digital representations. While the editors realise the breadth and diversity of computer applications, and hence the corresponding data-handling strategies employed therein, the essence of the book focuses mainly on three key data-driven classes of technologies, namely data analytics, data management and data visualisation.

In data analytics and processing, the book presents an authoritative set of chapters addressing various challenges commonly encountered in computer-based applications and systems, such as segmentation, detection, classification, recognition, etc., in the light of vision-based but also multimodal scenarios. In image segmentation, for instance, one chapter addresses the unsupervised segmentation of images using graph-based community detection. In particular, an overview of sequential mining algorithms and their extensions is presented in one chapter, while image/video classification is addressed using multimodal techniques in another chapter and using Gabor filters in yet another. An analysis of the current and future directions of object detection based on convolutional neural networks is also featured. Hand detection and gesture recognition within the context of human–computer interaction is addressed in a chapter that focuses on translating recognised hand gestures into functional ones to enable the real-time manipulation of a 2D image. The book also looks into the compression aspects of computer applications with multiview video codecs in perspective. In particular, one chapter addresses the multiview extensions of two contemporary video coding standards and provides a comparative analysis of their performance in terms of quality and compression efficiency.

In data management, the authors' contributions place a particular emphasis on the security aspects of data in networked computer applications which utilise cloud computing technologies. One chapter looks at utilising a combination of data encryption algorithms and a distributed system to improve data confidentiality for an acceptable overhead performance. Another chapter considers the malware detection algorithms and argues the complexity and computationally intensive nature of the process of identifying malicious codes in files or network traffic. It goes further to propose a novel hybrid solution that leverages the CPU/GPU computing capabilities for improving the performance and reducing the power consumption of string matching algorithms on devices such as laptops for instance. Furthermore, the security aspects of Software-Defined Networking (SDN) are examined from the standpoint of Distributed Denial of Service (DDoS) attacks. The chapter presents a controller placement model that helps reduce the impact of DDoS attacks and hence make SDN more secure and resilient. One interesting computer application considered in the book is the control of a permanent-magnet DC motor without the prior knowledge of its parameters. Another chapter dedicates specific attention to designing a new hardware/software platform that enables the real-time provision of all the parameters required for the control of the DC motor.

In data access and visualisation, the book presents a series of chapters that are concerned with providing a user-centric approach to multimedia applications and services, particularly in e-commerce, future Tactile Internet, search and retrieval as well as language recognition scenarios. One chapter reviews the state of the art in user-centric multimodal systems and presents a vision towards the realisation of an immersive, interactive and collaborative framework for the Internet of Multimodal Things (IoMT) system. Another chapter considers an automated approach to the optimisation of Web interfaces for e-commerce. The chapter emphasises primarily the vital role of User Experience (UX) and Customer Experience (CX) principles in the provision of any web-based service, application or product. The book embodies a chapter that features the design, development and evaluation of a web-based Arabic multimedia search engine that is based on a language transcriber. In order to enable an efficient and user-friendly human–computer interaction, a chapter focuses on the review and analysis of specific text-to-picture systems and approaches to facilitate education. Last but not least, one of the chapters explores both person-dependent and person-independent Arabic speech recognition systems and examines how hidden Markov models can be specifically exploited for the recognition of Arabic, rather than English, words.

This book offers the readers with the unique dual benefit of gaining a meticulous analysis of current technology trends in computer applications and simultaneously benefiting from a rich display of recent experimental research findings in a rather diverse and prolific technological field. While the book is inherently diverse in its scope and coverage, addressing a broad spectrum of technologies exploited by computer applications and systems today, the book editors are confident that this manuscript will put at the disposal of their audience, from both academia and industrial R&D sectors, a useful resource that will not only help expand the beneficiaries' knowledge base in the relevant fields but will also offer them a supportive guide that is equipped with a sufficient level of scientific originality, depth and rigour into a cluster of technological trends and most recent research developments in multimedia data handling and manipulation, with computer applications in perspective.

Jihad Mohamad Alja'am
Abdulmotaleb El Saddik
Abdul Hamid Sadka

Acknowledgements

This book would not have been possible without the hard work of all the authors. I am especially indebted to Prof. Abdul Hamid Sadka and Prof. Abdulmotaleb El Saddik who worked hard to select the best submitted chapters to be included in this book. I am grateful to all reviewers who spent a lot of time in the peer-review process to improve the quality of the accepted chapters. Each reviewer has provided excellent and professional scientific guidance for the authors to refine their works to meet the scientific standard. I am indebted to Dr. Nabil Khelifi (Senior Editor—Springer) who trusted me to lead this project successfully. Many thanks also for Reyhaneh Majidi (Springer DE) who has always been very supportive. Finally, I really appreciate the hard work of my assistant Dana Bandock who allocated an important part of her time in the management for the successful completion of this book.

Department of Computer Science and Engineering
College of Engineering
Qatar University
Doha, Qatar

Jihad Mohamad Alja'am

Contents

Part I
Data Analytics and Processing (Including Classification, Compression, Segmentation, Mining, Detection and Recognition etc.)

Overview on Sequential Mining Algorithms and Their Extensions

Carine Bou Rjeily, Georges Badr, Amir Hajjam Al Hassani, and Emmanuel Andres

1 Introduction

Interesting sequential patterns (SPs) in a sequence database are extracted using Sequential Pattern Mining algorithms. These patterns help in analyzing data and obtaining interesting and valuable knowledge from large amounts of data. Other techniques including Sequence Prediction and Sequential Rule Mining are also used nowadays for decision-making purposes. The main idea is to extract frequent subsequences, called patterns, from a massive amount of collected data and understand the relation(s) between these patterns. Many sectors are interested in these techniques. For example, analyzing customers' purchases to improve marketing strategy: Let's say a customer buys a camera and a lens. The next time he comes, he buys a tripod. That information could be used to predict customers' needs by understanding their interests. The company may then offer a tripod or a discount when buying a camera and a lens. Nowadays, Sequential Pattern Mining algorithms play an important role in the medical domain, for the notion of time is important in analyzing data related to patients or hospitals. Novel applications are based on sequential mining for decision-making in the medical field, such as in [1–4]. Sequence Prediction was also used to predict heart failure in [5, 6].

C. Bou Rjeily · A. H. Al Hassani
Nanomedicine Lab, Université de Bourgogne Franche – Comté, Belfort, France
e-mail: carine.bourjeily@utbm.fr; amir.hajjam-el-hassani@utbm.fr

G. Badr (✉)
TICKET Lab, Antonine University, Baabda, Lebanon
e-mail: georges.badr@ua.edu.lb

E. Andres
Université de Strasbourg, Centre Hospitalier Universitaire, Strasbourg, France
e-mail: emmanuel.andres@chru-strasbourg.fr

J. M. Alja'am et al. (eds.), *Recent Trends in Computer Applications*,
https://doi.org/10.1007/978-3-319-89914-5_1

The first part of this chapter defines important terms and notations in the field. The second shows a survey on the most important and recent sequential mining algorithms according to a clear classification. Lastly, the chapter concludes with a classification tree showing the main categories of the algorithms and their extensions. It is important to know that this chapter provides the essential definitions and functionalities of the algorithms. Knowing the appropriate outputs of the algorithms will help the user in choosing the most efficient one for his/her studies.

2 Important Terms and Notations

Before presenting the algorithms and their classification, it is important to define some basic terms used in Sequential Pattern Mining in order to understand the mining process. These terms are commonly used in data mining processes and especially in Sequential Pattern Mining.

1. An item is an entity that can have multiple attributes: date, size, color, and so on.
2. $I = \{i_1, \ldots, i_n\}$ is a nonempty set of items. A k-itemset is an itemset with k items.
3. A sequence "S" is an ordered list of itemsets. An itemset X_y in a sequence, with $1 \leq y \leq L$, is called a transaction. L denotes the length of the sequence, which refers to the number of its transactions. $S = \{(a,b); (b,c); (e,d)\}$, which means that the items a and b are occurring together in the same time, while the items b and c are occurring together although in the same time but after a and b occur together and so on.
4. A sequential database (SDB) is a list of sequences with a sequence ID (SID) (cf. Table 1).
5. A sequence β can have a subsequence α, making β a super-sequence of α.
6. A sequential rule r, denoted $X \rightarrow Y$, is a relationship between two unordered itemsets $X, Y \subseteq I$, where $X \cap Y = \varnothing$. $X \rightarrow Y$ means that if items of X appear in a sequence, items of Y will also occur in the same sequence.
7. The support of a rule r in a sequence database SDB is defined as the number of sequences that contains $X \cup Y$ divided by the number of sequences in the database:

$$supSDB(r) = \frac{|\{s; s \in SDB \wedge r \wedge s\}|}{|SDB|}$$

Table 1 A sequence database

SID	Sequence
1	$\langle \{a, b\}, \{c\}, \{f, g\}, \{g\}, \{e\} \rangle$
2	$\langle \{a, d\}, \{c\}, \{b\}, \{a, b, e, f\} \rangle$
3	$\langle \{a\}, \{b\}, \{f\}, \{e\} \rangle$
4	$\langle \{b\}, \{f, g\} \rangle$

8. The confidence of a rule r in a sequence database SDB is defined as the number of sequences that contains , divided by the number of sequences that contains X:

$$conf\,SDB(r) = \frac{|\{s; s \in SDB \wedge r \vee s\}|}{|SDB|}$$

9. A rule r is a frequent sequential rule iff *supSDB*(r) $\geq$ *minsup*, with *minsup* $\in [0, 1]$ being a threshold set by the user.
10. A rule r is a valid sequential rule iff it is frequent and *confSDB*(r) $\geq$ *minconf*, with *minconf* $\in [0, 1]$ being a threshold set by the user.
11. Apriori-based [7]: many mining algorithms are based on this technique. The main idea is to create a list of the most frequent items with respect to *minsup* and *minconf*. The list is increased progressively considering the support and the confidence.
12. Sequential Rule Mining is to find all frequent and valid sequential rules in an SDB [8].
13. Pattern Growth [9] is a method for extracting frequent sequences by partitioning the search space and then saving the frequent itemsets using a tree structure. Extraction is done by concatenating to the processed sequence (called prefix sequence) frequent items with respect to its prefix sequence. This method can be seen as depth-first traversal algorithm and eliminates the necessity to repetitively scan all of the SDB.
14. Searching processes:

    – Depth-First Search (DFS) is a searching process that traverses or searches tree or graph data structures. A node in the graph or tree is considered as the root where the search begins. In case of graph, some arbitrary nodes are selected as the root and explored as far as possible along each branch before backtracking.
    – Breadth-First Search (BFS) is a searching process for searching in trees or graph structures. It starts at the root like (DFS) and explores the neighbor nodes first, before exploring the next-level neighbors.

    Let $\beta = \langle\, \beta_1 \ \ldots \ \beta_\mathrm{n} \,\rangle$ and $\alpha = \langle\, \alpha_1 \ \ldots \ \alpha_\mathrm{m} \,\rangle$ be two sequences where $m \leq n$.
15. Sequence α is called the prefix of β iff $\forall$i $\in [1 \ldots m]$, $\alpha_i = \beta_i$.
16. Sequence $\beta = \langle\, \beta_1 \ \ldots \ \beta_\mathrm{n} \,\rangle$ is called the projection of some sequence S with regards to α, iif:

    – $\beta \preceq s$
    – α is a prefix of β
    – There exists no proper super-sequence β' of β such that $\beta' \preceq s$ and β' also has a prefix

17. Sequence $\gamma = \langle\, \beta_{m+1} \ \ldots \ \beta_n \,\rangle$ is called the suffix of s with regard to α. β is then the concatenation of α and γ.

    Let SDB be a sequence database.

Table 2 A vertical database for the sequence database of Table 1

A		B		C		D		E	
SID	Itemsets	SID	Itemsets	SID	Itemsets	SID	Itemsets	SID	Itemsets
1	1	1	1	1	2	1		1	5
2	1,4	2	3,4	2	2	2	1	2	4
3	1	3	2	3		3		3	4
4		4	1	4		4		4	

F		G	
SID	Itemsets	SID	Itemsets
1	3	1	3,4
2	4	2	
3	3	3	
4	2	4	2

Table 3 Projected database with regards to prefix "a"

$\langle a \rangle$: projected database
$\langle$ {_, b}, {c}, {f, g}, {g}, {e} $\rangle$
$\langle$ {_, d}, {c}, {b}, {a, b, e, f} $\rangle$
$\langle$ {b}, {f}, {e} $\rangle$
$\langle\ \rangle$

18. Horizontal database: each entry in a horizontal database is a sequence as shown in Table 1.
19. Vertical database: each entry represents an item and indicates the list of sequences where the item appears and the position(s) where it appears [10] (cf. Table 2).

20. Projected database: the α-projected database, denoted by $\mathrm{SDB}|_\alpha$, is the collection of suffixes of sequences in SDB with regard to prefix α. Table 3 shows an example of the projected database considering "a" as prefix.

3 Sequential Mining Algorithms

There exist many data mining techniques such as classification, clustering, association rule mining and others. This chapter focuses on sequential mining algorithms. We present the state of the art of recent algorithms, elaborating a classification based on their main objectives and principles. Thus, this classification divides the algorithms into three primary types: SP Mining, Sequential Rule Mining and Sequence Prediction. Each of these can be split into different criteria and strategies.

3.1 Sequential Pattern Mining

3.1.1 Frequent Sequential Pattern Mining

It consists of finding subsequences appearing frequently in a set of sequences called sequential pattern or frequent subsequence. The frequency of these patterns is no less than a minimum support threshold *minsup* specified by the user. The common frequent Sequential Pattern Mining algorithms are:

The Generalized Sequential Patterns (GSP) algorithm [11] is an Apriori-like method and was one of the first algorithms that studied SPs after Apriori-All. The database is scanned multiple times. The first pass determines the support of each item, which is the number of data sequences that include the item. It simply means counting the occurrences of singleton transactions (containing one element) in the given database (one scan of the whole database). After this process, nonfrequent items are removed, and each transaction consists now of its original frequent items. This result will be the input of the GSP algorithm. Like Apriori, GSP algorithm makes multiple database scans. At the first pass, all single items of length 1 sequences (1-sequences) are counted. At the second pass, frequent 1-sequences are used to define the sets of candidate 2-sequences, and another scan is made to calculate their support. Same process is used to discover the candidate 3-sequences but using frequent 2-sequences, and so on until no more frequent sequences are found. GSP algorithm is composed of two techniques:

1. Candidate Generation: Only candidates with minimum support or above are conserved until no new candidates are found. This technique generates an enormous number of candidate sequences and then tests each one with respect of the user-defined *minsup*.

   After the first scan of the database and obtaining frequent $(k-1)$-frequent sequences $F(k-1)$, a joining procedure of $F(k-1)$ with itself is made and any infrequent sequence is pruned if at least one of its subsequences is not frequent.
2. Support Counting: a hash tree-based search is used. Finally nonmaximal frequent sequences are removed.

The GSP algorithm also allows frequent sequences discovery with time constraints. It can calculate the difference between the end-time of the element just found and the start-time of the previous element. This time is user defined and called maximum and minimum gap. Furthermore, it supports the concept of a sliding window (defines the interval of time between items in the same transaction).

The Sequential PAttern Discovery using Equivalence classes, SPADE [10] is based on a vertical id-list database format in which each sequence is associated to a list of items in which it appears: each subsequence is originally associated to its occurrence list. The frequent sequences can be found by using the intersection on id-lists. The size of the id-lists is the number of sequences in which an item appears. SPADE reduces the search space by aggregating SPs into equivalent classes and thus reduces the execution time. Thereby, two k-length sequences are in the same equivalence class if they share the same k-1 length prefix.

In his first step, SPADE computes the support of length 1 sequences, and this is done in a single database scan. In its second step, SPADE computes the support of 2-sequences and this is done by transforming the vertical representation into a horizontal representation in memory. This counting process is done with one scan of data and uses a bi-dimensional matrix. The idea consists of joining $(n-1)$ sequences using their id-lists to obtain n-subsequences. If the size of id-list is greater than *minsup*, then the sequence is frequent. The algorithm can use a breadth-first or a depth-first search method for finding new sequences. The algorithm stops when no more frequent sequences are found.

Sequential PAttern Mining, SPAM [12], is a memory-based algorithm and uses vector of bytes (bitmap representation) to study the existence (1) or absence (0) of an item in a sequence after loading the database into the memory. Candidates are generated in a tree by an S-extension that adds an item in another transaction, and by an I-extension that appends the item in the same transaction. The candidates are verified by counting the bytes with a value of one with the defined *minsup*.

The algorithm is efficient for mining long sequential patterns. Depth-first search is used to generate candidate sequences, and various I-step pruning and s-Step pruning are used to reduce the search space.

The transactional data are stored using a vertical bitmap representation, which allows for efficient support counting as well as significant bitmap compression. One new feature introduced with SPAM is that it incrementally outputs new frequent itemsets in an online fashion.

The Prefix-projected Sequential Pattern Mining, known as PrefixSpan [13], is a pattern-growth-based algorithm that discovers SPs using the idea of projected database. The algorithm studies the prefix subsequences instead of exploring all the possible occurrences of frequent subsequences (refer to the definitions 16 and 17). Then, it performs a projection on their corresponding post-fix subsequences. Frequent sequences will grow by mining only local frequent patterns, showing the efficiency of this algorithm.

The Last Position Induction algorithm (LAPIN) [14] is used for the extraction of long sequences and the reduction of the search space. It uses a lexicographical tree as the search path with DFS strategy. LAPIN-LCI procedure tests each item in the local candidate list and directly decides whether the item can be added to the prefix sequence or not. It compares the item's last position with the prefix border position. The algorithm assumes that the last position of an item i is helpful to decide whether this item could be appended to a frequent sequence of length k in order to get a frequent sequence of $k + 1$ length.

The CM-SPAM and CM-SPADE [15] are extensions of the two well-known algorithms SPADE and SPAM to which is added a new structure called Co-Occurrence MAP (C-MAP). The latter is used to store co-occurrence information by dividing them into CMAPi and CMAPs substructures. The first stores the items that succeed each item by i-extension and the second stores the items that succeed each item by s-extension at least *minsup* times. Let S be the sequence $\{I_1, I_2, \ldots,$

$I_n\}$. An item k is said to succeed by i-extension to an item j in S, iff j and $k \in I_x$ for an integer x such that $1 \leq x \leq n$ and $k >_{\text{lex}} j$. An item k is said to succeed by s-extension to an item j in S, iff $j \in I_v$ and $k \in I_w$ for some integers v and w such that $1 \leq v < w \leq n$.

The i-extension of pattern P with an item x is considered nonfrequent if there exists an item i in the last itemset of P such that (i,x) is not in CMAPi. Same for the pruning of s-extension: The s-extension of a pattern P with an item x is infrequent if there exists an item i in P such that (i, x) is not in CMAPs.

3.1.2 Closed Sequential Pattern Mining

A Closed Sequential Pattern (CSP) is not necessarily included in another pattern having the same support. The set of CSPs is much smaller than the set of SPs making mining more efficient. There exists no super-pattern S′ of pattern S having the same support of S. Then S is a closed sequential pattern; in other words, Closed Pattern Mining means that for the same support the mining process will mine the longest pattern. Common Sequential Patterns algorithms are given in the following.

The CloSpan algorithm [16] is based on mining frequent closed sequences in large data sets instead of exploring all frequent sequences and is used to mine long sequences. Its main advantage is in time and space reduction. The algorithm is divided into two stages. In the first, it generates a set of all frequent sequences and eliminates the nonclosed sequences in the second. It represents data with lexicographical tree or order.

A Lexicographic Sequence Tree (LST) can be constructed as follows:

1. Each node in the tree corresponds to a sequence, and the root is a null sequence.
2. If a parent node corresponds to a sequence S_1, its child is either an itemset-extension of S_1, or a sequence-extension of S_1.
3. The left sibling is less than the right sibling in sequence lexicographic order.

The BI-Directional Extension (BIDE+) [17] is an extension of the BIDE algorithm that mines closed SPs and avoids problem of the candidate maintenance-and-test paradigm used by CloSpan. It works in a DFS manner in order to generate the frequent closed patterns and consumes less memory compared to the previous version.

The ClaSP [18] is based on the SPADE algorithm and was the first to mine closed frequent SPs in vertical databases. ClaSP has two phases: The first one generates a subset of frequent sequences called Frequent Closed Candidates (FCC), which is kept in main memory; and the second step executes a post-pruning phase to eliminate all nonclosed sequences from FCC to finally obtain exactly FCS.

CM-ClaSP [9] is an extension of ClaSP based on the new representation of data called C-MAP as discussed in CM-SPADE and CM-SPAM.

3.1.3 Maximal Sequential Pattern Mining

Sequential Pattern Mining may return too many results, making it difficult for the user to understand and analyze. Mining maximal SPs may be a solution. A Maximal SP is a pattern that is not included in another pattern. Maximal Pattern Mining algorithms are presented in the following.

The MaxSP [19] is inspired by the PrefixSpan algorithm. It is based on a pattern-growth algorithm that aims to extract maximal SPs without maintaining candidates. It has an integrated BIDE-like mechanism that checks if a pattern is maximal. MaxSp reduces the redundancy in SPs that could be time consuming and requires a lot of storage space.

The Vertical Maximal Sequence Patterns (VMSP) [20] is based on the SPAM search procedure that generates the pattern and explores candidate patterns having same prefix in a recursive manner. VMSP integrates three strategies: Efficient Filtering of Nonmaximal Patterns (EFN), Forward Maximal Extension Checking (FME) and Candidate Pruning by Co-Occurrence Map (CPC).

3.1.4 Compressing Sequential Pattern Mining

This kind of algorithm is used to reduce redundancy and thus to minimize the size of mining results.

GoKrimp and SeqKrimp [21] are two compressing SPs mining algorithms, based on the Krimp algorithm. They explore directly compressing patterns and avoid the resource-consuming candidate generation. SeqKrimp uses a frequent closed SPs mining algorithm to generate a set of candidate patterns. It gets the candidate pattern set and returns a good subset of compressing patterns, then greedily calculates the benefits of adding/extending a given pattern from the candidates. This procedure is repeated until no more useful patterns can be added. GoKrimp uses the same procedures but is an ameliorated version of SeqKrimp. It searches for a set of sequential patterns that compresses the data most based on the minimum description length principle; informally, the best model is the one that compresses the data the most. What differentiates GoKrimp is that it is parameter free. Users are not supposed to set a minimum support, which is a difficult decision in some cases. A dependency test is provided to consider only related patterns to extend a given pattern. This technique aims to avoid the excessive tests of all possible extensions and makes the GoKrimp faster than SeqKrimp.

3.1.5 Top-K Sequential Pattern Mining

In SP mining algorithm, tuning the *minsup* parameter to get enough patterns is a difficult and time-consuming task. To remedy this issue, Top-K Sequential Pattern mining algorithms were implemented to return k SPs.

TSP (Top-K Closed Sequential Patterns) [22] uses the concept of pattern-growth and projection-based SP mining of PrefixSpan algorithm, and then performs a multi-pass mining to find and grow patterns. After closed pattern verification phase, the algorithm applies the minimum length constraint verification, which reduces the search space.

TKS (Top-K Sequential Patterns) [23] uses a vertical bitmap database representation. It adapts the SPAM search procedure to explore the search space of patterns to transform it to a Top-K algorithm. Then, TSK extends the most promising patterns, meaning that it finds patterns with high support in an early stage and discards infrequent items. Finally, the algorithm uses a PMAP (Precedence MAP) data structure to prune the search space.

3.2 Sequential Rules Mining

3.2.1 Sequential Rules

Mining frequent patterns is not sufficient in decision-making. Sequential rules mining and sequence prediction (next section) are necessary. A sequential rule indicates that if some item(s) occur in a sequence, some other item(s) are likely to occur afterward with a given confidence or probability. Common Sequential Rules mining algorithms are presented in the following.

CMDeo [24] was first designed to explore rules in a single sequence. It explores the search space in BFS and extracts all valid rules of size 1*1 respecting minimum support and confidence. Similarly to Apriori, CMDeo generates a huge amount of valid rules by applying a left and a right expansion.

RuleGrowth [25] explores sequential rules for several sequences and not only for one. It is based on the pattern-growth approach in finding the sequential relations that explores rules between two items and expands them left and right.

CMRules [8] is an alternative of CMDeo. It searches for the association rules to reduce the search space, then it removes the rules that do not respect minimum support and confidence. Therefore, it could be used to discover both sequential rules and association rules at the same time.

The Equivalence class-based sequential Rule Miner (ERMiner) [26] algorithm uses a vertical representation to avoid database projection. It mines the search space through equivalence classes to generate rules with the same antecedent or consequent.

3.2.2 Top-K Sequential Rules Mining

Specifying the number of sequential rules to be found may overcome the difficulty in fine-tuning sequential rules parameters like *minsup* and *minconf*. The idea of discovering Top-K nonredundant rules comes after the difficulty and the time-consuming task to tune the minimum support value by the user. Moreover, the sequential rule mining algorithms usually return a high level of redundancy. To solve both problems, the Top-K Sequential Rules Mining algorithms let the user indicate k, which is the number of rules to be discovered.

The TopSeqRule [27] was the first to address the Top-K sequential rules mining. It generates rules for several sequences based on the RuleGrowth search strategy integrated with the general process for mining Top-K patterns. To optimize results, it first generates the most promising rules and reduces the search space by increasing *minsup*.

Top-K Nonredundant Sequential Rules TNS [28] is used to discover the Top-K non-redundant sequential rules. It adopts the TopSeqRule to mine the Top-K rules and adapts it to eliminate redundancy. The algorithm gives an approximation and thus does not guarantee to retrieve the Top-K nonredundant rules. TNS has a positive integer parameter called delta to increase the result's exactitude. Results are more exact with a higher value of delta.

3.2.3 Sequential Rules with Window Size Constraints

This kind of algorithm returns all sequential rules with regard to the specified *minsup* and *minconf* appearing within a window size.

TRuleGrowth [29] is an extension of the RuleGrowth with a sliding window constraint. It is very useful in the discovery of temporal patterns (patterns that happen within a maximum time interval). TRuleGrowth allows the user to specify other optional parameters like the minimum antecedent length and the maximum consequent length. These parameters define respectively the minimum number of items appearing in the left side and the maximum number of items in the right side of a rule, knowing that the left side is the antecedent and the right side is the consequent.

3.2.4 Sequence Prediction

In many applications, it is very important to predict the next element in a sequence. Given a set of sequences, the idea is to predict the next element in a sequence S, based on a set of training sequences. Various applications use the sequence prediction algorithms. For example, one may need to know the next web page to be visited by the user, based on his/her, and/or other users' histories.

The Compact Prediction Tree (CPT) [30] is a lossless sequence prediction model that uses all information in the sequence for prediction. It consists of two phases: the training phase and the prediction phase. The first compresses the sequences in a prediction tree. A given sequence S is predicted by finding all sequences that contain the last x items from S in any order and in any position. CPT is more efficient than other existent algorithms such as Prediction by Partial Matching (PPM) [31], Dependency Graph (DG) [32] and All-K-th-Order Markov [33].

CPT+ [34] is an enhanced version of CPT where Frequent Subsequence Compression (FSC), Simple Branch Compression (SBC) and Prediction with improved Noise Reduction (PNR) strategies were added to improve prediction time and precision.

4 Conclusion

This chapter summarizes the most recent and common algorithms on the sequential mining paradigm. It does not aim to give a deep explanation about each algorithm, but it mentions its purpose and gives an idea about how it works. One should refer to the related article of each algorithm for additional details. For further explanation and ease of understanding, this chapter also presents a classification for the sequential mining algorithms. They are arranged by their usage. This classification was based on three main axes: frequent sequential pattern mining, sequential rules mining and sequence prediction. Important terms and notations in the data mining domain were first introduced. Then, a short definition introduced each class to let the reader have a quick idea about it. Later, the most important and recent algorithms in each axis were investigated with a brief description about their methods and implementations.

5 Discussion

The diagram in Fig. 1 consists of a classification tree containing the most recent algorithms and their extensions. This tree can help researchers in choosing the appropriate algorithm according to their needs especially when it comes to sequential pattern mining. Sequential mining is efficient for applications that are time-based or take into consideration the order of the event. Sequential mining has proven its efficiency through time in the economic field starting from GSP that analyzes the transactions of customers in order to improve the income and marketing strategies. Sequential Mining started showing its importance in medical field, making it a very promising field for researchers and programmers.

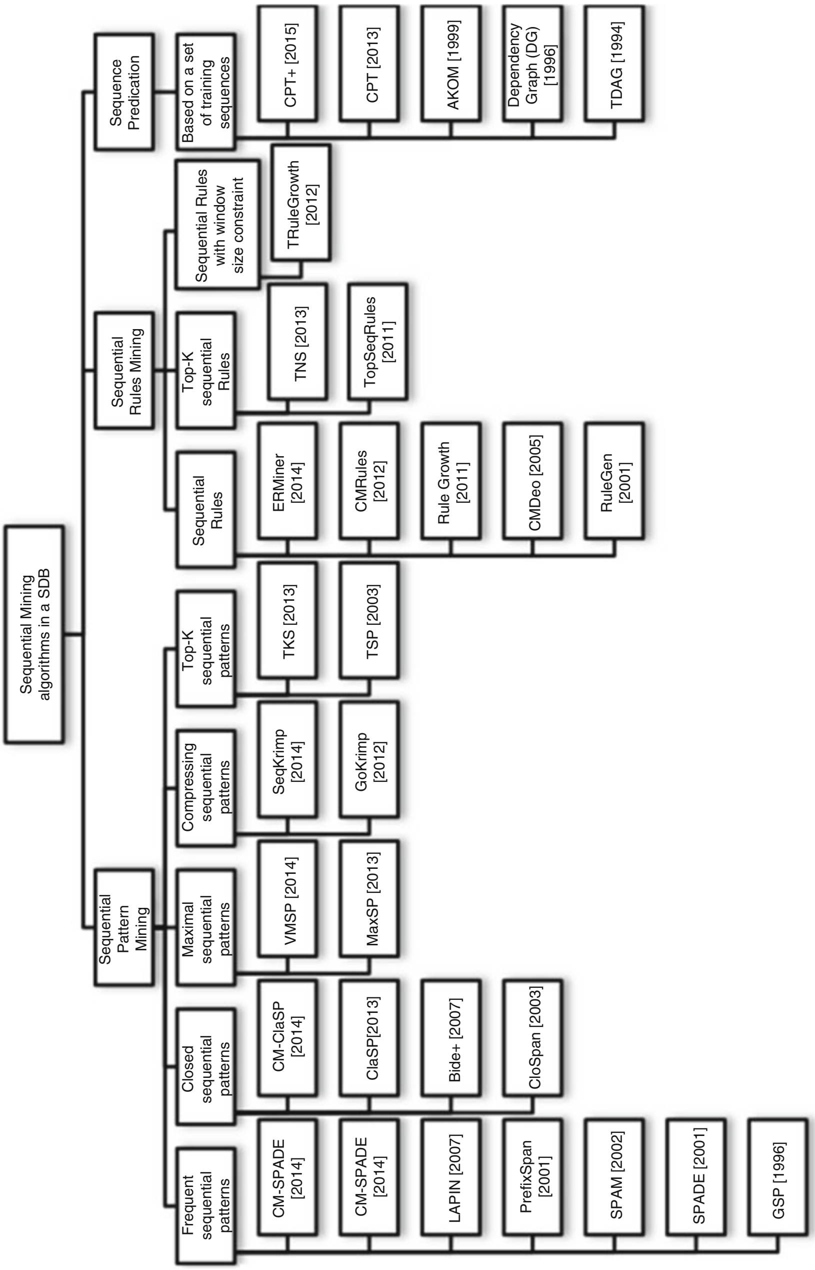

Fig. 1 Classification of sequential mining algorithms

References

1. A.P. Wright, A.T. Wright, A.B. McCoy and D.F. Sittig, "The use of sequential pattern mining to predict next prescribed medications". Journal of biomedical informatics, 53, pp.73–80, 2015.
2. G. Bruno and P. Garza, "Temporal pattern mining for medical applications". Data Mining: Foundations and Intelligent Paradigms, pp.9–18. 2012
3. K. Uragaki, T. Hosaka, Y. Arahori, M. Kushima, T. Yamazaki, K. Araki and H. Yokota, "Sequential pattern mining on electronic medical records with handling time intervals and the efficacy of medicines". In IEEE Symposium on Computers and Communication (ISCC), (pp. 20–25). IEEE. 2016.
4. K. Choi, S. Chung, H. Rhee and Y. Suh, Classification and sequential pattern analysis for improving managerial efficiency and providing better medical service in public healthcare centers. Healthcare informatics research, 16(2), pp.67–76, 2010.
5. C. Bou Rjeily, G. Badr, A. Hajjam El Hassani and E. Andres, "Sequence Prediction Algorithm for Heart Failure Prediction", International Conference e-Health, ISBN: 978-989-8533-65-4, pp.109–116, 2017.
6. C. Bou Rjeily, G. Badr, A. Hajjam El Hassani and E. Andres, "Predicting Heart Failure Class using a Sequence Prediction Algorithm", Fourth International Conference on Advances in Biomedical Engineering (ICABME), 2017
7. R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases", In ACM sigmod record(Vol. 22, No. 2, pp. 207–216). ACM, 1993.
8. P. Fournier-Viger, U. Faghihi, R. Nkambou, E. Mephu Nguifo, "CMRules: Mining Sequential Rules Common to Several Sequences. Knowledge-based Systems", Elsevier, 25(1): 63–76, 2012.
9. J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach", Data mining and knowledge discovery, 8(1), pp.53–87, 2000.
10. M.J. Zaki, "SPADE: An efficient algorithm for mining frequent sequences", Machine learning, 42(1–2), pp.31–60, 2001.
11. R. Srikant, and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements", In International Conference on Extending Database Technology (pp.1–17). Springer Berlin Heidelberg, 1996.
12. J. Ayres, J. Flannick, J. Gehrke, J. and T. Yiu, "Sequential pattern mining using a bitmap representation", In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining(pp. 429–435). ACM, 2002.
13. J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.C. Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth", In proceedings of the 17th international conference on data engineering, pp. 215–224, 2001.
14. Z. Yang, Y. Wang, and M. Kitsuregawa, M., "LAPIN: effective sequential pattern mining algorithms by last position induction for dense databases", In International Conference on Database systems for advanced applications (pp. 1020–1023). Springer Berlin Heidelberg, 2007.
15. P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas, "Fast vertical mining of sequential patterns using co-occurrence information", In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 40–52). Springer International Publishing, 2014.
16. X. Yan, J. Han, R. Afshar R., "CloSpan: Mining Closed Sequential Patterns in Large Datasets", Proceedings of the 2003 SIAM International Conference on Data Mining, 2003.
17. J. Wang, and J. Han, "BIDE: Efficient mining of frequent closed sequences", In Data Engineering, 2004. Proceedings. 20th International Conference on (pp. 79-90). IEEE, 2004.
18. A. Gomariz, M. Campos, R. Marin, and B. Goethals, "Clasp: An efficient algorithm for mining frequent closed sequences" In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 50-61). Springer Berlin Heidelberg, 2013.

19. P. Fournier-Viger, C.W. Wu, and V.S. Tseng, "Mining maximal sequential patterns without candidate maintenance", In International Conference on Advanced Data Mining and Applications (pp. 169-180). Springer Berlin Heidelberg, 2013.
20. P. Fournier-Viger, C.W. Wu, A. Gomariz, and V.S. Tseng, "VMSP: Efficient vertical mining of maximal sequential patterns", In Canadian Conference on Artificial Intelligence (pp. 83-94). Springer International Publishing, 2014.
21. H.T. Lam, F. Mörchen, D. Fradkin, and T. Calders, "Mining compressing sequential patterns", Statistical Analysis and Data Mining, 7(1), pp.34-52, 2014.
22. P. Tzvetkov, X. Yan, and J. Han, "TSP: Mining Top-k Closed Sequential Patterns", Knowledge and Information Systems, vol. 7, no. 4, pp. 438-457, 2005.
23. P. Fournier-Viger, A. Gomariz, T. Gueniche, E. Mwamikazi, and R. Thomas, "TKS: efficient mining of top-k sequential patterns", In International Conference on Advanced Data Mining and Applications (pp. 109-120). Springer Berlin Heidelberg, 2013.
24. J. Deogun, and L. Jiang, "Prediction mining–an approach to mining association rules for prediction", In International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (pp. 98-108). Springer Berlin Heidelberg, 2005.
25. P. Fournier-Viger, R. Nkambou, and V.S.M. Tseng, "RuleGrowth: mining sequential rules common to several sequences by pattern-growth", In Proceedings of the 2011 ACM symposium on applied computing (pp. 956-961), 2011.
26. P. Fournier-Viger, T. Gueniche, S. Zida, and V.S. Tseng, "ERMiner: sequential rule mining using equivalence classes", In International Symposium on Intelligent Data Analysis (pp. 108-119). Springer International Publishing, 2014.
27. P. Fournier-Viger, and V.S. Tseng, "Mining top-k sequential rules", In International Conference on Advanced Data Mining and Applications (pp. 180-194). Springer Berlin Heidelberg, 2011.
28. P. Fournier-Viger, and V. S Tseng, "TNS: mining top-k non-redundant sequential rules", In Proceedings of the 28th Annual ACM Symposium on Applied Computing, 2013.
29. P. Fournier-Viger, C.W. Wu, V.S. Tseng, and R. Nkambou, "Mining sequential rules common to several sequences with the window size constraint", In Canadian Conference on Artificial Intelligence (pp. 299-304). Springer Berlin Heidelberg, 2012.
30. T. Gueniche, P. Fournier-Viger, and V.S. Tseng, "Compact prediction tree: A lossless model for accurate sequence prediction", In International Conference on Advanced Data Mining and Applications (pp. 177-188). Springer Berlin Heidelberg, 2013.
31. J. Cleary, I. Witten, "Data compression using adaptive coding and partial string matching", IEEE Trans. on Inform. Theory, vol. 24, no. 4, pp. 413-421, 1984.
32. V. N, Padmanabhan, J.C. Mogul, "Using Prefetching to Improve World Wide Web Latency", Computer Communications, vol. 16, pp. 358-368, 1998.
33. J. Pitkow, P. Pirolli, "Mining longest repeating subsequence to predict world wide web surfing", In: USENIX Symposium on Internet Technologies and Systems, Boulder, CO, pp. 13-25, 1999.
34. T. Gueniche, P. Fournier-Viger, R. Raman, and V.S. Tseng, "CPT+: Decreasing the time/space complexity of the Compact Prediction Tree", In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 625-636). Springer International Publishing, 2015.

Object Detection Based on CNNs: Current and Future Directions

Long Chen, Abdul Hamid Sadka, Junyu Dong, and Huiyu Zhou

1 Introduction

The goal of object detection is to learn a visual model for concepts such as cars and use this model to localize these concepts in an image. As shown in Fig. 1, given an image, object detection aims at predicting the bounding box and the label of each object from the defined classes in the image. This requires the ability to robustly model invariants against illumination changes, deformations, occlusions and other intra-class variations. Among a number of vision tasks, object detection is one of the fastest moving areas due to its wide applications in surveillance [1, 2] and autonomous driving [3, 4].

L. Chen · H. Zhou
School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK
e-mail: lchen15@qub.ac.uk; h.zhou@ecit.qub.ac.uk

A. H. Sadka (✉)
Department of Electronic and Computer Engineering, Brunel University, London, UK
e-mail: abdul.sadka@brunel.ac.uk

J. Dong
Department of Computer Science and Technology, Ocean University of China, Qingdao, China
e-mail: dongjunyu@ouc.edu.cn

J. M. Alja'am et al. (eds.), *Recent Trends in Computer Applications*,
https://doi.org/10.1007/978-3-319-89914-5_2

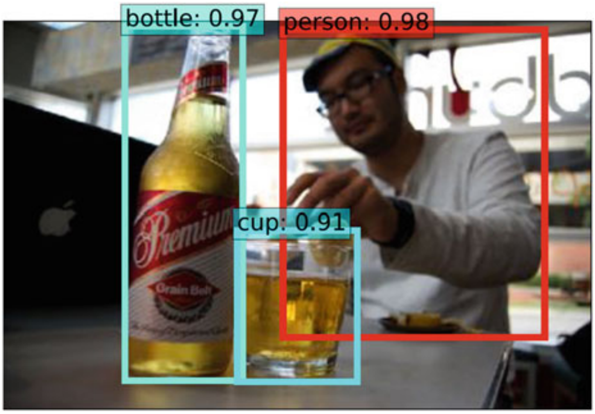

Fig. 1 Bounding boxes and labels with corresponding class probabilities predicted by detectors

2 From Handcrafted Features to Deep CNNs Methods

2.1 Handcrafted Features

Before deep CNNs, convolutional neural networks [5], were introduced, the progress on various visual recognition tasks had been considerably based on the use of handcrafted features, such as SIFT [6] and HOG [7]. Handcrafted features can be broadly divided into three categories:

1. Interest Point Detection. These methods use certain criteria to select pixels, edges and corners as well-defined local texture features. Among them, Sobel, Prewitt, Roberts, Canny and LoG (Laplacian of Gaussian) are typical edge detection operators [8–11], while Harris, FAST (Features from Accelerated Segment Test), CSS (Curvature Scale Space) and DOG (Difference of Gaussian) are typical corner detection operators [6, 12, 13]. Interest point detection methods usually have a certain geometric invariance which can be found at a small computational cost.
2. Methods based on local features. These methods mainly extract local features, which are different from global features such as colour histograms, which are ideal for dealing with partial occlusion of target objects. Commonly used local features include Scale-Invariant Feature Transform (SIFT) [6], HOG (Histogram of oriented gradient) [7], Haar-like [14] and Local Binary Pattern [15, 16]. Local features are informative, unique, with strong invariance and distinguishability. But the calculation is generally complicated, and local features are further developed to have better representations in recent years.
3. Methods based on multi-feature combination. A combination of interest point and local feature extraction methods can be used to handle the deficiency of using a single feature to represent target objects. DPM (Deformable Part-based

Model) [17] is an effective multi-feature combination model which has been widely applied to the object detection task and has achieved good performance, such as pedestrian detection [14, 16], face detection [15, 18] and human pose estimation [19]. In [20], three prohibitive steps in the cascade version of DPM were accelerated, which greatly improved the detection speed.

The characteristics of handcrafted features are largely dependent on experience and environments, where most of the test and adjustment workloads are undertaken by the user, which is time-consuming. In contrast, an important viewpoint in the deep learning theory, which has drawn much attention in recent years, is that handcrafted descriptors, as the first step in a visual system, tend to lose useful information. Directly learning task-related feature representation from raw images is more effective than handcrafted features [21].

For object detection tasks, handcrafted features based systems have become a dominant paradigm in the literature before deep CNNs were introduced. If we look at system performance on the canonical visual recognition task, PASCAL VOC object detection [22], it is acknowledged that certain progress has been made during 2010–2012, by building ensemble systems and employing variants of successful methods. Recently, Convolutional Neural Networks (CNNs) [5] have produced impressive performance improvements in many computer vision tasks since 2012, such as image classification, object detection and image segmentation. CNNs witnessed its frequent use in the 1990s (e.g., [5]), but then became less used, particularly in computer vision, with the powerful impact of support vector machines (SVMs) [23]. In 2012, Krizhevsky et al. [24] rekindled interests in CNNs by showing substantially high image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [25]. Their success resulted from training a large CNN on 1.2 million labelled images, together with a few twists on [5] (e.g., 'dropout' regularization). The significance of deep CNNs methods will be introduced in the following section.

2.2 Deep Learning Approaches

Convolutional Neural Networks [5] is the first successful method in deep learning approaches. The key difference between CNNs-based and conventional approaches is that in the former, the feature representation is learned instead of being designed by the user. These recent successes were built upon the powerful deep features that are learned from large-scale datasets, which accompany accurate annotations with the drawback that a large number of training samples are required for training the classifier. Among many variants of the CNNs-based approaches, they can be roughly divided into two streams: region proposal-based methods and proposal-free methods.