FACULTY OF ENGINEERING
AND ARCHITECTURE

Mining Subjectively Interesting Patterns in Rich Data

Junning Deng

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Computer Science Engineering

Supervisors
Prof. Tijl De Bie, PhD - Prof. Jefrey Lijffijt, PhD

Department of Electronics and Information Systems
Faculty of Engineering and Architecture, Ghent University

May 2021

GHENT
UNIVERSITY

ISBN 978-94-6355-484-8
NUR 980, 984
Wettelijk depot: D/2021/10.500/32

Members of the Examination Board

Chair

Prof. Filip De Turck, PhD, Ghent University

Other members entitled to vote

Anastasia Dimou, PhD, Ghent University
Florian Lemmerich, PhD, Rheinisch Westfälische Technische Hochschule Aachen, Germany
Prof. Femke Ongenae, PhD, Ghent University
Prof. Panagiotis Papapetrou, PhD, Stockholms universitet, Sweden
Prof. Celine Robardet, PhD, Institut National des Sciences Appliquées de Lyon, France

Supervisors

Prof. Tijl De Bie, PhD, Ghent University
Prof. Jefrey Lijffijt, PhD, Ghent University

Acknowledgements

Firstly, I would like to express my sincere gratitude to the members of the examination committee for accepting to participate in the end of my PhD journey. I appreciate their time and efforts devoted in carefully reading this thesis, and providing insightful and valuable comments.

Then a giant, gold, foil-stamped thank you to my supervisors Tijl De Bie and Jefrey Lijffijt, for the continuous support and the excellent guidance throughout my PhD journey, not only by giving good and timely advice, but also by setting best examples themselves. Especially, I thank Tijl for asking me two questions frequently: '*what do you think (or want to do)*? when I (unconsciously or consciously) rush to know the answer to a question, and '*do you like it*' when I am engaged in a new research problem. As a person who is prone to being bewildered, reckless and missing the sight of the goal, this is an important lesson for me: not only about the independent thinking, but also about the self-consciousness, paying attention to and building up my own reasoning process (still on my way of learning that). Also, I thank Jef for his magical comprehension to untangle my jumbled English and get what I want to express, for his many reassuring and encouraging words no matter when I am shivering before a climbing wall, or jittering before a challenging task, for his forthright and effective manner.

I am heavily indebted to Bo, for his extreme patience to answer my every question, no matter how silly or trivial it is, for his great generosity and passion to pour over the experience and the wisdom, and for his high working spirits that kept me motivated.

A babypink, shiny thank you to my lovely colleagues at the AIDA group (both current and former members): Florian, Robin, Paolo, Ahmad, Xi, Alex, Maryam, Maarten, Yoosof, Len, Raphaël, Dieter, Edith, and Sander, for the great joy, and the genuine day-to-day support ranged from academic issues to awkward life luggages (like getting a car unstuck in mud). Here, a special thank goes to my special old friend: Xi, for all those laughters, quarrels, hugs, heart-to-heart talks, enlightening moments and so on and on and on we have experienced together since the Master study time. Also, I am grateful to Robin for writing the Dutch summary of this thesis, as well as answering my endless questions about the PhD examination.

Now to my family and friends. Thanks, Mommy and Daddy, for bringing me up in honey pot but also with bees in, for respecting and supporting my every

decision even though some of them are against their initial wishes, for their big and golden hearts. Thanks, auntie two, for always occurring at the right time to share invaluable advice, and for giving me a great cousin. Thanks, Grandma, for her dramatic thoughts that made me lol. Thanks to Minglun, for the ton of fun, support, companions, surprising illuminations in Gent life. Also, I thank other friends in Gent who I have traveled with, been fed by, and played with. And always, I cannot go without thanking my dear old timeless bosom homies, for the belief, care and warmth beyond the distance, and for the mysterious force of lifting the corner of my mouth by just thinking of them.

Gent, May 2021
Junning Deng

Table of Contents

List of Acronyms

CAM	Canonical Adjacency Matrix
DFT	Discrete Fourier Transform
DL	Description Length
ECG	Electrocardiograph
ERGMs	Exponential Random Graph Models
FORSIED	Formalizing Subjective Interestingness In Exploratory Data Mining
FSM	Frequent Subgraph Mining
IC	Information Content
MDL	Minimum Description Length
RQ	Research Question
SAX	Symbolic Aggregation Approximations
SI	Subjective Interestingness

Samenvatting
– Summary in Dutch –

Steeds meer hedendaagse toepassingen vereisen dat gegevens in een rijke vorm worden gepresenteerd. De traditionele platte database die data-objecten onafhankelijk van elkaar behandelt, is immers niet in staat om informatie over relaties tussen objecten op te slaan. Het is dan ook geen verrassing dat patroondelving gebaseerd op dit soort gegevens geen inzichten oplevert in een groot aantal toepassingen, vooral wanneer deze informatie over de interacties tussen objecten nodig hebben. Daartoe moet men de gegevens in een rijkere vorm weergeven. Bijvoorbeeld in toepassingen waarbij de opeenvolging van objecten van belang is, delven we sequentiële gegevens (denk aan tijdreeksen, medische zorgtrajecten, DNA-sequenties, browsegeschiedenis, ...). In toepassingen waarbij interacties tussen koppels van objecten belangrijk zijn, delven we grafen (denk aan sociale netwerken, biologische netwerken, citatienetwerken, epidemiologische netwerken, ...). In toepassingen waarbij relaties in plaats en tijd van belang zijn, delven we ruimtelijk-temporele gegevens (denk aan weerkaarten, trajecten van bewegende objecten, draadloze communicatienetwerken, ...). Hoewel het dus kan lijken alsof dat de manier waarop we patronen delven in deze soorten rijke gegevens erg *toepassingsspecifiek* is—we delven een ad hoc type gegevens, afhankelijk van het soort informatie dat de toepassing vereist—stellen we vast dat patroondelving in wezen *gebruikersspecifiek* is. Patronen zijn er om de gebruiker tot inzichten te leiden die hem/haar de gegevens helpen te begrijpen, of om het resultaat van een opvolgende taak van de gebruiker te verbeteren. Ze dienen dus de uiteindelijke gebruiker. Deze kernfilosofie—*de gebruiker is koning*—zet onderzoekers binnen het gebied van gegevensdelving ertoe meer praktische methoden te ontwikkelen met een reeks gebruikersgerichte vragen in gedachten, zoals *"wat als de ontdekte patronen correcte informatie opleveren die waardevol is voor de gebruiker A, maar niet voor gebruiker B?"*, *"Wat moet ik doen als de patronen niet in een beknopte vorm worden gepresenteerd waarin de gebruiker ze gemakkelijk kan verwerken?"*, *"wat als ze overtollige informatie bevatten, die de gebruiker hindert?"*, of *"wat als ze niet duidelijk zijn voor de gebruiker?"*. Al deze zorgen komen neer op het stellen van één enkele vraag: *"zijn de verkregen patronen werkelijk interessant voor de gebruiker?"*.

Naar onze mening is een eerste stap voor het beantwoorden van deze vraag het meten van hoe *interessant* patronen zijn op een *subjectieve* manier, d.w.z. voor de gebruiker. Meer specifiek mogen gedolven patronen niet worden beoordeeld

op basis van een objectieve norm die beperkt is tot een bepaalde probleemsetting, maar moet dit afhangen van hoeveel ze de gebruiker kunnen tegenspreken of aanvullen gegevens diens voorkennis, en hoeveel moeite het kost om het patroon te verwerken gegeven de complexiteit om deze te beschrijven. Via een methode uitgerust met een dergelijke subjectieve maat voor interessantheid, kunnen verschillende gebruikers (mogelijks) verschillende patronen verkrijgen die precies aansluiten op hun behoeften en dus waardevol zijn, ook al analyseren ze ieder dezelfde dataset.

Het begrijpen van rijke gegevens is een veeleisend probleem, en zoals gezegd is patroondelving inherent gebruikersspecifiek. Deze inzichtelijke observaties maken nieuw onderzoek naar patroondelving mogelijk, maar stellen ook drie belangrijke uitdagingen voor: 1. *Hoe gaan we om met rijke gegevens?* 2. *Hoe kunnen we efficiënt patronen delven die ook rijke structuren of semantiek omvatten?* 3. *Hoe kunnen we de koning—de gebruiker van de ontwikkelde methode—werkelijk tevredenstellen?*

In dit proefschrift gaan we deze uitdagingen aan. We introduceren nieuwe methoden om subjectief interessante inzichten te verkrijgen over twee populaire soorten van rijke gegevens: *tijdreeksen* en *grafen met attributen*. Hiertoe stellen bouwstenen voor om patroondelving uit te voeren (d.w.z. syntaxis van patronen, interessantheidsmaten, en algoritmes), specifiek bedoeld voor rijke gegevens. Hiermee gaan we de bovengenoemde eerste en tweede uitdaging aan. De vooruitgang van ons werk ten opzichte van bestaande methoden is dat we een interessantheidsmaat op een subjectieve manier formaliseren, in plaats van één objectieve maat te introduceren voor alle gebruikers. Hiermee wordt tevens de derde hierboven genoemde uitdaging aangegaan. In wat volgt, vatten we de belangrijkste twee delen van ons werk samen, overeenkomend met de twee onderzochte soorten van rijke gegevens.

Deel 1. Het eerste deel van dit proefschrift introduceert ons werk over patroondelving op basis van tijdreeksgegevens. Hier zijn de specifieke patronen die we delven *motieven*, d.w.z. aangrenzende deelreeksen die vaak terugkeren in de tijdreeks. Motieven verwijzen meestal naar nuttige informatie over seizoensgebonden of tijdelijke associaties tussen gebeurtenissen. In de praktijk is het dan ook erg nuttig om deze motieven te ontdekken.

Het meest onderscheidende kenmerk van dit deel betrekt zich tot de interessantheid van motieven. Bestaande methoden gebruiken allemaal 'objectieve' maten, waarbij ofwel prioriteit wordt gegeven aan de gelijkenis tussen instanties (soms definiëren ze zelfs een motief als het meest gelijkaardige paar van deelreeksen), ofwel aan de steun van een motief (d.w.z. het aantal instanties dat deze bevat). Echter, wij kwantificeren de interessantheid van een motief op een subjectieve manier, waarbij we steunen op informatietheorie om rekening te houden met de eerdere verwachtingen die de gebruiker kan hebben over de tijdreeks. Dit resulteert in een zeer natuurlijke en elegante manier om een compromis te vinden tussen het gelijkaardig zijn van motieven en het aantal instanties dat deze bevatten, en om iteratief nieuwe motieven te delven (door eerder ontdekte motieven te beschouwen

als onderdeel van de eerdere overtuigingen van de gebruiker). Hoewel onderzoek naar de subjectieve interessantheid van patronen de laatste jaren een succesvolle vooruitgang gekend heeft, is de toepassing ervan op tijdreeksen geheel nieuw.

Een tweede onderscheidend kenmerk is het volgende. Bestaande methoden zijn doorgaans afhankelijk van twee essentiële bouwstenen: een maat voor gelijkenis zoals de Euclidische afstand of 'Dynamic Time Warping' (DTW), en een specifieke representatie van tijdreeksen zoals 'Symbolic Aggregate ApproXimation' (SAX), de discrete fouriertransformatie (DFT), of willekeurige projecties. Onze informatietheoretische benadering vereist geen van deze bouwstenen en is dus aantoonbaar minder willekeurig en eleganter dan reeds bestaande methoden.

Deel 2. Het tweede deel van dit proefschrift situeert zich op het gebied van patroondelving in grafen. Meer specifiek introduceren we nieuwe methoden om subjectief interessante lokale en globale deelgraafpatronen te vinden in een graaf met knoopattributen.

Een graaf met knoopattributen is een veelzijdige datastructuur—het kan zowel connectiviteitsrelaties tussen objecten (via *knopen en bogen*) alsook individuele kenmerken van elk object (via *knoopattributen*) weergeven. In de meeste gevallen is de connectiviteitsstructuur van grafen gerelateerd aan de attributen van de knopen. Bijvoorbeeld in een klanten-aankopen-goederen-graaf hangt de waarschijnlijkheid op een link tussen een klant en een product af van een reeks attributen die de klant kenmerken, zoals diens leeftijd, geslacht, salaris, en burgerlijke staat, alsook een reeks attributen van het product, zoals diens prijs, functie, merk, en recensies. Patronen van de vorm 'de deelgroep van objecten met bepaalde eigenschappen X zijn vaak (of zelden) geconnecteerd aan objecten in een andere deelgroep met bepaalde eigenschappen Y', kunnen dus mogelijks bruikbare en veralgemeenbare inzichten in grafen opleveren.

De methoden die wij zullen voorstellen kunnen dergelijke patronen delven. Meer specifiek stellen ze iemand in staat om op een effectieve en begrijpelijke manier grafen te beschrijven, in termen van eenvoudig voor te stellen blokpartities of lokale blokken, met interessante blokdensiteiten. Hierdoor zijn we in staat om verschillende bekende problemen binnen het gebied van graafdelving aan te pakken, waaronder het ontdekken van linkregels, het delven van dichte of schaarse (bipartiete) deelgrafen, en het beschrijven van grafen. Bovendien benaderen we de kwantificering van de interessantheid van de voorgestelde patronen op een subjectieve manier, rekening houdend met de verschillende soorten voorkennis die de gebruiker kan hebben over de graaf, inclusief inzichten verkregen uit eerdere patronen.

Summary

Nowadays, an increasingly large number of applications necessitate presenting data into a richer form. The traditional flat tabular form which treats data objects independently from each other, cannot store the information about relationships between objects. Not surprisingly, basing the pattern mining process on this kind of data fails to gain insights for a wide spectrum of applications, especially those relying on objects-interaction information. Richer data are thus brought into the picture: In applications where sequential relationships matter, we mine sequential data (e.g., any time series, healthcare trajectory, DNA sequence, web surfing history); In applications where pairwise interaction relationships matter, we mine graph structured data (e.g., social networks, biological networks, citation network, virus diffusion networks); In applications where spatial and temporal relationships matter, we mine spatial-temporal data (e.g., weather maps, moving objects trajectory, wireless communication networks).

Though it appears pattern mining in rich data is very *application specific*—we mine an ad-hoc type of rich data depending on what types of information is demanding in an application, we argue that pattern mining is intrinsically *user specific*. Mined patterns are there to provide insights that can either improve the user's understanding about the data or boost his or her performance on a downstream task, and hence they ultimately serve for users. This core philosophy—*user is king*—pushes researchers in data mining community to immerse themselves in developing more practical mining tools, with a series of user-centered questions to address in mind, such as: *What if the discovered patterns provide correct information that is valuable to the user A but not to another user B? What if they are not presented in succinct form for the user to easily assimilate? What if they include redundant information, getting the user bored? What if they are not self-explanatory to the user?* All these concerns boil down to asking a single question—*Are these obtained patterns truly interesting to the user?*

A foremost move towards answering this key question, we believe, is to measure the *interestingness* of patterns in a *subjective* manner—i.e., taking the user into account. More specifically, mined patterns should not be judged based on an objective standard limited to a certain problem setting, but rather, this should depend on how much they can contradict or complement what the user already held (i.e., considering the prior knowledge) and how much effort assimilating them needs (i.e., considering the descriptional complexity of the pattern). With a data mining tool equipped with such subjective interestingness measure, different users can obtain (potentially) different patterns that precisely match their needs and are

thus truly useful, even though they are analyzing the same dataset.

As being said, making sense of richer data types is highly demanding, and pattern mining is inherently user specific. These insightful observations provide a springboard for pattern mining research but also set up three main challenges: 1. *How to handle richer data?* 2. *How to efficiently mine patterns that also carry richer structures or semantics?* 3. *How to really satisfy the king—the user of the data mining tool?*

This thesis is a piece of work dedicated to handling these challenges—we present novel methods to obtain subjectively interesting insights on two popular rich data types: *time series* and *attributed graphs*. In a high-level view, we have proposed building blocks for the pattern mining process (i.e., pattern syntaxes, interestingness measures, mining algorithms) that are dedicated to rich data types—this is to approach the aforementioned first and second challenges. Moreover, the leap of our work with respect to the state-of-the-art is a formalization of a subjective interestingness measure, rather than an objective one for these data types—this addresses the third challenge mentioned above. In what follows, we summarize each of our two main works (corresponding to two rich data types we have investigated).

Part 1. The first part of the thesis presents our pattern mining work on time series data. Here, the specific patterns we consider to mine are *motifs*, i.e., contiguous subsequences that recur in the time series. Motifs usually hint at useful information about seasonal or temporal associations between events, and detecting them are very useful in practice.

To summarize, the most distinctive feature of our work is regarding the interestingness of motifs. Existing methods all use 'objective' measures, either prioritizing the similarity among instances (in some work even defining a motif as the most similar subsequence pair), or prioritizing the support (i.e., the number of instances in a motif). In contrast to this, we quantify the interestingness of a motif in a subjective manner, relying on information theory to take into account prior expectations the user may hold about the time series. This results in a very natural and elegant way of trading of similarity with numerosity of the instances of a motif, and of iteratively mining motifs (by considering previously discovered motifs as part of the prior beliefs of the user). While there is a growing and successful body of work on the subjective interestingness of data mining patterns in recent years, its application to time series is entirely novel.

A second distinctive feature is the following. State-of-the-art methods commonly depend on two essential building blocks: a similarity measure (e.g., Euclidean distance, dynamic time warping) and a special representation for time series (e.g., Symbolic Aggregate approXimation (SAX), Discrete Fourier Transform (DFT), Random projections). Our information-theoretic approach does not require either of these building blocks, making it arguably less arbitrary and more elegant than pre-existing methods.

Part 2. The second part of the thesis is situated in the field of graph pattern mining. More specifically, we present novel methods for finding subjectively interesting local and global subgraph patterns in a vertex-attributed graph.

A vertex-attributed graph is a versatile data structure—it can represent both connectivity relationships between objects (in terms of *vertices and edges*) and individual characteristics of each object (in terms of *vertex attributes*). More often than not, the connectivity structure of graphs is related to the attributes of the vertices. In a customers-purchase-goods network for instance, the probability of a link characterising a purchase relationship from a customer to an item depends on a range of attributes, such as the age, gender, salary, marital status on the customer's side, and the price, function, brand, reviews on the item's side. Thus, patterns of the form 'the subgroup of objects with certain properties X are often (or rarely) connected with objects in another subgroup defined by properties Y' can present potentially actionable and generalisable insights into the graph.

Our proposed methods can mine such patterns. More specifically, they allow one to effectively summarize graphs in an intelligible manner, in terms of an easy-to-describe block partitioning of the graph with interesting block densities, or in terms of easy-to-describe local blocks in the graph with interesting densities—tackling several well-known graph mining tasks simultaneously including: link rule discovery, dense/sparse (bipartite) subgraph mining and graph summarization. Moreover, we approach the quantification of the interestingness of proposed patterns in a subjective manner, with respect to several flexible types of prior knowledge the user may have about the graph, including insights obtained from previous patterns.

1 Introduction

1.1 Context

1.1.1 Why data mining?

In 1989, Gregory Piatetsky-Shapiro coined the term KDD (Knowledge Discovery in Databases) [1], and organized the first workshop named also KDD (which then grew into the annual ACM SIGKDD Conference in 1995, the most influential forum in the field of data mining). At that time, the only people who had enormous data sets and the motivation to make sense of them were members of the research community.

Around the early 1990s, the term *data mining* appeared in the database community to represent the application of specific algorithms for extracting useful knowledge from data. The distinction between KDD and data mining is that the former refers to the overall process of identifying knowledge from data while the latter refers to a particular sub-process or a step within this process [2].

In no time, data mining came into prominence, not only in research communities, but also in industries. This was due to a megatrend commencing at that time—data started to overwhelm the world. This trend was not accidental. On the one hand, the extraordinary development of hardware technologies led to the huge efficient data stores on hard disks, making the processing of immense volumes of data possible. On the other hand, the explosive growth of Internet digitalized almost every aspect of our lives, making a data source to easily reach anyone who cared to tap in. As a result, tremendous amounts of data are accumulated daily:

health records, patient monitoring, customer transactions, web-visiting logs, call data records, stock trading records, economic data, social network, search queries, weather data, geo-spatial data, and so forth—Indeed, we live in a world of data.

Clearly, *knowledge* is the end product of a data-driven discovery that data analysts (who mine the data or apply the mining tool) want to obtain to leverage their objectives. The explosively growing, gigantic body of data, nevertheless, has far exceeded the human ability to comprehend manually. Tools that can *automatically* unearth valuable knowledge from the big data are thus badly needed. Luckily, data mining is such a tool, and is a powerful one.

1.1.2 Pattern mining

Often, data analysts clearly know what patterns of information from the data are digestible and valuable to their end goals. Healthcare workers want to know how certain variables are associated with the onset of diabetes. Geneticists want to identify types of sequence segments upstream and downstream the gene region that signal the gene expression. Retailers want to know purchase habits of customers to provide them with more valuable and personalised services. Hence, data analysts with well-defined goals often expect the data mining tool to output information that is in certain patterns dedicated to end-goals rather than any knowledge, and this brings *pattern mining* into picture.

Pattern mining is a pattern-based approach to data mining concerned with the acquisition of patterns from data. According to Cambridge Dictionary, a *pattern* is expressed as a particular way in which something usually happens or is done. This agrees with our common perception of pattern—the one being based on (high) occurrence frequency of object. Hence, the corresponding *frequent pattern mining* [3], which aims to discover patterns with frequency of occurrence no less than a user-specified threshold, has been the most well known and widely studied type of pattern mining in recent decades. Here, a pattern can be *a frequent itemset* [4] when given a tabular dataset (e.g., sets of frequent-buying-together items in transaction data), or *a frequent sequence* [5] when given a sequential dataset (e.g., motifs representing normal heartbeats in time series of electrocardiodiagram), or *a frequent substructure* such as subgraphs [6], subtrees [7] or sublattices if the given dataset is a graph (e.g., graphlets related to specific biological functions in biomolecule interaction network).

As various kinds of data, user requests and applications burst in recent years, pattern is now a broader term that can be defined to represent any structure or a characteristic form of information of particular interest. For example, in contrast with frequent patterns which can also be mapped into *association rules* [8], a user may want to identify *exceptional patterns* (i.e., patterns that occur rarely but signal an important anomaly) [9, 10] or *negative patterns* (i.e., patterns that reveal a neg-

ative correlation between objects) [11–13]. Different patterns to be mined spark the development of different measures to quantify their interestingness along with different methodologies to search them. A general road map on pattern mining research classified along three dimensions (i.e., the kinds of patterns mined, interestingness measures, and mining methodologies) is given in Chapter 2. Now we take a look at two observations in today's pattern mining research that motivate the subject of this thesis.

1.1.3 The first motivation: mining richer data

When most people hear the word "data", a tabular form springs to mind: a set of objects arranged in horizontal rows and vertical columns such that each row has the same set of column headers. One example can be an epidemiological dataset comprised of infected patients (corresponding to rows). For each patient, information about a bunch of attributes such as the gender, the age, the blood type, the infection case and the health condition is stored (corresponding to columns).

Nevertheless, some objects often associate with each other. Data in this traditional tabular form which treats objects independently from each other is an inherently lossy representation. Not surprisingly, basing our mining process on this kind of data can fail for a broad spectrum of applications.

Therefore, an increasing amount of work has been invested in mining patterns on richer data—one taking a more complex form that enables to store various kinds of dependency information correlated with temporal, spatial, sequential, and social relationships such as sequential data, graphs, spatial-temporal data, multimedia data, and so on. Among these richer data, sequential data and graphs are two of the most popular types. We motivate mining patterns on them in the following.

Sequential data. In many domains, the sequential ordering of events or objects plays an important role. For example, to predict a customer's next purchase, it is often relevant to consider a sequence of past purchases of this customer; For many things being manufactured, a series of mechanical or manual operations must be performed in certain order. Even, it is the order of nucleotides that defines DNA—the code of life. To leverage sequential information of objects, *sequential* data is thus proposed.

Two categories of sequential data are commonly used in pattern mining: *time series* and *symbolic sequence* [14]. A *time series* is a sequence of numerical values in time order. Such data may provide useful information about trending, seasonal, irregular or temporal associations between objects or events, and is thus very pervasive, originating from sources as diverse as wearable devices, medical equipment, sensors in industrial plants, and our mother nature. The other category, *symbolic sequence*, is an ordered sequence of nominal data, recorded with or

without a concrete notion of time. Common examples include web surfing logs, customer purchase sequences, word sequences in texts and so on.

Graph data. Another kind of dataset that does not fit in the flat tabular setting is one involving pairwise interactions or relationships between defined objects. Rather, such dataset can be readily structured as *graphs* where objects are represented by *vertices* and relationships between them are represented by edges.

As a mounting body of applications rely on taking advantage of or making sense of relationships between objects, graphs are pervasively used. Examples include social networks, Semantic Webs, citation networks, protein interaction networks, traffic flows, among others. Thus, a *first reason* to mine patterns in graphs is their ubiquity.

A *second reason* is the fact that graph is a general model. Trees, lattices, sequences and items are all degenerated graphs. Moreover, graphs can be directed or undirected, attributed or unattributed (on edges or vertices), weighted or unweighted, static or dynamic, homogeneous or heterogeneous (on edges or vertices). Such flexibility makes graphs able to model not only pairwise relationships but also other types of information.

1.1.4 The second motivation: user is king—subjective interestingness

Once having data in hand which is in appropriate form, we then design the pattern syntax such that it expresses the form of information that the user is looking for. Now what we are confronted to mine is not the vast set of data objects, but rather the vast set of patterns. With no doubt, not all of the patterns are interesting, and only those most interesting ones which typically account for a small fraction should be presented to the user. Then naturally, a new question is raised for pattern mining—*What makes a pattern interesting?*

The vast majority of pattern mining research, especially those in early times, tacitly quantify the interestingness in an *objective* way—i.e., they design the interestingness measure with merely data or patterns to be mined in mind. Not surprisingly, following this objective notion of interestingness has led to a large number of diverse interestingness measures, each for a particular criteria of a specific task. Even only regarding the task of frequent pattern mining, examples of interestingness measures are numerous, including frequency, support, area, lift, growth rate and so on. Nevertheless, what this raft number of differently proposed measures for a same specific pattern mining task signifies is the wide void between available interestingness measures and practical needs. Because of this void, researchers usually quickly identified the old measures' limitations and proposed new ones which outperform or complement the old, again and again.

The source of this void, we believe, is the objective notion of interestingness. Patterns ultimately serve users, but users differ from each other. An objective standard may be able to represent a common sense or several similar users' beliefs, but cannot represent various users. That means, a pattern may be deemed interesting by a user A but not by another user B, because this pattern is previously known or can be easily implied by user B's prior knowledge. To discover truly interesting patterns, data mining tools therefore need to take users into account, such that the discovered patterns should be surprising or unexpected to the user (as interests are always caught by surprise), i.e., representing knowledge which contradicts or complements the user's prior knowledge to a sufficient extent.

To do that, an obvious method is to explicitly make the data mining tool an interactive one such that it enables the user to control several relevant factors as constraints of the results or send feedbacks about preliminary patterns as navigations. Nevertheless, this is often a slow process of trial and error, and more undesirably, this method requires a body of users' efforts. One method which can circumvent these issues is to make the interestingness measure a subjective one. Ideally, such a measure should work as a function, with inputs as a pattern together with the user's prior knowledge, outputting a score accordingly. Here, the user's prior knowledge should be expressed in a form which requires users' efforts as minimal as possible for saving his or her efforts, because as always—*user is king*.

1.2 Contributions

Richer data types are powerful. Interestingness is inherently more subjective than objective. These observations bring about actionable insights, but also new challenges—*how to handle richer data, how to efficiently mine patterns that also carry richer structures or semantics, and how to really satisfy the king—the user of the data mining tool*. Driven by these, the research reported in this thesis joins the line of investigating personalized pattern mining methods on richer data. This thesis presents our contributions for that purpose—i.e., novel methods for obtaining subjectively interesting insights on time series (first contribution) and graphs (second contribution):

First contribution. Our first contribution is dedicated to mining one common type of sequential data: time series. Numerical time series data is pervasive, originating from sources as diverse as wearable devices, medical equipment, to sensors in industrial plants. In many cases, time series contain interesting information in terms of subsequences that recur in approximate form, so-called *motifs* (see an example of a motif in a household electric usage time series [15] in Fig 1.1). Major open challenges in this area include how one can formalize the interestingness of such motifs, and how the most interesting ones can be found.

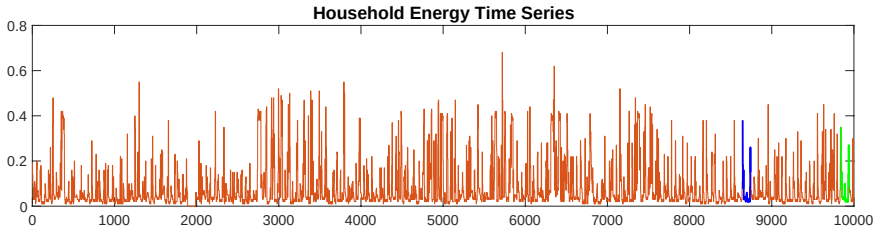


Figure 1.1: A motif of length 100 is discovered at two different locations (blue and green) in a household electric usage time series.

We introduce a novel approach that tackles these issues. We formalize a notion of such subsequence patterns in an intuitive manner, and present an information-theoretic approach for quantifying their interestingness with respect to any prior expectations a user may have about the time series. The resulting interestingness measure is thus a *subjective* measure, enabling a user to find motifs that are truly interesting *to them*. Although finding the best motif appears computationally intractable, we develop relaxations and a branch-and-bound approach implemented in a constraint programming solver. As shown in experiments on synthetic data and two real-world data sets, this enables us to mine interesting patterns in small or mid-sized time series. This contribution is mainly based on the manuscript that appeared as:

- Junning Deng, Jefrey Lijffijt, Bo Kang and Tijl De Bie. SIMIT: Subjectively Interesting Motifs in Time Series. *Entropy*, 21(6), 2019.

Early results were published in a workshop paper:

- Junning Deng, Jefrey Lijffijt, Bo Kang and Tijl De Bie. Subjectively Interesting Motifs in Time Series. *In 3rd International Workshop on Advanced Analytics and Learning on Temporal Data, held with ECML/PKDD*, 2018.

Second contribution. We also contribute to pattern mining on graphs, more precisely, graphs with attributes on vertices. The connectivity structure of graphs is typically related to the attributes of the vertices. In social networks for example, the probability of a friendship between any pair of people depends on a range of attributes, such as their age, residence location, workplace, and hobbies. The high-level structure of a graph can thus possibly be described well by means of patterns of the form 'the subgroup of all individuals with certain properties X are often (or rarely) friends with individuals in another subgroup defined by properties Y', ideally relative to their expected connectivity. Such rules present potentially actionable and generalizable insight into the graph.

Prior work has already considered the search for dense subgraphs ('communities') with homogeneous attributes. The first contribution in this paper is to generalize this type of pattern to densities between a *pair of subgroups* (e.g., a pattern that describes the friendship between a particular subgroup of female and a subgroup of male individuals in a social network), as well as between *all pairs from a set of subgroups that partition the vertices* (e.g., a *global* pattern that describes the friendship between any pair of subgroups selected from a set of subgroups that form a partition of the individuals in a social network). Second, we develop a novel information-theoretic approach for quantifying the subjective interestingness of such patterns, by contrasting them with prior information a user may have about the graph's connectivity. We demonstrate empirically that in the special case of dense subgraphs, this approach yields results that are superior to the state-of-the-art. Finally, we propose algorithms for efficiently finding interesting patterns of these different types. This contribution has been published as:

- Junning Deng, Bo Kang, Jefrey Lijffijt, and Tijl De Bie. Mining Explainable Local and Global Subgraph Patterns with Surprising Densities. *Data Mining and Knowledge Discovery*, 2020.

A subset of results appeared earlier in a conference paper:

- Junning Deng, Bo Kang, Jefrey Lijffijt, and Tijl De Bie. Explainable subgraphs with surprising densities : a subgroup discovery approach. *In Proceedings of SIAM International Conference on Data Mining (SDM)*, 2020.

Outline. The thesis is outlined as follows:

- **Pattern mining basics [Chapter 2]**. This chapter introduces the reader to the necessary background on pattern mining. More specifically, we start by providing an overview of the pattern mining process that is composed of three essential building blocks—pattern syntaxes, interestingness measure, and mining algorithms. We then detail each building block by introducing its role and relevant state-of-the-art along some key aspects.
- **Subjectively interesting motifs in time series [Chapter 3]**. This chapter presents our first main contribution: mining subjective interesting motifs in time series.
- **Explainable local and global subgraph patterns with surprising densities [Chapter 4]**. In this chapter, we presents our second main contribution: mining explainable local and global subgraph patterns with surprising densities.
- **Conclusions [Chapter 5]**. This chapter concludes our work and discusses the future directions from a general view.

1.3 Publications

Publications in international journals

- Junning Deng, Bo Kang, Jefrey Lijffijt, and Tijl De Bie. Mining Explainable Local and Global Subgraph Patterns with Surprising Densities. *Data Mining and Knowledge Discovery*, 2020.
- Junning Deng, Jefrey Lijffijt, Bo Kang and Tijl De Bie. SIMIT: Subjectively Interesting Motifs in Time Series. *Entropy*, 21(6), 2019.

Publications in archived proceedings

- Junning Deng, Bo Kang, Jefrey Lijffijt, and Tijl De Bie. Explainable subgraphs with surprising densities : a subgroup discovery approach. *In Proceedings of SIAM International Conference on Data Mining (SDM)*, 2020.

Publications in non-archived proceedings

- Junning Deng, Bo Kang, Jefrey Lijffijt, and Tijl De Bie. Explainable subgraphs with surprising densities : a subgroup discovery approach. *In 15th International Workshop on Mining and Learning with Graphs, held with SIGKDD*, 2019.
- Junning Deng, Jefrey Lijffijt, Bo Kang and Tijl De Bie. Subjectively Interesting Motifs in Time Series. *In 3rd International Workshop on Advanced Analytics and Learning on Temporal Data, held with ECML/PKDD*, 2018.

References

[1] G. Piatetsky-Shapiro. *Knowledge discovery in real databases: A report on the IJCAI-89 workshop*. AI magazine, 11(4):68–68, 1990.

[2] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. *Knowledge Discovery and Data Mining: Towards a Unifying Framework*. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pages 82–88, 1996.

[3] J. Han, H. Cheng, D. Xin, and X. Yan. *Frequent pattern mining: Current status and future directions*. Data Mining and Knowledge Discovery, 15(1):55–86, 2007.

[4] R. Agrawal, T. Imieliński, and A. Swami. *Mining association rules between sets of items in large databases*. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207–216, 1993.

[5] R. Agrawal and R. Srikant. *Mining sequential patterns*. In Proceedings of the 11th International Conference on Data Engineering, pages 3–14, 1995.

[6] D. J. Cook and L. B. Holder. *Substructure Discovery Using Minimum Description Length and Background Knowledge*. Journal of Artificial Intelligence Research, 1(1):231–255, 1994.

[7] M. J. Zaki. *Efficiently mining frequent trees in a forest: Algorithms and applications*. IEEE Transactions on Knowledge and Data Engineering, 17(8):1021–1035, 2005.

[8] R. Agrawal, R. Srikant, et al. *Fast algorithms for mining association rules*. In Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, volume 1215, pages 487–499, 1994.

[9] W. Duivesteijn, A. Feelders, and A. Knobbe. *Different slopes for different folks: mining for exceptional regression models with cook's distance*. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, pages 868–876, 2012.

[10] W. Duivesteijn, A. J. Feelders, and A. Knobbe. *Exceptional model mining*. Data Mining and Knowledge Discovery, 30(1):47–98, 2016.

[11] Z. Zheng, Y. Zhao, Z. Zuo, and L. Cao. *Negative-GSP: An efficient method for mining negative sequential patterns*. In Conferences in Research and Practice in Information Technology Series, 2009.

[12] X. Dong, Z. Zheng, L. Cao, Y. Zhao, C. Zhang, J. Li, W. Wei, and Y. Ou. *e-NSP: efficient negative sequential pattern mining based on identified positive patterns without database rescanning*. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pages 825–830, 2011.

[13] S.-C. Hsueh, M.-Y. Lin, and C.-L. Chen. *Mining negative sequential patterns for e-commerce recommendations*. In 2008 IEEE Asia-Pacific Services Computing Conference, pages 1213–1218, 2008.

[14] J. Han, M. Kamber, and J. Pei. *13-data mining trends and research frontiers*. Data Mining (Third Edition), ed Boston: Morgan Kaufmann, pages 585–631, 2012.

[15] J. Gopinadhan. *House Hold Energy Data - Time Series*, 2019. Accessed: January 25, 2021. Available from: https://www.kaggle.com/jaganadhg/house-hold-energy-data.

2 Pattern Mining Basics

2.1 Overview

Pattern syntaxes, interestingness measures, and mining algorithms are the three building blocks for a pattern mining task.

The *pattern syntax* z is the abstract form of patterns that the user wishes to find, defined as a Boolean function $z : \mathcal{L} \rightarrow \{\text{true}, \text{false}\}$ where $\mathcal{L}$ is the overall pattern space and $z(\pi) \triangleq$ true if and only if the pattern $\pi \in \mathcal{L}$ follows the syntax. For a given dataset $\mathcal{D}$, we denote the pattern space subject to it (i.e., the set of all patterns that are facts for $\mathcal{D}$) by $\mathcal{L}_{\mathcal{D}}$. Given a task of mining patterns with a syntax z from a dataset $\mathcal{D}$, we can determine the *pattern language* $\mathcal{L}_{(\mathcal{D},z)}$, the domain of patterns to be enumerated, i.e., $\mathcal{L}_{(\mathcal{D},z)} = \left\{\pi \in \mathcal{L}_{\mathcal{D}} : z(\pi) \triangleq \text{true}\right\}$.

Needless to say, some patterns are more interesting than others and should thus be prioritized as output. This prioritization is often navigated by the so-called *interestingness measure* such that most interesting patterns should be those having largest values of this measure. More formally, the interestingness measure is defined as a function $m_{\mathcal{D}} : \mathcal{L}_{(\mathcal{D},z)} \rightarrow \mathbb{R}_{\geq 0}$, that assigns a nonnegative numeric value to a pattern $\pi \in \mathcal{L}_{(\mathcal{D},z)}$, given a dataset $\mathcal{D}$.

Once these two building blocks are properly determined, the next question is: *How to mine the most interesting patterns efficiently?* The efficiency is explicitly required here because more often than not, the pattern language exhibits large search space due to the sheer size of the confronted data.

Example 1. *Now let us illustrate these concepts by perhaps the most well-known*

example in pattern mining: frequent itemset mining problem in market basket analysis [1]. In this problem, a dataset $\mathcal{D} = (I, O)$ is given where I is a set of items and O is a set of transactions such that each transaction $o \in O$ is a subset of items from I , i.e., $o \subseteq I$. Clearly, the pattern syntax z here is a form of itemset (with the corresponding pattern language $\mathcal{L}_{((I,O),z)} = 2^I$). The interestingness of a pattern $\pi \in 2^I$ can be straightforwardly measured by, for instance, the number of transactions that contain all the items in π i.e., $m_{\mathcal{D}}(\pi) = \text{freq}(\pi)$ where $\text{freq}(\pi) = |\{o \in O : \pi \subseteq o\}|$.

In a nutshell, pattern mining process is all about: to design the right pattern syntax z and the right interesting measure $m_{\mathcal{D}}$ according to the given dataset $\mathcal{D}$ and the user's need, and then use a mining algorithm to find interesting patterns w.r.t. $m_{\mathcal{D}}$ from the pattern language $\mathcal{L}_{(\mathcal{D},z)}$.

Two categories. Depending on how an interesting pattern is defined, pattern mining methods can be divided into two categories [2]: one is formulated as a *satisfaction* problem (termed *constraint-based* pattern mining)—where a pattern π is deemed interesting if $m_{\mathcal{D}}(\pi)$ (i.e., the interestingness measure of this pattern π) is larger than a user-specified threshold q, the other is formulated as an optimisation problem (termed *preference-based* pattern mining)—where a pattern π is interesting when no other pattern (or only k patterns) has a larger value w.r.t. $m_{\mathcal{D}}$.

More formal definitions for constraint-based pattern mining and preference-based pattern mining are provided in the following:

Problem 1. (Constraint-based pattern mining). Given a dataset $\mathcal{D}$, a pattern syntax z, an interestingness measure $m_{\mathcal{D}}$ and a threshold q, constraint-based pattern mining aims to find all patterns from the pattern language $\mathcal{L}_{(\mathcal{D},z)}$ such that their values w.r.t. $m_{\mathcal{D}}$ are no less than q:

$$\mathcal{T}h_q(\mathcal{D}, z, m_{\mathcal{D}}) = \{\pi \in \mathcal{L}_{(\mathcal{D},z)} : m_{\mathcal{D}}(\pi) \geq q\}$$

Problem 2. (Preference-based pattern mining). Given a dataset $\mathcal{D}$, a pattern syntax z, an interestingness measure m and a threshold k, preference-based pattern mining aims to find all patterns from the pattern language $\mathcal{L}_{(\mathcal{D},z)}$ which are not dominated by at least k patterns:

$$\begin{aligned}\mathcal{B}est_k(\mathcal{D}, z, m_{\mathcal{D}}) = \big\{&\pi \in \mathcal{L}_{(\mathcal{D},z)} : \nexists \Phi \text{ s.t., } |\Phi| \geq k, \\ &\forall \phi \in \Phi, \phi \in \mathcal{L}_{(\mathcal{D},z)} \text{ and } m_{\mathcal{D}}(\phi) \geq m_{\mathcal{D}}(\pi)\big\}\end{aligned}$$

Example 2. *Back to our example of frequent itemset mining in market basket analysis. The constraint-based pattern mining of this problem aims to find a itemset with occurrence frequency no less than a user-defined threshold (denoted as* minfreq*):*

$$\mathcal{T}h_{\text{minfreq}}((I, O), z, \text{freq}) = \{\pi \in 2^I : \text{freq}(\pi) \geq \text{minfreq}\}.$$

The preference-based version is to find top-k frequent itemsets:

$$\mathcal{B}est_k\left((I,O),z,\text{freq}\right) = \big\{\pi \in 2^I : \nexists \Phi \text{ s.t., } |\Phi| \geq k, \\ \forall \phi \in \Phi, \phi \in 2^I \text{ and freq}(\phi) \geq \text{freq}(\pi)\big\}.$$

In recent decades, these two categories of pattern mining problems have been instantiated in a vast number of data mining tasks—by specifying a pattern syntax, an interestingness measure and a mining algorithm that match the dataset to be mined, the user's intention, or the application. Now let us take a look at some main aspects of each of these three building blocks.

2.2 Building blocks for pattern mining

2.2.1 Pattern syntaxes

A myriad kinds of pattern syntaxes have been proposed, all for a specific problem setting faced or imagined. Common examples include:

- emerging patterns [3, 4], association rules [1, 5, 6], subgroups [7–9] in *flat tabular data*;
- motifs (or frequent subsequences) [10–13], time series shapelets [14–16], outliers [17–21], episodes [22–25] in *sequential data*;
- frequent subgraphs [26–29], dense subgraphs [30–33], trees [34–36], cycles [37, 38] in *graph data*;
- moving together patterns [39–41], sequential trajectories [42–44], outlier trajectories [45, 46] in *spatial-temporal data*;

2.2.2 Interestingness measures

Interestingness measures are intended to quantify how good a discovered pattern is deemed to be. Though contemporary research on formalizing interestingness measures has been hugely active and well-established, so far there is no consensus of how the interestingness should be precisely defined. Among this large number of diverse definitions and approaches to interestingness proposed in the literature, two major categories are exhibited [47]: *objective* and *subjective* interestingness measures—according to whether the user's prior knowledge is considered.

Objective interestingness measures only depend on the data and patterns. Most of them are formalized based on theories in probability, statistics, or information theory, and are intended to prioritize or consider factors such as *conciseness* (e.g., cardinality of a pattern set), *coverage* (e.g., *support* of an itemset [48]), *reliability* (e.g., *accuracy* of a classification rule [49]), *peculiarity* (e.g., *peculiarity* of a data

object [50]) or *diversity* (e.g., *cover redundancy* of a subgroup set [9]). Though straightforward forms of objective measures often lend themselves to efficient mining algorithms, the discovered patterns w.r.t. them are often either obvious or already known by the user (and are thus not truly interesting). This is due to the ignorance of variations among users—what is interesting to one may be nothing but useless information to another.

Subjective measures take into account the user, in addition to the data and patterns. To do that, the user's prior knowledge or expectations about the data are modelled by the commonly called *background knowledge*, and then the interestingness of a discovered pattern is measured by how much this pattern deviated from the background knowledge. According to the way the background knowledge is encoded and the deviation is defined, two major classes of methods can be distinguished: the *syntactical* and the *probabilistic* methods. The former encodes the background knowledge by a collection of independent patterns with the same syntax as patterns to be mined (thus the user has to hold some explicit knowledge of the required form), and then employs a distance measure to evaluate their similarity or difference—the more distant, the more interesting (e.g., using fuzzy matching [51], logical contradiction [52]). The latter utilizes a probability distribution of the data (expressed explicitly or implicitly), called *background model*, to encode the background knowledge. Then the interestingness can be either measured by the deviation between a statistic of the pattern calculated on the empirical data and that on the background model (e.g., using Bayesian networks [53], swap randomization [54, 55]), or directly by the probability of this pattern under the background model (e.g., using maximal entropy models [56, 57]).

For excellent surveys of different interestingness measures, we refer interested readers to [47, 58, 59].

2.2.3 Mining algorithms

An ideal mining algorithm is an effective and efficient one—i.e., it can identify interesting patterns within a short response time. Existing algorithms can be categorized into three groups:

1. Candidate generation-and-test algorithms. The first line of research adopts a *candidate generation-and-test* approach, where a sufficiently large enumeration space (hopefully) guaranteed to contain all interesting patterns is generated, and then a portion of top-scoring patterns are selected out by testing the quality of each candidate. Existing algorithms of this paradigm can be further divided into the following three types:

- *Exhaustive search plus pruning strategies.* Exhaustive search is often used in tandem with some pruning strategies, in which the former serves for the

effectiveness, and the latter serves for the efficiency. A bellwether algorithm of this type is *Apriori* [48], also known as the *level-wise algorithm*. Though originally designed for association rule mining, it has now been substantially studied and extended to mine many other patterns. Basically, Apriori adopts a breadth-first manner—i.e., pattern candidates of size k are generated using size-$(k-1)$ candidates, along with a pruning strategy based on an *anti-monotone* property—i.e., if a size-$(k-1)$ itemset is not frequent, none of its size-k super itemset can be frequent. Other algorithms of this type mostly borrow the spirit of Apriori, but differ in the properties exploited to prune the potentially complete search space (e.g., monotonicity [60], convertible constraints [12], succinctness [60, 61], condensed representation [62]). Though equipped with pruning strategies, exhaustive enumeration may still be infeasible, especially when confronting gigantic data.

- *Heuristic search.* The second type explores the enumeration space heuristically—i.e., picks most promising branches according to a certain rule of thumb, termed *heuristic*. Typical examples include *hill climbing* [63, 64], *beam search* [7, 9, 65], *evolutionary algorithms* [66, 67], among others. These algorithms scale better than the exhaustive search with pruning strategies, but they cannot guarantee the optimality. For many of them, even an error bound cannot be given as well.

- *Anytime algorithms.* Anytime pattern mining algorithms are also enumerative methods, but exhibit the so-called *anytime* feature [68]: they can be interrupted at any point of time to supply patterns whose quality gradually improves over time, and hence the whole process would converge to an exhaustive search if sufficient time is given, guaranteeing to return the exact result. Recently, this type of algorithms have been employed to mine frequent itemset [69], interval patterns [70], outliers [71] and so on. Particularly in Belfodil et al. [70], the proposed algorithm can always provide a guaranteed bounding of the quality difference between the top found pattern and the top possible pattern.

  It is worth mentioning that *Branch-and-bound (BnB)* [72], a principal algorithmic methodology which is usually used to find exact solutions to combinatorial optimization problems, can be naturally adapted as an anytime one: BnB algorithms evaluate the search space in a gradual way such that a given problem is decomposed into smaller subproblems (according to a certain *branching* rule) and each of those subproblems may be further decomposed or pruned (according to a certain *bounding* rule), and clearly they find better solutions as less unexplored subproblems remain. BnB algorithms have also been applied to mine patterns such as boxes [73], maximal cliques [74, 75], discriminative patterns [76], and so forth.

2. Pattern-growth algorithms. More often than not, the candidate set generation is costly. To circumvent this, the second main line of research follows a *pattern-growth* paradigm [77–80] led by Han et al. [77] in handling frequent pattern mining. The core of this paradigm is the construction of a highly compact data structure, e.g., *frequent pattern tree (FP-tree)* [77], which stores compressed, crucial information about frequent patterns. Then a pattern-growth method is performed in a divide-and-conquer manner: the database is partitioned and projected based on the currently discovered frequent patterns, and new longer patterns are directly attained by growing discovered ones—through a traversal on that compact data structure. Hence, unlike candidate generation-and-test approaches that require many scans of the entire database (the k-th scan checks the frequency of each size-k candidate), pattern-growth approaches only need two scans (for constructing the compact data structure), then the rest steps are mining rather the compact structure than the (usually substantially larger) original data, which saves a huge cost.

3. Algorithms applying the sampling strategy. Recent pattern mining approaches set up the third and elegant paradigm—i.e., relying on a (controlled) *sampling* strategy [81–84]. More specifically, algorithms falling under this category design an efficient sampling procedure to access the pattern language $\mathcal{L}$, i.e., simulating a distribution $s : \mathcal{L} \to [0, 1]$ that considers the corresponding interestingness measure m, e.g., $s(\cdot) = \frac{m(\cdot)}{Z}$ where Z is a normalizing constant. This enables us to obtain a pattern collection that is of size under control and is representative for the distribution s and hence for the underlying interestingness m, without expensive candidate set generation. However, it is still very probable to draw an uninteresting pattern, because the distribution s is *long-tailed*—there are much more uninteresting patterns than interesting ones.

2.3 Positioning of our methods

Before we embark on the detailed reporting of our research work, let us point out where the methods proposed in this work are situated with respect to the aforementioned aspects of each building block.

Let us first look at the interestingness measures, as making them subjective is what we value most. To this aim, we built upon De Bie's FORSIED (Formalizing Subjective Interestingness in Exploratory Data Mining) framework [56, 57] by instantiating it towards the specific pattern mining problems of our interest. The key idea of FORSIED is to model the user's prior belief state of the data by a probabilistic distribution (called *background distribution*), and this leads to an explicit probabilistic method of formalizing subjective interestingness. The advantages of FORSIED are manifold. We highlight two of them in the following. First, FOR-

SIED is able to incorporate broad classes of user's prior knowledge, whereas what other methods can take into account are more limited, impractical and often dedicated to some specific problem settings. For example, syntactical methods can only account for the user's knowledge about the patterns he or she thinks exist in the data; An implicit probabilistic method for finding frequent itemsets which uses a Bayesian network model requires the user to encode his or her prior information in a Bayesian network [53]. Another implicit probabilistic method which relies on swap randomization can only account for the row and column sums of rectangular databases [54]. Second, in FORSIED, the background distribution can be updated efficiently to incorporate the user's newly acquired patterns, and thus allows for an iterative and interactive pattern mining process. Moreover, we want to stress that, because of these advantages, FORSIED is not just a method for subjective interestingness, but rather a data mining framework.

Our pattern syntaxes not only represent the formats of patterns we want to mine (i.e., motifs in time series, as well as explainable local and global subgraph patterns in attributed graphs), but also, more importantly, they are designed to be in harmony with our formalization of the subjective interestingness (such that the found patterns of these syntaxes can be contrasted with our model of the user's belief state about the data to quantify their interestingness to the user). Clearly, the traditional syntaxes of motifs and subgraph patterns are not applicable here.

As for the mining algorithms, we went for the heuristic ones. This is to counter the extra computational burdens injected by the adoption of subjective interestingness measures.

References

[1] R. Agrawal, T. Imieliński, and A. Swami. *Mining Association Rules between Sets of Items in Large Databases*. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207–216, 1993.

[2] A. Soulet. *Two Decades of Pattern Mining: Principles and Methods*. In P. Marcel and E. Zimányi, editors, Business Intelligence, pages 59–78, 2017.

[3] G. Dong and J. Li. *Efficient mining of emerging patterns: Discovering trends and differences*. In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, pages 43–52, 1999.

[4] J. Bailey, T. Manoukian, and K. Ramamohanarao. *Fast algorithms for mining emerging patterns*. In European Conference on Principles of Data Mining and Knowledge Discovery, pages 39–50, 2002.

[5] B. Liu, W. Hsu, Y. Ma, et al. *Integrating classification and association rule mining.* In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, volume 98, pages 80–86, 1998.

[6] C. Hidber. *Online association rule mining*. ACM Sigmod Record, 28(2):145–156, 1999.

[7] N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski. *Subgroup discovery with CN2-SD*. Journal of Machine Learning Research, 5:153–188, 2004.

[8] M. Atzmueller and F. Puppe. *SD-Map–A fast algorithm for exhaustive subgroup discovery*. In European Conference on Principles of Data Mining and Knowledge Discovery, pages 6–17, 2006.

[9] M. Van Leeuwen and A. Knobbe. *Diverse subgroup set discovery*. Data Mining and Knowledge Discovery, 25(2):208–242, 2012.

[10] R. Agrawal and R. Srikant. *Mining sequential patterns*. In Proceedings of the 11th International Conference on Data Engineering, pages 3–14, 1995.

[11] J. Lin. *Finding Motifs in Time Series*. In Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining, 2002.

[12] Jian Pei, Jiawei Han, B. Mortazavi-Asl, Jianyong Wang, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu. *Mining sequential patterns by pattern-growth: the PrefixSpan approach*. IEEE Transactions on Knowledge and Data Engineering, 16(11):1424–1440, 2004.

[13] K. L. Jensen, M. P. Styczynski, I. Rigoutsos, and G. N. Stephanopoulos. *A generic motif discovery algorithm for sequential data*. Bioinformatics, 22(1):21–28, 2006.

[14] L. Ye and E. Keogh. *Time series shapelets: a new primitive for data mining*. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, pages 947–956, 2009.

[15] T. Rakthanmanon and E. Keogh. *Fast shapelets: A scalable algorithm for discovering time series shapelets*. In Proceedings of the 2013 SIAM International Conference on Data Mining, pages 668–676. SIAM, 2013.

[16] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme. *Learning time-series shapelets*. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 392–401, 2014.

[17] J.-I. Takeuchi and K. Yamanishi. *A unifying framework for detecting outliers and change points from time series*. IEEE Transactions on Knowledge and Data Engineering, 18(4):482–492, 2006.

[18] S. Basu and M. Meckesheimer. *Automatic outlier detection for time series: an application to sensor data*. Knowledge and Information Systems, 11(2):137–154, 2007.

[19] D. Cucina, A. Di Salvatore, and M. K. Protopapas. *Outliers detection in multivariate time series using genetic algorithms*. Chemometrics and Intelligent Laboratory Systems, 132:103–110, 2014.

[20] M. Gupta, J. Gao, C. Aggarwal, and J. Han. *Outlier detection for temporal data*. Synthesis Lectures on Data Mining and Knowledge Discovery, 5(1):1–129, 2014.

[21] D. Carrera, B. Rossi, P. Fragneto, and G. Boracchi. *Online anomaly detection for long-term ecg monitoring using wearable devices*. Pattern Recognition, 88:482–492, 2019.

[22] H. Mannila, H. Toivonen, and A. I. Verkamo. *Discovery of frequent episodes in event sequences*. Data Mining and Knowledge Discovery, 1(3):259–289, 1997.

[23] H. Mannila and H. Toivonen. *Discovering Generalized Episodes Using Minimal Occurrences.* In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, volume 96, pages 146–151, 1996.

[24] G. Casas-Garriga. *Discovering unbounded episodes in sequential data*. In European Conference on Principles of Data Mining and Knowledge Discovery, pages 83–94, 2003.

[25] N. Méger and C. Rigotti. *Constraint-based mining of episode rules and optimal window sizes*. In European Conference on Principles of Data Mining and Knowledge Discovery, pages 313–324, 2004.

[26] Xifeng Yan and Jiawei Han. *gSpan: graph-based substructure pattern mining*. In 2002 IEEE International Conference on Data Mining, pages 721–724, 2002.

[27] M. Kuramochi and G. Karypis. *Frequent subgraph discovery*. In Proceedings 2001 IEEE International Conference on Data Mining, pages 313–320, 2001.

[28] J. Huan, W. Wang, and J. Prins. *Efficient mining of frequent subgraphs in the presence of isomorphism*. In Third IEEE International Conference on Data Mining, pages 549–552, 2003.

[29] B. Bringmann and S. Nijssen. *What Is Frequent in a Single Graph?* In T. Washio, E. Suzuki, K. M. Ting, and A. Inokuchi, editors, Advances in Knowledge Discovery and Data Mining, pages 858–863, 2008.

[30] J. Chen and Y. Saad. *Dense Subgraph Extraction with Application to Community Detection*. IEEE Transactions on Knowledge and Data Engineering, 24(7):1216–1230, 2012.

[31] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli. *Denser than the Densest Subgraph: Extracting Optimal Quasi-Cliques with Quality Guarantees*. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 104–112, 2013.

[32] L. Qin, R.-H. Li, L. Chang, and C. Zhang. *Locally Densest Subgraph Discovery*. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 965–974, 2015.

[33] Y. Fang, K. Yu, R. Cheng, L. V. S. Lakshmanan, and X. Lin. *Efficient Algorithms for Densest Subgraph Discovery*. Proceedings of the VLDB Endowment, 12(11):1719–1732, 2019.

[34] Y. Chi, Y. Yang, and R. R. Muntz. *HybridTreeMiner: An efficient algorithm for mining frequent rooted trees and free trees using canonical forms*. In Proceedings of the16th International Conference on Scientific and Statistical Database Management, pages 11–20, 2004.

[35] Y. Chi, R. R. Muntz, S. Nijssen, and J. N. Kok. *Frequent subtree mining–an overview*. Fundamenta Informaticae, 66(1-2):161–198, 2005.

[36] F. Adriaens, J. Lijffijt, and T. De Bie. *Subjectively interesting connecting trees*. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 53–69, 2017.

[37] E. Marinari, G. Semerjian, and V. Van Kerrebroeck. *Finding long cycles in graphs*. Physical Review E, 75(6):066708, 2007.

[38] F. Adriaens, C. Aslay, T. De Bie, A. Gionis, and J. Lijffijt. *Discovering Interesting Cycles in Directed Graphs*. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pages 1191–1200, 2019.

[39] J. Gudmundsson and M. van Kreveld. *Computing Longest Duration Flocks in Trajectory Data*. In Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems, pages 35–42, New York, NY, USA, 2006.

[40] J. Gudmundsson, M. van Kreveld, and B. Speckmann. *Efficient detection of motion patterns in spatio-temporal data sets*. In Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems, pages 250–257, 2004.

[41] Z. Li, B. Ding, J. Han, and R. Kays. *Swarm: Mining relaxed temporal moving object clusters*. Proceedings of the VLDB Endowment, 3(1-2):723–734, 2010.

[42] H. Cao, N. Mamoulis, and D. W. Cheung. *Mining frequent spatio-temporal sequential patterns*. In the 5th IEEE International Conference on Data Mining, pages 8–pp, 2005.

[43] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. *Mining individual life pattern based on location history*. In the 10th International Conference on Mobile Data Management: Systems, Services and Middleware, pages 1–10, 2009.

[44] R. Song, W. Sun, B. Zheng, and Y. Zheng. *PRESS: A Novel Framework of Trajectory Compression in Road Networks*. Proceedings of the VLDB Endowment, 7(9), 2014.

[45] S. Liu, L. M. Ni, and R. Krishnan. *Fraud detection from taxis' driving behaviors*. IEEE Transactions on Vehicular Technology, 63(1):464–472, 2013.

[46] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li. *iBAT: detecting anomalous taxi trajectories from GPS traces*. In Proceedings of the 13th International Conference on Ubiquitous Computing, pages 99–108, 2011.

[47] K.-N. Kontonasios, E. Spyropoulou, and T. De Bie. *Knowledge discovery interestingness measures based on unexpectedness*. WIREs Data Mining and Knowledge Discovery, 2(5):386–399.

[48] R. Agrawal and R. Srikant. *Fast Algorithms for Mining Association Rules in Large Databases*. In Proceedings of the 20th International Conference on Very Large Data Bases, pages 487–499, 1994.

[49] X. Yin and J. Han. *CPAR: Classification based on predictive association rules*. In Proceedings of the 2003 SIAM International Conference on Data Mining, pages 331–335, 2003.

[50] N. Zhong, Y. Y. Yao, and S. Ohsuga. *Peculiarity Oriented Multi-database Mining*. In J. M. Żytkow and J. Rauch, editors, Principles of Data Mining and Knowledge Discovery, pages 136–146, 1999.

[51] Bing Liu, Wynne Hsu, Lai-Fun Mun, and Hing-Yan Lee. *Finding interesting patterns using user expectations*. IEEE Transactions on Knowledge and Data Engineering, 11(6):817–832, 1999.

[52] B. Padmanabhan. *A belief-driven method for discovering unexpected patterns*. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pages 94–100, 1998.

[53] S. Jaroszewicz and D. A. Simovici. *Interestingness of Frequent Itemsets Using Bayesian Networks as Background Knowledge*. In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 178–186, 2004.

[54] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. *Assessing Data Mining Results via Swap Randomization*. ACM Transactions on Knowledge Discovery from Data, 1(3):14–es, 2007.

[55] S. Hanhijärvi, M. Ojala, N. Vuokko, K. Puolamäki, N. Tatti, and H. Mannila. *Tell Me Something I Don't Know: Randomization Strategies for Iterative Data Mining*. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 379–388, 2009.

[56] T. De Bie. *An Information Theoretic Framework for Data Mining*. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 564–572, 2011.

[57] T. De Bie. *Subjective Interestingness in Exploratory Data Mining*. In Proceedings of the 12th International Symposium on Advances in Intelligent Data Analysis, pages 19–31, 2013.

[58] L. Geng and H. J. Hamilton. *Interestingness measures for data mining: A survey*. ACM Computing Surveys (CSUR), 38(3):9–es, 2006.

[59] J. Vreeken and N. Tatti. *Interesting Patterns*. In C. C. Aggarwal and J. Han, editors, Frequent Pattern Mining, pages 105–134. 2014.

[60] G. Grahne, L. V. S. Lakshmanan, and X. Wang. *Efficient Mining of Constrained Correlated Sets*. In Proceedings of the 16th International Conference on Data Engineering, pages 512–521, 2000.

[61] L. V. Lakshmanan, R. Ng, J. Han, and A. Pang. *Optimization of constrained frequent set queries with 2-variable constraints*. In Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, pages 157–168, 1999.

[62] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. *Discovering Frequent Closed Itemsets for Association Rules*. In Proceedings of the 7th International Conference on Database Theory, pages 398–416, 1999.

[63] S. Pool, F. Bonchi, and M. v. Leeuwen. *Description-driven community detection*. ACM Transactions on Intelligent Systems and Technology, 5(2):1–28, 2014.

[64] M. van Leeuwen, T. De Bie, E. Spyropoulou, and C. Mesnage. *Subjective interestingness of subgraph patterns*. Machine Learning, 105(1):41–75, 2016.

[65] N. Lavrač, B. Cestnik, D. Gamberger, and P. Flach. *Decision support through subgroup discovery: three case studies and the lessons learned*. Machine Learning, 57(1-2):115–143, 2004.

[66] J. Mata, J.-L. Alvarez, and J.-C. Riquelme. *Discovering numeric association rules via evolutionary algorithm*. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 40–51, 2002.

[67] N. S. Rai, S. Jain, and A. Jain. *Mining Interesting Positive and Negative Association Rule Based on Improved Genetic Algorithm (MIPNAR_GA)*. International Journal of Advanced Computer Science and Applications, 5(1), 2014.

[68] S. Zilberstein. *Using Anytime Algorithms in Intelligent Systems*. AI Magazine, 17:73–83, 1996.

[69] Q. Hu and T. Imielinski. *ALPINE: Anytime Mining with Definite Guarantees*. CoRR, abs/1610.07649, 2016. arXiv:1610.07649.

[70] A. Belfodil, A. Belfodil, and M. Kaytoue. *Anytime Subgroup Discovery in Numerical Domains with Guarantees*. In M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Ifrim, editors, Machine Learning and Knowledge Discovery in Databases, pages 500–516, 2019.

[71] A. Giacometti and A. Soulet. *Anytime algorithm for frequent pattern outlier detection*. International Journal of Data Science and Analytics, pages 1–12, 2016.

[72] A. Land and A. Doig. *An Automatic Method of Solving Discrete Programming Problems*. Econometrica, 28(3):497–520, 1960.

[73] Q. Louveaux and S. Mathieu. *A combinatorial branch-and-bound algorithm for box search*. Discrete Optimization, 13:36–48, 2014.

[74] E. Tomita and T. Kameda. *An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments*. Journal of Global optimization, 37(1):95–111, 2007.

[75] E. Tomita, Y. Sutani, T. Higashi, S. Takahashi, and M. Wakatsuki. *A simple and faster branch-and-bound algorithm for finding a maximum clique*. In International Workshop on Algorithms and Computation, pages 191–203, 2010.

[76] H. Cheng, X. Yan, J. Han, and S. Y. Philip. *Direct discriminative pattern mining for effective classification*. In 2008 IEEE 24th International Conference on Data Engineering, pages 169–178, 2008.

[77] J. Han, J. Pei, and Y. Yin. *Mining Frequent Patterns without Candidate Generation*. ACM SIGMOD Record, 29(2):1–12, 2000.

[78] J. Han and J. Pei. *Mining frequent patterns by pattern-growth: methodology and implications*. ACM SIGKDD Explorations Newsletter, 2(2):14–20, 2000.

[79] J. Pei, J. Han, and R. Mao. *CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets*. In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2000.

[80] Z. Ding, X. Ding, and Q. Wei. *An improved FP-Growth algorithm based on compound single linked list*. In The 4th International Conference on Information and Computing, volume 1, pages 351–353, 2009.

[81] M. Al Hasan and M. J. Zaki. *Output Space Sampling for Graph Patterns*. Proceedings of the VLDB Endowment, 2(1):730–741, 2009.

[82] M. Boley, C. Lucchese, D. Paurat, and T. Gärtner. *Direct Local Pattern Sampling by Efficient Two-Step Random Procedures*. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 582–590, New York, NY, USA, 2011.

[83] S. Moens and M. Boley. *Instant exceptional model mining using weighted controlled pattern sampling*. In International Symposium on Intelligent Data Analysis, pages 203–214, 2014.

[84] A. Giacometti and A. Soulet. *Dense neighborhood pattern sampling in numerical data*. In Proceedings of the 2018 SIAM International Conference on Data Mining, pages 756–764, 2018.

3

Subjective Interesting Motifs in Time series

3.1 Introduction

There exist a myriad of data mining methods for time series data, ranging from fully automated change detection, classification, and prediction methods, to exploratory techniques such as clustering and motif detection. Change and motif detection are related in the sense that local patterns (motifs) and longer-running changes in the profile of the time series need to be evaluated against a prior that specifies what the expected profile is, typically in the form of a probability distribution.

Prior work on time series motif detection tends to evaluate a motif's interestingness by assessing its significance against some objectively chosen prior distribution for the time series (either explicitly or implicitly). The result is that the most 'interesting' motifs found are often trivial, implied by the user's prior expectations. For example, given an Electrocardiogram (ECG) which records the electrical activity of a patient's heartbeat, a method that adopts an objective interestingness measure always identifies the same best motif (e.g., the one signalling the normal heartbeat), albeit for different users who typically hold different beliefs about this patient's heart condition. It is often the case that this best motif has already been known by some users, and thus is of little interest to them.

In contrast to this, we introduce an approach to identify recurring subsequence patterns that are *subjectively interesting*, i.e., interesting when contrasted with the

user's prior expectations. A recurring subsequence is a subsequence that is found at several positions within the time series with some variation, and will be called a *motif*.

To achieve this, we define subsequence patterns as local probabilistic models. The subjective interestingness of a subsequence pattern is then defined in terms of the amount of information (in an information-theoretic sense) contained in this local model, when contrasted with a *background distribution* that represents the user's expectations. Initially, the background distribution is computed as the distribution of maximum entropy subject to any prior user expectations as constraints, such as constraints on the expected mean, variance, and co-variance between neighboring points in the time-series. Upon revealing the presence of a subsequence pattern, the background distribution is updated to account for this new knowledge, such that it continues to represent the (now updated) expectations of the user as subsequence patterns are revealed throughout an iterative analysis. The amount of information gained by the time series can be computed by contrasting the prior distribution and the updated distribution.

To find the most informative motifs and outliers efficiently, we develop relaxations, and propose an effective search algorithm implemented in a constraint programming solver. Together with an additional heuristic pruning technique, this enables one to mine subsequence patterns relatively efficiently.

Our specific contributions are:

– Novel definitions of motifs as probabilistic patterns. [Sect. 3.3]

– A quantification of their Subjective Interestingness (SI), based on how much information a user gains when observing this pattern. [Sect. 3.4]

– A relaxation of the exact setting and an algorithm to efficiently mine the most interesting subsequence patterns to a user. [Sect. 3.5.1]

– Several speedup techniques which result in a computational more efficient algorithm [Sect. 3.5.2]

– An empirical evaluation of this algorithm on one synthetic data set and two real-world data sets, to investigate its ability to encode the user's prior beliefs and identify interesting subsequence patterns. [Sect. 3.6]

3.2 Related work

Time series motifs usually hint at useful information about seasonal or temporal associations between events, and detecting such patterns can be very useful in practice. A myriad of techniques for motif discovery have been proposed. These can be categorized from different perspectives, starting with the definition of the

interestingness measure for a motif. In general, two main aspects for judging the interestingness of a motif exist in literature, namely the similarity among instances and the support (i.e., the number of instances in a motif) [1]. More specifically, one prioritizes a motif whose instances exhibit maximum similarity, or even more strictly, defining a motif as the most similar subsequence pair (e.g., [2–4]); whereas the second prioritizes one with the highest support given a minimum similarity between all instances of a motif (e.g., [5, 6]).

For existing work adopting either similarity-based or support-based interestingness, the similarity measure plays a key role in the motif discovery algorithms, and typical ones include Euclidean distance and dynamic time warping. Regarding the massive computational cost, some efforts are made to representing time series in low dimensional space. Examples of such representations include Symbolic Aggregate approXimation (SAX), Discrete Fourier Transform (DFT), and random projections. A review of motif discovery algorithms based on their similarity measure and representation is provided by Mueen [1].

In addition to these aspects, there exist several challenging issues in this pattern discovery problem, including scalability [3, 7, 8], the detection of motifs with various lengths [9, 10], multi-dimensional time-series [11], coping with streaming data [12, 13] and handling distortions [14]. For a more comprehensive review of existing publications regarding these issues, we refer interested readers to Torkamani & Lohweg [15].

Our work explores a new aspect, shining light on the essence of the interestingness for a motif, which we believe depends on a user's prior knowledge. Previous measures that prioritize either the similarity or support are all objective. However, for a user with prior information about the time series (a common situation), the resulting motifs may be trivial. Hence, we propose a novel subjective interestingness measure, which enables one to identify motifs that contradict their prior expectations and are truly interesting to them. Additionally, the information-theoretic view that we take immediately provides a balance between the similarity and numerosity for a set of subsequences to form a motif.

3.3 Pattern syntaxes for motifs and motif templates

We denote a *time series* as $\hat{\mathbf{x}} \triangleq (\hat{x}_1, \ldots, \hat{x}_n)' \in \mathbb{R}^n$, i.e., an ordered collection of n real numbers $\hat{x}_i \in \mathbb{R}$, where $i \in [n] = [1, \ldots, n]$. We write $\hat{\mathbf{x}}_{i,l}$ for $\hat{\mathbf{x}}$ for the *subsequence* of length $l \leq n - i + 1$ starting from position i. That is, $\hat{\mathbf{x}}_{i,l} \triangleq (\hat{x}_i, \ldots, \hat{x}_{i+l-1})' \in \mathbb{R}^l$. By sliding a window of size l along $\hat{\mathbf{x}}$ and extracting each subsequence, we can obtain a set containing all the subsequences of length l. We denote this set as $\mathbb{S}_l$, i.e., $\mathbb{S}_l = \{\hat{\mathbf{x}}_{i,l} | i = 1, 2, \ldots, n - l + 1\}$. Note hatted symbols represent empirical values and their non-hatted equivalents are used to denote the respective random variables.

3.3.1 Motif

A *motif* of length l denoted by $\mathbb{T}_l$ is a subset of $\mathbb{S}_l$ containing more than 2 non-overlapping subsequences. That is, $\mathbb{T}_l \subseteq \mathbb{S}_l$, $|\mathbb{T}_l| \geq 2$, $|i-j| \geq l$, $\forall \hat{\mathbf{x}}_{i,l}, \hat{\mathbf{x}}_{j,l} \in \mathbb{T}$ and $i \neq j$.

Each subsequence in a motif is said to be an instance of the motif. As we focus on identifying motifs of a fixed length (i.e., l), we write $\mathbb{T}$ for $\mathbb{T}_l$ in the rest of the paper for convenience. Not every motif is equally interesting. The criterion by which we judge the quality of a motif is explained below.

The index set of a motif $\mathbb{T}$ is denoted as $\mathbb{I}_\mathbb{T}$, i.e., $\mathbb{I}_\mathbb{T} = \{i | \hat{\mathbf{x}}_{i,l} \in \mathbb{T}\}$.

3.3.2 Motif template

Our general aim is to find subjectively interesting 'motifs'. However, what one typically means is not actually a set of subsequences that are similar, but a general subsequence pattern that is reccurring in a time-series. To avoid working with a set of subsequences, one could use a single exemplar. Here we introduce a probabilistic local model as the target object, the *motif template*, instead.

Definition 1 (Motif template)**.** A *motif template* is a probability distribution over the space of motif instances, i.e., $\mathbb{R}^l$.

More concretely, we propose a template where we capture the mean and variance statistics of instances and call this as a *mean-variance motif template*. We deem the roles played by these two statistics essential, as the mean serves a figure about the motif shape and the variance tells the extent of the similarity among these instances. A typical choice of model is multivariate Gaussian distribution parameterized by the mean and variance statistics. It is in principle straightforward to also use covariance statistics, but such a model has $\mathcal{O}(l^2)$ parameters and is not interpretable. Thus, we define a *mean-variance motif template* as:

Definition 2 (Mean-variance motif template)**.** A *mean-variance motif template* is a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ over the space of motif instances. $\boldsymbol{\Sigma}$ is the diagonal matrix with the values of standard deviations as the main diagonal and zero elsewhere. Hence, this distribution can be essentially parameterized by a tuple $(\boldsymbol{\mu}, \boldsymbol{\sigma})$, where $\boldsymbol{\mu}$ is a vector of means and $\boldsymbol{\sigma}$ is a vector of standard deviations, both of length l.

In this paper, we take $\boldsymbol{\mu}, \boldsymbol{\sigma}$ as the maximum likelihood parameters over the set of instances in a motif. We denote the parameter tuple for motif template learned from the motif $\mathbb{T}$ as $(\boldsymbol{\mu}_\mathbb{T}, \boldsymbol{\sigma}_\mathbb{T})$. That is, $\boldsymbol{\mu}_\mathbb{T} = \frac{1}{|\mathbb{T}|} \sum_{i \in \mathbb{I}_\mathbb{T}} \hat{\mathbf{x}}_{i,l}$, $\boldsymbol{\sigma}_\mathbb{T} = \frac{1}{|\mathbb{T}|-1} \sum_{i \in \mathbb{I}_\mathbb{T}} (\hat{\mathbf{x}}_{i,l} - \boldsymbol{\mu}_\mathbb{T})^2$. Examples are given in Fig. 3.1, 3.2 and 3.3.

3.4 Formalizing the subjective interestingness

Previous motif discovery work tended to quantify the interestingness in an objective way (See Sect. 3.2). For a data analyst with prior knowledge about the time series, which we believe is common, the discovered patterns may be trivial to the end user and could be easily implied. To pre-empt this, we propose to use a more flexible subjective measure of interestingness.

3.4.1 The background distribution

We follow the so-called FORSIED[1] framework [16, 17] to quantify the subjective interestingness of a motif. The basic procedure is that a *background distribution* is defined over the space of all possible data sets, which here would be all possible realizations of a time series $\mathbf{x}$. Since $\mathbf{x} \in \mathbb{R}^n$, the background distribution is defined by a probability density function p. The background distribution essentially encodes the beliefs and expectations of the user about the data. More specifically, it assigns a probability density to each possible data value according to how tenable the user thinks this value to be. It was argued that a good choice for the background distribution is the maximum entropy distribution subject to constraints that capture the user's prior expectations about the data.

3.4.1.1 The initial background distribution

We wish to define constraints and compute a maximal entropy distribution such that these constraints are preserved in expectation. For the initial background distribution, we consider three kinds of constraints. They respectively express the user's prior knowledge about the mean and the variance of each data point, as well as the first order difference in $\mathbf{x}$. Notice these expectation values can be anything, here we equate them to the empirical values. With these three constraints, the initial background distribution is the solution to Prob. 3 as stated as follows

Problem 3.

$$\max_{p} \int -p(\mathbf{x}) \log(p(\mathbf{x}))\mathrm{d}\mathbf{x}, \tag{3.1}$$

$$\text{s.t.} \int p(\mathbf{x}) \frac{1}{n} \sum_{i=1}^{n} x_i \mathrm{d}\mathbf{x} = \hat{m}\mathbf{1}, \tag{3.2}$$

$$\int p(\mathbf{x}) \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{m})^2 \mathrm{d}\mathbf{x} = \hat{v}\mathbf{1}, \tag{3.3}$$

$$\int p(\mathbf{x}) \frac{1}{n-1} \sum_{i=1}^{n-1} (x_i - x_{i+1})^2 \mathrm{d}\mathbf{x} = \hat{d}\mathbf{1}, \tag{3.4}$$

[1] An acronym for 'Formalizing subjective interestingness in exploratory data mining'

$$\text{where } \hat{m} = \frac{1}{n}\sum_{i=1}^{n}\hat{x}_i,\ \hat{v} = \frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - \hat{m})^2, \hat{d} = \frac{1}{n-1}\sum_{i=1}^{n-1}(\hat{x}_i - \hat{x}_{i+1})^2.$$

and $\mathbf{1}$ is a n-dimensional vector with all the entries as 1.

The solution to Prob. 3 is a multivariate Gaussian distribution parameterised by a n-dimensional mean vector $\mathbf{m}$ and a $n \times n$ covariance matrix $\mathbf{V}$. The values of $\mathbf{m}$ and $\mathbf{V}$ can be derived by applying the Lagrange multiplier method. Also, we further improve the computation efficiency by using the property that maximizing the entropy and maximizing the likelihood are the dual of each other in the class of exponential form distributions [18]. The computation details are given in Appendix 3.A.

3.4.1.2 Updating the background distribution

Once a motif template along with its instances is identified and shown to the user, the user's belief state changes, and the background distribution needs to be updated. The background distributions p for all prior belief types discussed in this paper are essentially multivariate Gaussian distributions each of which is parametrized by $\mathbf{m}$ and $\mathbf{V}$. As mentioned, the motif template is also described by a multivariate Gaussian distribution, $\mathcal{N}(\boldsymbol{\mu}_{\mathbb{T}}, \boldsymbol{\Sigma}_{\mathbb{T}})$. To make the updated background distribution reflect the user's newly acquired knowledge, we simply set the blocks of current $\mathbf{m}$ and $\mathbf{V}$ corresponding to the subsequence instances equal to $\boldsymbol{\mu}_{\mathbb{T}}$ and $\boldsymbol{\Sigma}_{\mathbb{T}}$, and the off-diagonal elements of $\mathbf{V}$ corresponding to instances equal to 0. We denote the background distribution having incorporated $\mathbb{T}$ as $p_{\mathbb{T}}$.

3.4.2 A remark about no independence assumpation

Remark 1. *We do not assume independence between time points. While in the local motif model (i.e. the mean-variance motif template), time points are indeed independently distributed (see Sect. 3.3.2), this is not the case for the model of the whole time series* $\mathbf{x}$ *(indeed the full covariance matrix is not necessarily diagonal). Moreover, it is important to realize that the background distribution is a model for the user's belief state—it is not a model for the stochastic source of the data. In other words, if the background distribution does not exhibit a certain dependency, this does not mean that the data may not come from a stochastic source that exhibits this dependency. It only means that the user whose belief state is modelled by this background distribution is not yet aware of it. As the covariance matrix is not diagonal, it is indeed the case that updating the expected value even for a single point in the time series can ripple across the sequences and modify the expected values throughout.*

3.4.3 The subjective interestingness measure

Intuitively, a good motif is one whose instances are strongly similar to each other and together account for a considerable portion on the whole time series. Consider such a good motif $\mathbb{T}$. If all instances are similar to each other, it directly follows that the values of $\boldsymbol{\mu}_{\mathbb{T}}$ are similar to those of each instance, and the diagonal entries of $\boldsymbol{\Sigma}_{\mathbb{T}}$ are small. After revealing the motif to the user, the background distribution is updated to be $p_{\mathbb{T}}$. Since the parameters of $p_{\mathbb{T}}$ consist of $\boldsymbol{\mu}_{\mathbb{T}}$ and $\boldsymbol{\Sigma}_{\mathbb{T}}$, the new background distribution $p_{\mathbb{T}}$ will thus be a more accurate model for the time series. More precisely, the probability of the data under $p_{\mathbb{T}}$ is larger. To quantify the amount of information *gained* by the motif, we can compare this probability to the one under the previous background distribution p. The more strongly they differ, the more this motif enhances with the user's beliefs about the data.

Mathematically, we define the Information Content (IC) of a motif as the difference between the log probability for the whole time series $\hat{\mathbf{x}}$ under $p_{\mathbb{T}}$ and that under p:

$$\mathrm{IC}(\mathbb{T}) = \log p_{\mathbb{T}}(\hat{\mathbf{x}}) - \log p(\hat{\mathbf{x}}). \tag{3.5}$$

The rationale is that minus the log probability of the data represents the number of bits of information the data contains with respect to the probability distribution—so this difference corresponds to the amount of information (in bits) the user has gained by seeing the motif.

Note that the expected value of $\mathrm{IC}(\mathbb{T})$ w.r.t. $p_{\mathbb{T}}(\hat{\mathbf{x}})$ takes the same form as the Kullback-Leibler divergence, but this does not mean IC and KL-divergence are equivalent concepts. The KL-divergence measures the difference between two probability distributions, but here the $p_{\mathbb{T}}(\hat{\mathbf{x}})$ and $p(\hat{\mathbf{x}})$ in the definition of $\mathrm{IC}(\mathbb{T})$ are probabilities rather than distributions.

3.4.4 Finding the most subjectively interesting motif template

Now we can formalize our goal of finding the most interesting motif in a time series as an optimization problem with the following objective:

$$\text{Objective 1: } \operatorname*{argmax}_{\mathbb{T}} \log p_{\mathbb{T}}(\hat{\mathbf{x}}) - \log p(\hat{\mathbf{x}}).$$

Objective 1 accounts for the probability of the whole data. This probability depends on the parameter updating of p (i.e. $\mathbf{m}$ and $\mathbf{V}$) from incorporating subsequences, and can thus embody the quality of the choice for template instances. Note that the key changes of $\mathbf{m}$ and $\mathbf{V}$ only take place on part of their entries that represent instances in $\mathbb{T}$. That means, the rise in the probability of the whole data is mostly related to the probability of those instances in $\mathbb{T}$. Based on this observation, we propose a relaxed version of *Objective 1* which only depends on the

probability of instances in $\mathbb{T}$. This objective is similar to *Objective 1*, but is more straightforward to optimize efficiently.

$$\text{Objective 2: } \underset{\mathbb{T}}{\text{argmax}} \sum_{i \in \mathbb{I}_{\mathbb{T}}} \log p_{\mathbb{T}}(\hat{\mathbf{x}}_{i,l}) - \sum_{i \in \mathbb{I}_{\mathbb{T}}} \log p(\hat{\mathbf{x}}_{i,l}).$$

3.5 Algorithm

3.5.1 Mining algorithm

In this work, we adopted a greedy search algorithm to identify the most interesting motif. The general idea is to first seed $\mathbb{T}$ by finding a small set of k instances according to *Objective 2* and then greedily grow that set using *Objective 1*.

The algorithm consists of three major steps:

1. Model the user's prior belief by the initial background distribution;

2. Seed by finding a small set of instances which optimizes *Objective 2*;

3. Grow that set by adding an instance which optimizes *Objective 1*, and iterate.

Remark 2. Although the three basic steps are for finding a single motif (i.e. the most interesting one to the user), our algorithm is not limited to that. A new search for another motif can be triggered by running step 2 and step 3 again based on an updated background distribution, the one that has already incorporated the user's knowledge of the previous motif.

How we compute the initial background distribution (i.e., step 1) is described in the above (see Sect. 3.4.1.1). In the following, we go into more details of step 2 and 3.

3.5.1.1 Step 2: finding a seed motif $\mathbb{T}^{(0)}$ with k instances

The search starts by finding k non-overlapping optimal instances which constitute a seed set for $\mathbb{T}$. We denote such a seed set by $\mathbb{T}^{(0)}$. The most subjectively interesting $\mathbb{T}^{(0)}$ is identified by optimizing *Objective 2*. This problem can be formulated as

Problem 4.

$$\begin{aligned}
&\underset{\mathbb{T}^{(0)}}{\text{argmax}} \sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \log p_{\mathbb{T}^{(0)}}(\hat{\mathbf{x}}_{i,l}) - \sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \log p(\hat{\mathbf{x}}_{i,l}) \equiv \\
&\underset{\mathbb{T}^{(0)}}{\text{argmax}} \sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \log \mathcal{N}\left(\hat{\mathbf{x}}_{i,l}|\boldsymbol{\mu}_{\mathbb{T}^{(0)}}, \boldsymbol{\Sigma}_{\mathbb{T}^{(0)}}\right) \\
&\qquad - \sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \log \mathcal{N}\left(\hat{\mathbf{x}}_{i,l}|\mathbf{m}^{(i:i+l-1)}, \mathbf{V}^{(i:i+l-1,i:i+l-1)}\right), \qquad (3.6)
\end{aligned}$$

$$\text{where } \boldsymbol{\mu}_{\mathbb{T}^{(0)}} = \frac{1}{k}\sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \hat{\mathbf{x}}_{i,l}, \quad \boldsymbol{\Sigma}_{\mathbb{T}^{(0)}} = \text{diag}\left(\frac{1}{k-1}\sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \left(\hat{\mathbf{x}}_{i,l} - \boldsymbol{\mu}_{\mathbb{T}^{(0)}}\right)^2\right).$$

The superscript of a vector or matrix symbol is used to denote the corresponding entry. Using the expression for the multivariate Gaussian distribution, we can write Eqs. (3.6) as:

$$\begin{aligned}
&\sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \log \mathcal{N}(\hat{\mathbf{x}}_{i,l}|\boldsymbol{\mu}_{\mathbb{T}^{(0)}}, \boldsymbol{\Sigma}_{\mathbb{T}^{(0)}}) \\
&\quad - \sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \log \mathcal{N}\left(\hat{\mathbf{x}}_{i,l}|\mathbf{m}^{(i:i+l-1)}, \mathbf{V}^{(i:i+l-1,i:i+l-1)}\right) \\
=& -\frac{kl}{2}\log(2\pi) + \frac{kl}{2}\left\{\log k + \log(k-1)\right\} \\
&\underbrace{-\frac{k}{2}\sum_{h\in[l]} \log\left\{\sum_{\substack{i,j\in\mathbb{I}_{\mathbb{T}^{(0)}} \\ i<j}} (\hat{\mathbf{x}}_{i,l}^{(h)} - \hat{\mathbf{x}}_{j,l}^{(h)})^2\right\}}_{I} - \frac{1}{2}(k-1)l \\
&\underbrace{- \sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \log \mathcal{N}\left(\hat{\mathbf{x}}_{i,l}|\mathbf{m}^{(i:i+l-1)}, \mathbf{V}^{(i:i+l-1,i:i+l-1)}\right)}_{II}. \qquad (3.7)
\end{aligned}$$

Note the parts related to the choice of instances in $\mathbb{T}^{(0)}$ are underbraced and numbered respectively as I and II. By taking a closer look, we can see part II is essentially the sum of all the individual negative log probability of $\hat{\mathbf{x}}_{i,l}$ under p, and the values for parameters $\mathbf{m}$ and $\mathbf{V}$ do not subject to which instances to incorporate. This allows to gain some computational benefits by simply pre-computing each log probability. Nevertheless, I expresses a mutual relationship among all the instances in $\mathbb{T}^{(0)}$, due to it being in the summation form for the logarithm of a summation. Pre-computation is not trivial, which makes the search for optimal instances computationally demanding, reaching $\mathcal{O}(n^k k^2)$. We thus adopted a

strategy to mitigate a certain factor of this time complexity, as well as a heuristic to prune the search space. A detailed description is provided in Sect. 3.5.2.

3.5.1.2 Step 3: greedily searching for a new instance

The algorithm then continues to search for a new subsequence which optimizes *Objective 1*. The search stops when no new subsequence exists such that incorporating it can increase the probability of the time series under the background distribution, i.e., $\nexists i \in [n-l+1]$ s.t. $\mathbb{T} \cup \{x_{i,l}\}$ is a motif and $p_{\mathbb{T}\cup\{x_{i,l}\}}(\hat{\mathbf{x}}) - p_{\mathbb{T}}(\hat{\mathbf{x}}) \geq 0$.

To gain some speedup, we prune subsequences which pose little potential according to a heuristic (see Sect. 3.5.2).

3.5.2 Speedup techniques

In this section, we describe some speedup techniques applied to step 2 (Sect. 3.5.2.1) and step 3 (Sect. 3.5.2.2).

3.5.2.1 Speeding up the step 2

Strategy 1: Bounding the objective 2 and finding the submatrix with the maximal sum. Recall only term I and II in the objective of Prob. 4 (i.e. Eqs. (3.7)) are affected by the chosen of instances for $\mathbb{T}^{(0)}$, and Term I makes the search computationally expensive. To mitigate the time complexity, we consider optimizing a relaxed objective of Prob. 4 based on bounding the term I. Via applying Jensen's inequality [19], term I can be upper bounded by a summation form taken from all the instances pairs:

$$
\begin{aligned}
I : &- \frac{k}{2} \sum_{h\in[l]} \log \left\{ \sum_{\substack{i,j\in\mathbb{I}_{\mathbb{T}^{(0)}} \\ i<j}} (\hat{\mathbf{x}}_{i,l}^{(h)} - \hat{\mathbf{x}}_{j,l}^{(h)})^2 \right\} \\
&\leq \underbrace{\sum_{\substack{i,j\in\mathbb{I}_{\mathbb{T}^{(0)}} \\ i<j}} \left\{ -\frac{1}{k-1} \sum_{h\in[l]} \log \left\{ (\hat{\mathbf{x}}_{i,l}^{(h)} - \hat{\mathbf{x}}_{j,l}^{(h)})^2 \right\} \right\}}_{III} - \frac{kl}{2} \log \left\{ \frac{k(k-1)}{2} \right\}. \quad (3.8)
\end{aligned}
$$

Substituting Eqs. (3.8) into Eqs. (3.7) yields:

$$\begin{aligned}
&\sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \log\mathcal{N}\left(\hat{\mathbf{x}}_{i,l}|\boldsymbol{\mu}_{\mathbb{T}^{(0)}}, \boldsymbol{\Sigma}_{\mathbb{T}^{(0)}}\right) \\
&- \sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \log\mathcal{N}\left(\hat{\mathbf{x}}_{i,l}|\mathbf{m}^{(i:i+l-1)}, \mathbf{V}^{(i:i+l-1,i:i+l-1)}\right) \\
\leq &-\frac{kl}{2}\log(2\pi) + \frac{kl}{2}\left\{\log k + \log(k-1)\right\} \\
&+ \underbrace{\sum_{\substack{i,j\in\mathbb{I}_{\mathbb{T}^{(0)}} \\ i<j}} \left\{-\frac{1}{k-1}\sum_{h\in[l]}\log\{(\hat{\mathbf{x}}_{i,l}^{(h)} - \hat{\mathbf{x}}_{j,l}^{(h)})^2\}\right\}}_{III} - \frac{kl}{2}\log\left\{\frac{k(k-1)}{2}\right\} \\
&- \frac{1}{2}(k-1)l \underbrace{- \sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \log\mathcal{N}\left(\hat{\mathbf{x}}_{i,l}|\mathbf{m}^{(i:i+l-1)}, \mathbf{V}^{(i:i+l-1,i:i+l-1)}\right)}_{II}. \qquad (3.9)
\end{aligned}$$

Finding the maximal value for the objective Eqs. (3.9) is essentially the same as maximising term $III+$ term II. Then we construct a matrix $\hat{\mathbf{M}}$, with rows and columns representing subsequence candidates, the i-th diagonal entry $\hat{\mathbf{M}}_{i,i}$ being the part of the term II inside the summation (Eqs. (3.11) in the below), and the entry at the i-th row and j-th column $\hat{\mathbf{M}}_{i,j}$ being the part of the term III inside the outer summation (Eqs. (3.12) in the below).
Solving Prob. 4 corresponds to finding the upper triangular matrix inside $\hat{\mathbf{M}}$ with the maximum sum, as expressed in the following problem:

Problem 5.

$$\underset{\mathbb{T}^{(0)}}{\text{argmax}} \sum_{i\in\mathbb{I}_{\mathbb{T}^{(0)}}} \sum_{j\in\mathbb{I}_{\mathbb{T}^{(0)}}} \hat{\mathbf{M}}_{i,j}, \qquad (3.10)$$

$$\text{where } \hat{\mathbf{M}}_{i,i} = \log\mathcal{N}(\hat{\mathbf{x}}_{i,l}|\mathbf{m}^{(i:i+l-1)}, \mathbf{V}^{(i:i+l-1,i:i+l-1)}) \text{ for } i \in \mathbb{I}_{\mathbb{T}^{(0)}}, \qquad (3.11)$$

$$\hat{\mathbf{M}}_{i,j} = -\frac{1}{k-1}\sum_{h\in[l]}\log\{(\hat{\mathbf{x}}_{i,l}^{(h)} - \hat{\mathbf{x}}_{j,l}^{(h)})^2\} \text{ for } i,j \in \mathbb{I}_{\mathbb{T}^{(0)}}, \qquad (3.12)$$

$$\mathbb{T}^{(0)} \subseteq \text{PrunedSubsequenceSet}, \qquad (3.13)$$

$$|i-j| \geq l, \qquad \forall i,j \in \mathbb{I}_{\mathbb{T}^{(0)}} \text{ and } i \neq j. \qquad (3.14)$$

The fourth set of constraints (Eqs. (3.14)) is to ensure instances in $\mathbb{T}^{(0)}$ are non-overlapping to each other. This speedup technique enables us to compute the matrix $\hat{\mathbf{M}}$ in advance and then do the search using the constraint programming (CP). The time complexity of a relaxed Prob. 4 is $\mathcal{O}(n^k)$, a factor of k^2 less than

the Prob. 4. Clearly, it still appears intractable for real-world applications. To counter this, we deliberately reduce the search space so that each element of $\mathbb{T}^{(0)}$ is constrained to be in a pruned range, denoted by $PrunedSubsequenceSet$ (Eqs. (3.13)). The way we construct $PrunedSubsequenceSet$ is described in the following.

Strategy 2: Pruning. The exhaustive search for a solution to the relaxed *Problem* 4 is still computationally demanding for a large $\hat{\mathbf{M}}$. We thus adopt a heuristic strategy so that the search is among a considerably reduced space but the quality of the found motifs is guaranteed.

It appears that an off-diagonal entry at the i-th row and j-th column $\hat{\mathbf{M}}_{i,j}$ models a sort of similarity between the subsequence $\hat{\mathbf{x}}_{i,l}$ and $\hat{\mathbf{x}}_{j,l}$. As the transition property of the similarity suggests, if $\hat{\mathbf{M}}_{i,k}$ and $\hat{\mathbf{M}}_{j,k}$ are large, then does the $\hat{\mathbf{M}}_{i,j}$. We can deduce that all the entries in $\hat{\mathbf{M}}$ mapped from $\mathbb{I}_{\mathbb{T}^{(0)}}$ should have relatively larger value. Hence, we can deliberately perform the search in a pruned range of subsequences whose indices corresponding to largest entries in $\hat{\mathbf{M}}$. Specifically, we fix the first instance to be a certain subsequence and search the others among subsequences corresponding to the largest 1% entries at a row of $\hat{\mathbf{M}}$ corresponding to this instance (i.e. pruning factor $= 99\%$). To find the globally optimal k instances for $\mathbb{T}^{(0)}$, we fix the first instance to be each possible subsequence, and solve the relaxed *Problem* 4 each time. The final solution should be the one that leads to the maximal objective value.

3.5.2.2 Speeding up the step 3

In step 3, the exhaustive search for a new optimal instance requires checking the result of *Objective 1* value of incorporating every possible subsequence, which is apparently time-consuming for large $\hat{\mathbf{x}}$. Clearly, incorporating subsequences which bear strong similarity with instances in $\mathbb{T}^{(0)}$ can result in a high value for *Objective 1*. As the off-diagonal entries in $\hat{\mathbf{M}}$ encode a similarity between subsequence pairs, we apply a heuristic pruning strategy based on entries in $\hat{\mathbf{M}}$ to reduce the search space.

Assume we are in the stage of having incorporated all the k instances in $\mathbb{T}^{(0)}$. Let us denote the current $\hat{\mathbf{M}}$ as $\hat{\mathbf{M}}^k$. The new optimal instance must be among those that can produce a relatively large value of the objective for Prob. 5, but based on $\hat{\mathbf{M}}^{k+1}$, whose entry at the i-th row and j-th column $(i \neq j)$ is $-\frac{1}{k}\sum_{h\in[l]}\log\{(\hat{\mathbf{x}}_{i,l}^{(h)} - \hat{\mathbf{x}}_{j,l}^{(h)})^2\} = \frac{k-1}{k}\hat{\mathbf{M}}_{i,j}^k$ (recall $\hat{\mathbf{M}}_{i,j}^k$ is computed by Eqs. (3.12)). The objective for Prob. 5 is in the form of summing some entries of $\hat{\mathbf{M}}^{k+1}$ which correspond to instances in $\mathbb{T}^{(0)}$ and the new subsequence (e.g. $\hat{\mathbf{x}}_{r,l}$). Thus, the potential of $\hat{\mathbf{x}}_{r,l}$ (i.e., Potential$(\hat{\mathbf{x}}_{r,l})$) can be captured by how much the value

of objective for Prob. 5 (Eqs.(3.10)) increases if incorporating $\hat{\mathbf{x}}_{r,l}$:

$$\text{Potential}(\hat{\mathbf{x}}_{r,l}) = \sum_{i \in \mathbb{I}_{\mathbb{T}}(0)} \hat{\mathbf{M}}_{r,i}^{k+1} + \hat{\mathbf{M}}_{r,r}^{k+1}.$$

The algorithm ranks all possible subsequences in a descending order according to their Potential values. Then the search is implemented in a greatly reduced space (i.e. among those in the top 1% of the rank). Let us denote the optimal subsequence which leads to the highest *Objective 1* value by $\mathbf{T}_{k+1}$. We first check whether the probability of the time series increases under the new background distribution. If so, we include $\mathbf{T}_{k+1}$ in $\mathbb{T}$. Nevertheless, for incorporating the next subsequence, a further check is performed. First we update the search domain as well as the potential rank by deleting all the subsequences overlapped with $\mathbf{T}_{k+1}$. We do step 3 to identify the new optimal one. If this subsequence is still among the top 3 of the potential rank and incorporating it did not trigger the stop condition, we make it the $(k+2)$-th instance to $\mathbb{T}$. Otherwise, we recompute the potential and rank all the subsequences again, according to $\hat{\mathbf{M}}_{k+2}$, the one considering $\mathbf{T}_{k+1}$ as an incorporated instance. Then step 3 is done again among an updated search domain. However, there might occur a situation where the new optimal one is still not ranked among the top 3. In this case, if the stop condition is not reached, we make it an instance to $\mathbb{T}$ anyway. By this lazy greedy strategy, the search space is significantly reduced, while a good quality of the incorporated instance is ensured to a satisfiable extent.

3.6 Experiments

This section describes the evaluation of our proposed algorithm on a synthetic and two real-world datasets. In the following, we first describe the datasets (Sect. 4.6.1). Then we discuss the results of the conducted experiments that are directed at finding the answers to the following questions:

RQ1 Is our motif discovery algorithm sensitive to the pruning percentage in the initial set selection? (Sect. 3.6.2)

RQ2 How does our algorithm scale? (Sect. 3.6.2)

RQ3 Is our method able to identify subjectively interesting motifs, such that they are in contrast to what the user already knew? (Sect. 3.6.3)

All experiments were conducted on a PC with Ubuntu OS, Intel(R) Core(TM) i7-7700K 4.20GHz CPUs, and 32 GB of RAM. The main algorithm was implemented in Matlab R2016b. The step of identifying the initial motif template was

coded in Python 3.5, in which the open source software *OR-Tools* 6.10 [20] developed by Google was used as the constraint programming solver. All the computer codes are available at https://bitbucket.org/ghentdatascience/simit-public/src.

3.6.1 Data

- **Synthetic time series.** We synthesized a time series of length 15000. This series includes 2 sorts of motif trends, and their prototypes are taken from 2 subsequence instances in the UCR Trace Data [21]. Both instances are of the same length (i.e., 275), but belong to different classes. Subsequences for each motif are generated by sampling from a Gaussian distribution with the mean as the corresponding instance and a reasonably small variance as 0.01. There are in total 12 subsequences for each motif. The remaining are standard Gaussian noises, and they constitute a major part in the whole series. More details about the data synthesizing process are described in the pseudo code Procedure 3.1 in Appendix 3.B.

- **MIT-BIH arrhythmia ECG recording.** This data set is recording #205 in the MIT-BIH Arrhythmia DataBase [22]. This recoding was created from digitalizing the ECG signals at 360 samples per second. We chose a part of 20 seconds (7200 samples) to experiment on that includes normal heartbeats and ventricular tachycardia beats.

- **Belgium Power Load Data.** This data set is taken from *Open Power System Data* [23]. The primary source of this data is ENTSO-E Data Portal/Power Statistics [24]. *Open Power System Data* then resampled and merged the original data in a large CSV file with hourly resolution. The part we selected to experiment on records the total load in Belgium during the year 2007, for a total length of $24 * 365 = 8760$.

3.6.2 Pruning and Scalability (RQ1 and RQ2)

For all the experiments, we first identify an initial motif $\mathbb{T}^{(0)}$ with $k = 4$ instances. As mentioned above, our algorithm searches among a space pruned in a particular heuristic way to gain some relative amount of efficiency. The effects of pruning in the initial set identification were tested on the synthetic time series, for which the correct answers were known. The results indicate the optimal one was still found even with the heaviest prunning (99.9%) . Therefore, we used 99% pruning in the experiments on the real-world datasets. The scalability of our algorithm, with respect to the length of the motif template and to the length of the time-series, was evaluated on the ECG recording. *Tab*. 3.1 shows that the length of the motif template does not influence the computational cost that much, but the influence of the time-series length is more than quadratic.

Table 3.1: Run-time to search the initial motif set, with pruning factor 99%.

n	l	Time(s)	n	l	Time(s)	n	l	Time(s)
1800	100	9.96	3600	100	50.12	7200	100	369.92
7200	25	328.09	7200	50	350.65	7200	100	369.92

Summary. Overall, our algorithm is not sensitive to the pruning percentage. The run time is not influenced by the increasing length of the motif template that much, whereas it grows faster than quadratically with the increase of the length of the time series.

3.6.3 Results on the discovered motifs (RQ3)

3.6.3.1 Synthetic data

In this experiment, we specified the length of the motif instance same as the length of the subsequence synthesized by sampling (i.e. $l = 275$). As expected, our algorithm identified two motifs embedded in this synthetic time series, the result of which is shown in Fig. 3.1. The whole time series is plotted in Fig. 3.1(a), and subsequences incorporated into the first motif set are exactly those sampled from the same Gaussian distribution (in red). Fig. 3.1 (b) illustrates the first motif template by plotting the mean of all the instances incorporated into this motif as well as the error bars indicating the variance of each point. Our algorithm also correctly identifies the second motif (marked in green in Fig. 3.1(a)). We model the user's knowledge about the first motif by triggering the new search on a new original background distribution, the one that takes into account of all the instances for the first motif. The second motif is displayed in Fig. 3.1(c).

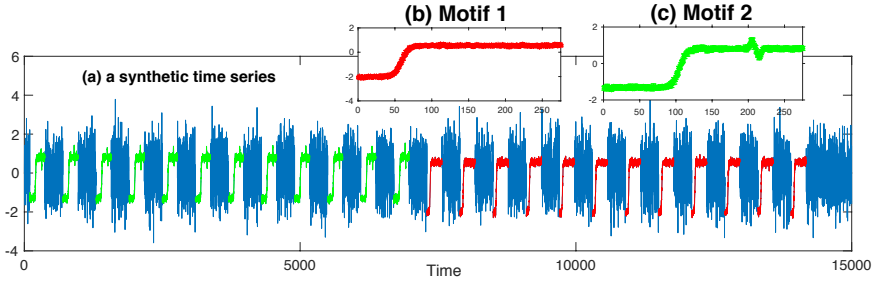


Figure 3.1: The algorithm correctly retrieves the two patterns in the synthetic data.

3.6.3.2 ECG time-series

We analyzed the ECG data by identifying motifs with length 100, corresponding to a duration of $0.28s$. In this fairly short recording (see Fig. 3.2a), our algo-

rithm identified three motifs. The first two motifs correspond to normal heartbeats (highlighted with red and green, templates shown in Fig. 3.2b and Fig. 3.2c). We see their shapes mostly coincide, with a horizontal shift. Normal heart beats are deemed to be similar to each other, but within each one there may exist a particular subsection that bear more similarity than other subsections. Since the motif length is set to be less than a period of a normal heart beat, our algorithm is prone to regard those subsections that bear the similarity to different extent to be in different motif sets. Another motif identified by the algorithm lies in the area of ventricular tachycardia (pink sections). The instances do not cover all the ventricular tachycardia heart beats, but the small error bars in Fig. 3.2d indicate that these instances are uncannily similar to each other, and the reason why other ventricular tachycardia subsequences lose the membership for this motif set is their smaller similarity.

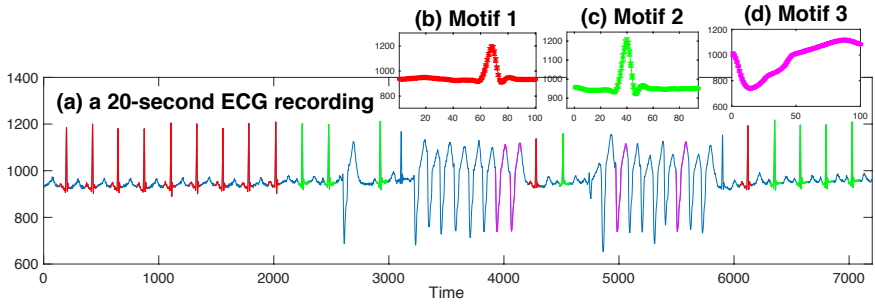


Figure 3.2: Three motif templates identified in the 20-second ECG recording.

3.6.3.3 Belgium Power Load Data

We analyzed this data searching for motifs of length 24 (one day). The first four motifs discovered by our algorithm are displayed in Fig. 3.3. The first motif covers many weekdays, except for Fridays, during cold seasons (highlighted with red in Fig. 3.3(a)). All these 24 hour periods start at 15:00 pm. Note not all the Monday to Thursday during these months are identified as the motif instance, for example, those blue sections both at the very beginning and the end of this whole series. The reason could be that they correspond to holidays rather than normal workdays. As for other workdays in winter excluding Friday that do not belong to this motif, these are very interesting for energy analyst to analyze the reason. After modelling user's knowledge about this motif, our algorithm then identified the second motif, corresponding to Monday to Thursday as well, but during hot seasons (highlighted with green in Fig. 3.3(a)). Most days in July are not instances of this motif. This might be due to them being in summer holiday time (a noticeable blue and pink section which divides the green section in Fig. 3.3 (a)). Actually, part of these days (i.e. Monday to Thursday in the last two weeks of July) constitute the third motif

(pink sections in Fig. 3.3(a)). The first 3 motifs are all related to normal workdays excluding Friday, but in different temperature conditions. It seems that power consumption in hot seasons is less regular than that during cold seasons, as the normal workday pattern relating to cold periods are identified first (i.e. the first motif). This phenomenon could be very interesting for energy analyst to investigate. By incorporating these 3 motifs into the user's belief model, our algorithm identified the fourth motif, corresponding to some Sunday time from middle of April to the beginning of October (black sections in Fig. 3.3(a)). All the instances belonging to the same motif corresponds to daytime starting at exactly the same hour, and they are strongly similar to each other, as reflected by the small error bars in the illustration of each motif template (Fig. 3.3(b)-(d)).

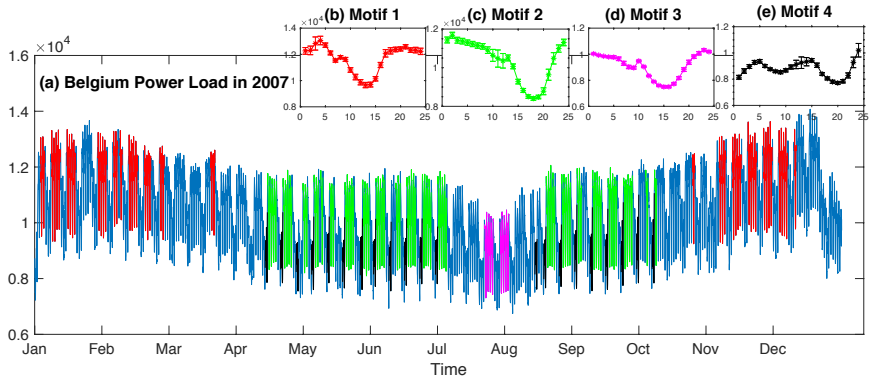


Figure 3.3: Four motif templates identified in the Belgium power load data.

Summary. As shown by these case studies on different datasets, our method can incorporate the user's newly acquired motifs into the background distribution for subsequent iterations, and identify motifs which contrast with this knowledge.

3.7 Conclusions

Subsequence patterns can provide valuable insights into both local and global characteristics of data. Because different users have different beliefs and prior knowledge about data, motifs ranked using statistically objective measures may not be of great interest to every user. We propose a new methodology for motif discovery and a concrete implementation for a specific type of motifs where the interestingness score can incorporate prior beliefs, and hence they are subjectively interesting. Although mining the most subjectively interesting motif appears intractable, we develop a relaxation of this interestingness score with bounds that can be optimized relatively efficiently using constraint programming. An

empirical evaluation demonstrates the potential of the proposed approach.

For future work, it would be useful to detect motifs that exhibit distortions (also known as *ill-known* motifs), e.g., those being stretched/squeezed, shifted in time, scaled in amplitude, or overlayed with noises. This can be approached by developing a motif template that incorporates a form of time warping. Secondly, the length of the subsequences considered is currently a parameter, and could be optimized as well. To make this possible, further speedup techniques should be developed. In contrast to motifs, *outliers* are subsequences that are unusual and non-reccuring in a time series. Identifying subjective interesting outliers can also be interesting. Moreover, the proposed motif templates are based on multivariate gaussian distributions. An extension to multivariate non-gaussian distribution [25] with the use of non-symmetrical entropy [26] seems promising. Finally, an extension towards multivariate time series is useful.

Acknowledgment

This work was supported by the ERC under the EU's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, FWO (project no. G091017N, G0F9816N), and the EU's Horizon 2020 research and innovation programme with the FWO under the MSC Grant Agreement no. 665501. We thank Panagiotis Papapetrou, Raúl Santos-Rodríguez, and Niall Twomey for their helpful input and discussions.

Appendices

3.A Solving problem 1

The solution of Prob. 3 follows directly from applying the method of *Lagrange Multipliers*. Let us use Lagrange multipliers $\lambda_1, \lambda_2, \lambda_3$ for constraints (Eqs. (3.2) to Eqs. (3.4)) respectively, $\boldsymbol{\lambda}$ for the vector containing all Lagrange multipliers.

The Lagrangian is formulated as:

$$\begin{aligned} L\left(\boldsymbol{\lambda}, p(\mathbf{x})\right) = &- \int p(\mathbf{x}) \log p(\mathbf{x}) \mathrm{d}\mathbf{x} \\ &+ \lambda_1 \left(\int p(\mathbf{x}) \frac{1}{n} \sum_{i=1}^{n} x_i \mathrm{d}\mathbf{x} - \hat{m}\mathbf{1} \right) \\ &+ \lambda_2 \left(\int p(\mathbf{x}) \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{m})^2 \mathrm{d}\mathbf{x} - \hat{v}\mathbf{1} \right) \\ &+ \lambda_3 \left(\int p(\mathbf{x}) \frac{1}{n-1} \sum_{i=1}^{n-1} (x_i - x_{i+1})^2 \mathrm{d}\mathbf{x} - \hat{d}\mathbf{1} \right). \end{aligned} \tag{3.15}$$

Differentiating Eqs. (3.15) w.r.t. $p(\mathbf{x})$ and renormalizing by dropping the $\mathrm{d}\mathbf{x}$ factor yields:

$$\begin{aligned} \frac{\partial}{\partial p(\mathbf{x}) \mathrm{d}\mathbf{x}} L\left(\boldsymbol{\lambda}, p(\mathbf{x})\right) = &- \log p(\mathbf{x}) - 1 \\ &+ \lambda_1 \left(\frac{1}{n} \sum_{i=1}^{n} x_i \right) + \lambda_2 \left(\frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{m})^2 \right) \\ &+ \lambda_3 \left(\frac{1}{n-1} \sum_{i=1}^{n-1} (x_i - x_{i+1})^2 \right). \end{aligned} \tag{3.16}$$

Equating Eqs. (3.16) to 0 and solving for $p(\mathbf{x})$ gives:

$$\begin{aligned} p(\mathbf{x}) = &\frac{1}{Z} \exp \left\{ \lambda_1 \left(\frac{1}{n} \sum_{i=1}^{n} x_i \right) + \lambda_2 \left(\frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{m})^2 \right) \right. \\ &\left. + \lambda_3 \left(\frac{1}{n-1} \sum_{i=1}^{n-1} (x_i - x_{i+1})^2 \right) \right\}. \end{aligned} \tag{3.17}$$

where Z is the normalization variable, as the implicit normalization constraint $\int p(\mathbf{x}) \mathrm{d}\mathbf{x} = 1$ can be imposed constructively by setting Z to be:

$$\begin{aligned} Z = \int \exp &\left\{ \lambda_1 \left(\frac{1}{n} \sum_{i=1}^{n} x_i \right) + \lambda_2 \left(\frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{m})^2 \right) \right. \\ &\left. + \lambda_3 \left(\frac{1}{n-1} \sum_{i=1}^{n-1} (x_i - x_{i+1})^2 \right) \right\} \mathrm{d}\mathbf{x}. \end{aligned}$$

By expanding quadratic terms in Eqs. (3.17) and reorganizing, we obtain:

$$p(\mathbf{x}) \propto \exp\Bigg\{(\lambda_2+\lambda_3)x_1^2+\sum_{i=2}^{n-1}(\lambda_2+2\lambda_3)x_i^2+(\lambda_2+\lambda_3)x_n^2 + \sum_{i=1}^{n}(\lambda_1-2\lambda_2\hat{m})x_i-\sum_{i=1}^{n-1}2\lambda_3 x_i x_{i+1}+\lambda_2 n\hat{m}^2\Bigg\}.$$

After some algebra:

$$p(\mathbf{x}) \propto \exp\left\{\frac{1}{2}(\mathbf{x}-\mathbf{m})^{\mathrm{T}}\mathbf{V}^{-1}(\mathbf{x}-\mathbf{m})\right\}. \tag{3.18}$$

where $\mathbf{m}=\hat{m}\mathbf{1}$,
$\mathbf{V}^{-1}_{1,1}=\mathbf{V}^{-1}_{n,n}=\lambda_2+\lambda_3, \mathbf{V}^{-1}_{i,i}=\lambda_2+2\lambda_3$ for $i=2,\ldots,n-1$
$\mathbf{V}^{-1}_{i,j}=-\lambda_3$ for $|i-j|=1, \mathbf{V}^{-1}_{i,j}=0$ for $|i-j|\geq 2$,
$\lambda_1=0$.

Clearly seen from Eqs. (3.18), $p(\mathbf{x})$ is essentially a multivariate Gaussian distribution with the mean vector $\mathbf{m}$ and the covariance matrix $\mathbf{V}$. Note the inverse of the covariance matrix $\mathbf{V}^{-1}$, known as the *precision matrix*, in our case is a tridiagonal matrix whose nonzero elements can be determined by λ_2 and λ_3.

As maximizing the entropy and maximizing the likelihood are the dual of each other in the class of exponential family [18], the optimal values of the Lagrange multipliers λ_2, λ_3 can be found by maximizing the likelihood over the observations $\mathcal{L}(\hat{\mathbf{x}}|\boldsymbol{\lambda})$. This problem can be formulated as:

Problem 6.

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}}\ \mathcal{L}(\hat{\mathbf{x}}|\boldsymbol{\lambda}).$$

Prob. 6 is convex and can be solved by using standard techniques for convex optimisation (e.g., the interior point method [27]).

3.B Pseudo code for generating the synthetic data

Algorithm 3.1: Synthetic time series generation

input : Trace instance 1, Trace instance 2
output: A synthesized time series $\hat{\mathbf{x}}$

$n \leftarrow 15000$ `// The length of the synthesized time series ;`
$l \leftarrow 275$`// The length of each subsequence in a motif whose prototype is taken from Trance instance 1 or 2;`
$\mathbf{S} \leftarrow$ An $n \times n$ diagonal matrix with each diagonal entry as 0.001 ;
$\mathbb{Q}_{\text{prototype1}} \leftarrow$ The set containing the beginning indices for 12 subsequences for prototype 1 ;
$\mathbb{Q}_{\text{prototype2}} \leftarrow$ The set containing the beginning indices for 12 subsequences for prototype 2 ;
$\mathbb{Q}_{\text{others}} \leftarrow$ The set containing indices which are not covered by subsequences for prototype 1 or 2 ;
`// Generating subsequences for prototype 1 by sampling`
for $i \in \mathbb{Q}_{\text{prototype1}}$ **do**
    $\hat{\mathbf{x}}_{i,l} \sim \mathcal{N}(\text{Trace instance } 1, \mathbf{S})$;
`// Generating subsequences for prototype 2 by sampling`
for $i \in \mathbb{Q}_{\text{prototype2}}$ **do**
    $\hat{\mathbf{x}}_{i,l} \sim \mathcal{N}(\text{Trace instance } 2, \mathbf{S})$;
`// Making the remaining standard Gaussian noises`
for $i \in \mathbb{Q}_{\text{others}}$ **do**
    $\hat{\mathbf{x}}_{i,1} \sim \mathcal{N}(0, 1)$

References

[1] A. Mueen. *Time series motif discovery: dimensions and applications*. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(2):152–159, 2014.

[2] A. Mueen, E. J. Keogh, Q. Zhu, S. Cash, and M. B. Westover. *Exact Discovery of Time Series Motifs*. In Proceedings of the 2009 SIAM international conference on data mining, 2009.

[3] C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh. *Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets*. In IEEE International Conference on Data Mining, pages 1317–1322, 2016.

[4] A. Mueen and N. Chavoshi. *Enumeration of time series motifs of all lengths*. Knowledge and Information Systems, 45(1):105–132, 2015.

[5] J. Lin, E. Keogh, S. Lonardi, and P. Patel. *Finding Motifs in Time Series*. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 53–68, 2002.

[6] B. Chiu, E. Keogh, and S. Lonardi. *Probabilistic Discovery of Time Series Motifs*. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 493–498, 2003.

[7] T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. *Searching and mining trillions of time series subsequences under dynamic time warping*. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 262–270, 2012.

[8] C. E. Yoon, O. O'Reilly, K. J. Bergen, and G. C. Beroza. *Earthquake detection through computationally efficient similarity search*. Science Advances, 1(11), 2015.

[9] P. Senin, J. Lin, X. Wang, T. Oates, S. Gandh, A. P. Boedihardjo, C. Chen, and S. Frankenstein. *GrammarViz 3.0: Interactive Discovery of Variable-Length Time Series Patterns*. The ACM Transactions on Knowledge Discovery from Data, 12(1):10:1–10:28, 2018.

[10] M. Linardi, Y. Zhu, T. Palpanas, and E. Keogh. *Matrix Profile X: VALMOD - Scalable Discovery of Variable-Length Motifs in Data Series*. In SIGMOD, pages 1053–1066, 2018.

[11] C. M. Yeh, N. Kavantzas, and E. Keogh. *Matrix Profile VI: Meaningful Multidimensional Motif Discovery*. In IEEE International Conference on Data Mining, pages 565–574, 2017.

[12] A. Mueen and E. Keogh. *Online Discovery and Maintenance of Time Series Motifs*. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1089–1098, 2010.

[13] J. Lin and Y. Li. *Finding approximate frequent patterns in streaming medical data*. In IEEE the 23rd International Symposium on Computer-Based Medical Systems, pages 13–18, 2010.

[14] E. Keogh, L. Wei, X. Xi, S. Lee, and M. Vlachos. *LB_Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures*. In Proceedings of the 32nd International Conference on Very Large Data Bases, pages 882–893, 2006.

[15] S. Torkamani and V. Lohweg. *Survey on time series motif discovery*. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7(2):e1199, 2017.

[16] T. De Bie. *An information-theoretic framework for data mining*. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 564–572, 2011.

[17] T. De Bie. *Subjective interestingness in exploratory data mining*. In Proceedings of the 12th International Symposium on Advances in Intelligent Data Analysis, pages 19–31, 2013.

[18] T. De Bie. *Maximum entropy models and subjective interestingness: an application to tiles in binary databases*. Data Mining and Knowledge Discovery, 23(3):407–446, 2011.

[19] J. L. W. V. Jensen. *Sur les fonctions convexes et les inégalités entre les valeurs moyennes*. Acta Mathematica, 30(1):175–193, 1906.

[20] Google. *Google Optimization Tools(OR-Tools)*. https://github.com/google/or-tools.

[21] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. *The UCR Time Series Classification Archive*, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.

[22] G. B. Moody and R. G. Mark. *The impact of the MIT-BIH Arrhythmia Database*. IEEE Engineering in Medicine and Biology Magazine, 20(3):45–50, 2001.

[23] Open power system data. *Data Package Time series. Version 2018-03-13*, 2018.

[24] ENTOSO-E. *Detailed hourly load data for all countries 2006-2015*. https://www.entsoe.eu/data/data-portal/, 2015.

[25] J. E. Contreras-Reyes. *Renyi entropy and complexity measure for skew-gaussian distributions and related families*. Physica A: Statistical Mechanics and its Applications, 433:84 – 91, 2015.

[26] C. shi Liu. *Nonsymmetric entropy and maximum nonsymmetric entropy principle*. Chaos, Solitons, Fractals, 40(5):2469 – 2474, 2009.

[27] F. A. Potra and S. J. Wright. *Interior-point methods*. Journal of Computational and Applied Mathematics, 124(1):281 – 302, 2000.

4

Explainable Local and Global Subgraph Patterns with Surprising Densities

4.1 Introduction

Real-life graphs (also known as networks) often contain attributes for the vertices. In social networks for example, where vertices correspond to individuals, vertex attributes can include the individuals' interests, education, residency, and more. The connectivity of the network is usually highly related to those attributes [1–4]. The attributes of individuals affect the likelihood of them meeting in the first place, and, if they meet, of becoming friends. Hence, it appears likely it should be possible to understand the connectivity of a graph in terms of those attributes, at least to a certain extent.

One approach to identify the relations between the connectivity and the attributes is to train a link prediction classifier, with as input the attribute values of a vertex pair, predicting the edge as present or absent (e.g., [5–8]). Such global models often fail to provide insight though. To address this, the local pattern mining community introduced the concept of *subgroup discovery*, where the aim is to identify subgroups of data points for which a target attribute has homogeneous and/or outstanding values [9, 10]. Such subgroup rules are local patterns, in that they provide information only about a certain part of the data.

Research on local pattern mining in attributed graphs has so far focused on

identifying dense vertex-induced subgraphs, dubbed *communities*, that are coherent also in terms of attributes. There are two complementary approaches, as stated in [11]. The first explores the space of communities that meet certain criteria in terms of density, in search for those that are also homogeneous with respect to some of the attributes (e.g., [12, 13]) The second explores the space of rules over the attributes, in search for those that define subgroups (of vertices) that form a dense community (e.g., [11, 14, 15]). This is effectively a subgroup discovery approach to dense subgraph mining.

Limitations of the state-of-the-art. Both these approaches hinge on the existence of attribute homophily in the network: the tendency of links to exist between vertices with similar attributes [2]. Yet, while the assumption of homophily is often reasonable, it limits the scope of application of prior work. A *first limitation* of the state-of-the-art is thus its inability to find e.g. sparse subgraphs.

A *second limitation* is the fact that the interestingness of such patterns has invariably been quantified using objective measures—i.e., measures that do not depend on the user's prior knowledge. Yet, the most 'interesting' patterns found are often obvious and implied by such prior knowledge (e.g., communities involving high-degree vertices, or in a student friendship network, communities involving individuals practicing the same sport). Not only may uninteresting patterns appear interesting if prior knowledge is ignored, also interesting patterns may appear uninteresting and are hence not found. E.g., a pattern in a student friendship network that indicates tennis lovers are rarely connected may be due to the lack of suitable facilities or a tennis club.

A *third limitation* of prior work is that the patterns describe only the connectivity within a single group and not between two potentially distinct groups. As an obvious example, this excludes patterns that describe friendships between a particular subgroup of female and a subgroup of male individuals in a social network, but as we will show in the experiments real-life networks contain many less obvious examples.

Contributions. We depart from the existing literature in formalizing a subjective interestingness measure, rather than an objective one, and this for sparse as well as for dense subgraph patterns. In this way, we overcome the first and second limitations of prior work discussed above. More specifically, we build on the ideas from the exploratory data mining framework FORSIED [16, 17]. This framework stipulates in abstract terms how to formalize the subjective interestingness of patterns. Basically, a *background distribution* is constructed to model prior beliefs the user holds about the data. Given that, one can identify patterns which strongly contrast to this background knowledge and are highly surprising to the user. Moreover, this interestingness measure is naturally applicable for patterns describing a pair of subgroups, to which we will refer as *bi-subgroup patterns*. Hence, our method overcomes the third limitation of prior work. Finally, apart from a local pattern

mining strategy which is used to identify interesting patterns one by one, we also propose a strategy to mine patterns globally, that is, to summarize the whole graph in a meaningful way such that all the interesting patterns can immediately be seen. The resulting summarization can be considered as a type of global pattern. Our specific contributions are:

- Novel definitions of single-subgroup patterns and bi-subgroup patterns, as well as patterns that are global summaries for attributed graphs. [Sec. 4.3]
- A quantification of their Subjective Interestingness (SI), based on what prior beliefs a user holds, or what information a user gains when observing a pattern. [Sec. 4.4]
- An algorithm to mine bi-subgroup patterns based on beam search. [Sec. 4.5]
- An algorithm to mine global (or summarization) patterns from which a series of interesting single-subgroup and bi-subgroup patterns can be revealed. [Sec. 4.5]
- An empirical evaluation of our method on real-world data, to investigate its ability to encode the user's prior beliefs and identify subjective interesting patterns. [Sec. 4.6]

4.2 Related work

To grapple with graph pattern mining, state-of-the-art techniques either design those three key building blocks (i.e., pattern syntaxes, interestingness measures, and mining algorithms) dedicated to graph types, or resort to strategies that can represent a graph into a flat vector form (a.k.a., *network embedding*) such that pattern mining methods for traditional flat data can be applied. In this section, we provide a literature review of dedicated graph pattern mining (Sec. 4.2.1 for plain graphs, Sec. 4.2.2 for attributed graphs), and network embedding (Sec. 4.2.3). Lastly, we briefly review some graph modelling work (Sec. 4.2.4), which provides the foundation of model fitting, model selection, for various applications including the pattern mining.

4.2.1 Pattern mining in plain graphs

In this brief review of pattern mining work related to plain graphs (denoted as $G = (V, E)$ where V is a set of vertices, $E \subseteq \binom{V}{2}$ is a set of unweighted and undirected edges), we focus on two most well-known and extensively studied problems: dense subgraph mining and frequent subgraph mining.

Dense subgraphs. Dense subgraphs often indicate importance and discovering them can be very useful for numerous applications. Nowadays, graph mining research abounds with dense subgraph mining methods. Different methods are aimed with different applications, and find different definitions of *dense subgraph* useful.

In general, there exist two classes of density definitions: absolute density and relative density [18] [1]. Methods applying absolute density aim to identify a dense substructure that satisfies some particular rules, e.g., *cliques* (i.e., fully-connected subgraphs) [19, 20], *density-based quasi-cliques* (i.e., subgraphs with connectivity density larger than a threshold τ) [21, 22], *degree-based quasi-clique* (i.e., subgraphs where each vertex connects to at least τ percent of other vertices) [23–25], *k-core* (i.e., subgraphs where each vertex connects to at least k other vertices) [26], *k-plex* (i.e., subgraphs where each vertex is missing not more than k connections to others) [27, 28], *k-club* (i.e., subgraphs where the shortest path between any two vertices is not more than k) [29, 30] so forth. In contrast, methods applying relative density identify a set of top subgraphs with respect to a certain interestingness measure. Here, whether a subgraph is presented as output to the user depends on other subgraphs, and is thus relative. Common interestingness measures for dense subgraphs include *density* [31], *average degree* [32, 33], *modularity* [34], edge surplus [35] among others. We provide a detailed description of these measures in Sec. 4.6.3.1 and Appendix 4.A.

Frequent subgraphs. Frequent subgraph mining (FSM) deals with identifying frequently occurring subgraphs that occur no less than a specified threshold in a given set of graphs or a single large graph. As far as we know, existing methods for this task all use *occurrence of frequency* as the interestingness measure. Therefore, the following review of state-of-the-arts is made along two perspectives that correspond to two other building blocks of a pattern mining framework: pattern syntaxes and mining algorithms.

Designing pattern syntaxes for a FSM task is all about the graph or subgraph representation. Most methods use an *adjacency matrix* (e.g., *AGM* [36], *Subdue* [37]) or an *adjacency list* to represent graphs. Nevertheless, neither of them can take into account graph isomorphism, which means a set of graphs isomorphic to each other may not share the same adjacency matrix or adjacency list. In other words, they cannot uniquely identify a graph. Recently, *canonical labelling* is proposed to provide an isomorphism-invariant representation for a graph. Two approaches of canonical labelling are commonly employed in FSM. The first one is *canonical adjacency matrix (CAM)*. The CAM code is a string concatenation of

[1] These two classes of density definitions correspond to constraint-based pattern mining and preference-based pattern mining respectively (which are two categories of pattern mining previously introduced in Sec 2.1 of Chapter 2)

the upper or lower triangular entries of the adjacency matrix. To ensure the unique representation, the adjacency matrix must be the one that gives the minimum or maximum canonical code with respect to the lexicographic order. FSM methods using CAM include *FSG* [38], *FFSM* [39], *HSIGRM* and *VSIGRAM* [40]. The other canonical labelling approach is *minimum depth-first search (DFS) code*. The idea is to first construct a set of sequential DFS codes each of which is obtained by traversing the given graph in a DFS-fashion, and then assign the minimum DFS code with respect to the lexicographic order to be the canonical representation of the graph. Methods such as *gSpan* [41], *CloseGraph* [42], *p-gSpan* [43] and *GERM* [44] use this approach.

Now we adopt a perspective of mining algorithms. In general, existing FSM algorithms can be divided into two categories: Apriori-based and pattern growth-based[2]. FSM Apriori-based algorithms (e.g., *AGM* [36], *FSG* [38], Path# [45]) first generate candidates based on breadth-first search (BFS) i.e., each candidate subgraph of a larger size is generated by merging two subgraphs with smaller size, and thus proceeding to the next larger size requires the complete enumeration of all candidates of the current size. Then they compute the frequencies of subgraphs by applying subgraph isomorphism testing, which is a well-known NP-complete problem [46] and is thus very costly. Pattern growth-based approaches (e.g., *SPIN* [47], *MOFA* [48], *gSpan* [41], *FFSM* [39], and *Gaston* [49]) follow a depth-first search (DFS) fashion to generate candidates, i.e., extending a frequent subgraph by adding one extra edge at every time until they are still frequent. This extension often needs to meet some particular criterias (e.g., adding edge only at the rightmost path) to avoid the generation of redundant candidates. We refer to [50] for a comprehensive and structured survey on FSM.

4.2.2 Pattern mining in attributed graphs

Real-life graphs often have attributes on the vertices. Pattern mining considering both the structural aspect and the attribute information promises more meaningful and accurate results, and thus has received increasing research attention. In this section, we give a more extensive literature review of mining attributed graphs, as methods proposed in this chapter are dedicated to this particular structure. This review is along two dimensions, concerning local patterns (Sec. 4.2.2.1) and global patterns (Sec. 4.2.2.2) respectively.

4.2.2.1 Local pattern mining

Dense subgraphs. Research on local pattern mining in attributed graphs focuses on identifying dense vertex-induced subgraphs (also known as *communities* or *co-*

[2]These two categories are also covered in previous context where we describe existing mining algorithms from a broader view (Sec. 2.2.3 of Chapter 2).

hesive patterns) that also show high similarity according to their attribute values. Existing methods generally exhibit two types: one emphasizes the graph structure, considering attributes as complementary information for restricting the possible subgraph sets; whereas the other emphasizes the attribute, aiming to mine descriptive patterns for obtaining subgraph candidates evaluated by graph structural property.

In the family of the first type, one of the first approaches is proposed by Moser et al. [12]. They define a *cohesive pattern* as a subgraph H that satisfies three constraints: (1) connectivity constraint—i.e., H is connected, (2) density constraint—i.e., the edge density of H exceeds a user-defined *density threshold*, and (3) subspace cohesion constraint—i.e., its corresponding vertices exhibit homogeneity in the attribute space w.r.t. the user-defined *subspace cohesion threshold* (to enforce homogenous feature values) and *dimensionality threshold* (to enforce sufficiently large subspace). They also propose an algorithm, called CoPam, to efficiently find all cohesive patterns that are *maximal* (i.e., neither its corresponding vertex set nor attribute subspace is part of that for any other cohesive pattern).

Mougel et al. [13] consider graphs with Boolean attributes associated to each vertex, and propose to find *Collection of Homogeneous k-clique Percolated components (CoHoP)*, i.e., a union of at least γ cliques of size k connected by overlaps of $k-1$ vertices with all its vertices having in common more than α true-valued attributes (where γ, k and α are all user-defined parameters). For this task, they also give a sound and complete algorithm based on the subgraph enumeration.

Now we look at existing work of the second type. As an example, Silva et al. [51] focus on finding attribute sets that explain the formation of dense subgraphs through correlation, a task they call *structural correlation pattern mining*. More specifically, given an attributed graph and four user-defined parameters (i.e., σ_{min}, δ_{min}, η_{min}, min_size), this task consists of identifying the set of *structural correlation patterns* (S, Q), such that S is an attribute set whose induced vertex set is with the size larger than σ_{min} and whose dependence to the density of the associated vertices (called *structural correlation measure*) is larger than δ_{min}, and Q is a quasi-clique from the graph induced by S where a *quasi-clique* parameterized by η_{min} and min_size is a vertex set with more than min_size vertices such that each vertex is connected at least to a fraction η_{min} of the others. They also formalize a measure based on statistical significance to access the interestingness of structural correlation patterns, and propose an efficient algorithm to mine them.

Diverse top-k descriptive community mining, introduced by Pool et al. [14] aims to identify a diverse set of k (possibly overlapping) cohesive communities that have a concise description in the vertices' attribute space. Towards this aim, they propose a heuristic algorithm based on alternating between two phases: (1) finding top-quality communities, (2) inducing concise descriptions for them. For evaluating the quality of communities, they propose a cohesiveness measure based

on counting erroneous links (i.e., connections that are either missing or obsolete w.r.t. the 'ideal' community given the induced subgraph). To a limited extent, their method can be driven by the user's domain-specific background knowledge which is a preliminary description or a set of vertices expected to be part of a community. Then the search is triggered by those seed candidates. Our proposed SI, in contrast, is more versatile in a sense that allows incorporating more general background knowledge.

Galbrun et al. [15]'s work focuses on a similar task to Pool et al.'s, but theirs relies on a different density measure which is essentially the average degree, and a different mining algorithm which is based on a generic greedy scheme with three variants.

Atzmueller et al. [11] introduce description-oriented community detection which is with a similar target to the aforementioned Galbrun et al.'s and Pool et al.'s. They apply a subgroup discovery approach to mine patterns in the description space so it comes naturally that the identified communities have a succinct description—this is the main distinguishing feature of their method to the other two.

All previous works quantify the interestingness in an objective manner, in the sense that they cannot consider a user's prior beliefs and thus operate regardless of context. Also, all previous works focus on a set of communities or dense subgraphs, overlooking other meaningful structures such as a sparse or dense subgraph between two different subgroups of vertices.

Proximity patterns. Frequent subgraphs and frequent itemsets often fail to capture fuzzy patterns due to their inelastic pattern definition, and in particular, mining frequent subgraphs in large graph space is often excessively inefficient due to the complexity of isomorphism testing. To overcome these issues, Khan et al. [52] depart from the traditional concept of frequent subgraphs and frequent itemsets, and introduce the novel *proximity pattern*, i.e., a subset of labels that frequently appear in multiple tightly connected subgraphs in a labelled graph. It's essentially a frequent itemset, but considering an itemset as a union of labels for a set of tightly-connected vertices instead of merely individual labels (in this case, it is a traditional itemset mining problem). They also propose a complete pipeline to mine proximity patterns in massive graphs in a scalable manner.

Exceptional patterns. Unlike dense subgraphs and proximity patterns which represent regularities within a large graph, *exceptional subgraphs* are subgraphs with distinguishing features from others, and thus represent peculiarity. As an example, Bendimerad et al. [53] introduce the novel task of mining *connected* subgraphs whose vertices share some *distinguishing characteristics* (i.e., unusually large or small numerical values on a subset of their attributes) from the rest of the

graph, and propose a complete algorithm together with a sampling approach for this task. In their later work [54], they propose to mine subgraphs that are with distinguishing characteristics, but are also *cohesive* such that all vertices in the same subgraph are at a bounded distance to a set of certain vertices (named *core*) with potentially few exceptions—they call such subgraphs *Cohesive Subgraphs with Exceptional Attributes* (CSEA patterns). One innovative feature of this work is using a subjective interestingness measure. Both our local pattern mining work and theirs formalize the subjective interestingness of patterns based on De Bie's framework [16, 17]. The difference is the problem setting: theirs focus on exceptional attribute values, whereas ours focus on exceptional densities.

4.2.2.2 Global pattern mining

Discovering global patterns that can uncover useful insights in attributed graphs are typically tailored to a graph summarization or a clustering task. Although these two tasks can both output graph summaries, their goals (even when solely considering the structural aspect) are fundamentally different. Graph summarization seeks to group together vertices that connect with the rest of the graph in a similar way, while clustering simply group vertices that are densely connected to each other and are well separated from other groups [55].

Graph summarization. Most existing method handle graph summarization by incorporating *database-style functionalities*. For example, Tian et al. [56] propose two database-style operations, *SNAP* (Summarization by grouping Nodes on Attributes and Pairwise relationships) and *k-SNAP*, for controlled and intuitive graph summarization. Like OLAP (Online Analytical Processing) [57, 58], a popular tool in the traditional database systems that allows users to interactively view and analyze the database from different perspectives and with multiple granularities, their proposed operations provide an analogous functionality for the graph data: *SNAP* can produce customized summaries based on user-selected attributes and relationships that are of interest, and *k-SNAP* can allow the user to control the resolutions of the resulting summaries.

Then Zhang et al. [59] further build on *k-SNAP* by addressing two key limitations. First, they allow automatic categorization of numeric attributes (which is a common scenario). Second, they propose a measure to access the interestingness of summaries so that the user does not have to manually inspect a large number of summaries to find the interesting ones. However, their interestingness measure is not subjective, simply considering the tradeoff among diversity, coverage and conciseness.

Chen et al. [60] develop a *graph OLAP* framework which allows to analyze the graph dataset in an OLAP manner (i.e., presenting a multi-dimensional and multi-level view over graphs). This framework naturally provides a graph summary based on the selected attributes and the given input information. Another

graph summarization work from them [61] is *SUMMARIZE-MINE*, a framework that summarizes the original graphs into small summaries which are then mined for frequent subgraphs mining. Also, *SUMMARIZE-MINE* can tackle the potential pattern loss issue effectively and efficiently: false positives (i.e., subgraphs frequent in the summarized graph but not frequent in the original graph) can be verified on the resulting summarization, and false negatives (i.e., subgraphs frequent in the original graph but not frequent in the summarized graph) can be recovered as *SUMMARIZE-MINE* generates randomised summaries for multiple iterations (in this way, a lossy compression can be effectively turned into a virtually lossless one).

Another common category of graph summarization methods rely on concepts and techniques in information theory. For example, many of them [62–64] leverage *Minimum Description Length (MDL)*, i.e., a model selection principle of choosing the minimum description of the data as the best model [65]. For graph summarization, applying MDL is formulated as to minimize the description length of the model, plus the description length of the graph given the model M (i.e. the compression loss). A key advantage of applying MDL is its automatic optimization of the number of vertex groups or attributes groups, which are usually required to be user-specified in aforementioned database-style graph summarization methods.

In addition to pattern discovery, graph summarization on attributed graphs can serve for several applications including compression [63, 66], query answering [67, 68], influence analysis [69, 70] and so on. For a more in depth review of existing publications regarding these goals, we refer interested readers to a survey paper by Liu et al. [55].

Graph clustering. Prior methods of clustering attributed graphs seek to partition the given graph into clusters with cohesive intra-cluster structures and homogeneous attribute values. Some enforce homogeneity in all attributes [71–73]. However, they are not guaranteed to reveal meaningful patterns in datasets without efforts of attribute selection, since irrelevant attributes can strongly obfuscate clusters.

More recently, Gunnemann et al. [74, 75] loosen this constraint by *subspace clustering*, i.e., a method that finds clusters in different subspaces [76]. In attributed graph data, this allows ones to consider subsets of attributes to determine similarity between vertices of the same cluster. Perozzi et al. [77] detect *focused* clusters and outliers based on user preferences, allowing the user to control the relevance of attributes and as a consequence, the graph mining results. Wang et al. [78] propose a novel nonnegative matrix factorization (NMF) model in which the sparsity penalty is introduced to select the most relevant attributes for each cluster. Many other methods resort to deep learning techniques, as one of their biggest advantages is the ability to execute feature engineering by themselves. For graph clustering task, they are often based on learning an embedding and consist

of two steps: first, deep learning is used to learn a compact representation in a form of the node embedding considering both the structural relationship and vertex attributes information, and then a classic clustering methods like k-means or the spectral clustering algorithm is applied upon such embedding (see Sec. 4.2.3 for a brief review of existing network embedding methods). Nevertheless, because the learnt embedding is not dedicated to the subsequent clustering task, this mismatch may produce unsatisfying clustering performance. In contrast to this two-step embedding learning method, Wang et al. [79] propose a unified attributed graph clustering framework that can learn the graph embedding and perform clustering simultaneously. Though the application of deep learning eliminates the need to do feature engineering, one downside is its missing explainability, i.e., it is hard to trace back which features have contributed to the output.

Unlike all previous graph summarization or clustering methods where the resulting vertex groups are forced to satisfy some pre-specified topologies or edges structures (e.g., being more densely connected within the group), patterns revealed in our summarization approach are not limited to that, as their interestingness is quantified by a subjective measure depending on the user's prior expectation.

4.2.3 Network embedding

What makes graph data powerful and distinguishable is its ability to model the relationship between data objects—or in other words, a graph can represent each object in terms of other objects (imaging each row of the corresponding adjacency matrix). Not surprisingly, this also makes pattern mining on this special kind of data more challenging. Typically, existing graph mining methods either construct pattern syntaxes, interestingness measures and mining algorithms that are dedicated to graph data type (as described above), or intentionally fill the gap between graph data and classic pattern mining methods for flat tabular data. In the latter case, *network embedding*, a technique that represents a network's nodes to vectors in an embedding space while preserving their properties, is exactly the gap filler.

Recently, network embedding has gained massive research attention, and plentiful various methods have been proposed. In general, these methods can be divided into four broad categories: (1) factorization based; (2) random walk based; (3) deep learning based; (4) probability theory based.

Given a graph, *factorization based* methods (e.g., *Locally Linear Embeddings (LLE)* [80], *Laplacian Eigenmaps (LE)* [81], *Graph Factorization (GF)* [82], *GraRep* [83], *HOPE* [84]) first represent vertex similarity in a matrix form (such as the adjacency matrix or its polynomials, the incidence matrix, Laplacian matrix, the node transition probability matrix and Katz similarity matrix, among others), and then apply matrix factorization to generate a low-dimensional embedding.

Random walk based methods (e.g., *DeepWalk* [85], *Node2vec* [86], *Struc2vec*

[87], HARP [88]) rely on some random walk strategies to estimate the probability that a node pair co-occur on a random walk. Then the embedding is obtained by optimizing the likelihood of these random walk statistics. Compared to factorization based methods, *random walk* based ones are more expressive and more efficient: they encode the vertex similarity in a stochastic manner which allows to incorporate both local and higher-order proximity information, and they only need to consider co-occurring vertex pairs instead of all pairs.

Deep learning is well-known for its ability to efficiently learn non-linear function. With no doubt, network embedding methods, which aim to learn a complicated and highly non-linear mapping function between network space and low-dimensional network space, can benefit from utilizing deep learning. Examples of this category include *Structural Deep Network Embedding (SDNE)* [89], *PRUNE* [90], *VERSE* [91].

Finally, there are some *probabilistic* embedding methods (e.g., *LINE* [92], *CNE* [93]). In particular, *CNE* can obtain an embedding that is optimal with respect to the user's prior knowledge about the network.

For comprehensive studies and evaluations of different network embedding methods, we refer interested readers to [94, 95].

4.2.4 Graph modelling

Graph modelling typically considers a given network (i.e., the one we observe) as merely a realization among a large number of possibilities. All possible realizations including the observed one that are consistent with some given aggregate statistics, form the so-called *statistical ensemble of networks*.

A well-founded probabilistic framework to such graph modelling is provided by exponential random graph models (ERGMs) [96, 97]. In ERGMs, each graph has a probability that depends on a number of chosen statistics of the network. Such models allow one to sample random graphs that match certain graph properties as closely as possible, without the need to know the underlying network generation process [98]. Nevertheless, a downside of ERGMs is their intractable fitting on large, finite networks. Recently, Casiraghi et al. introduce a broad class of analytically tractable statistical ensembles of finite, directed and weighted networks, referred to as *generalized hypergeometric ensembles* [99].

Unlike ERGMs that aim to be an accurate and objective probabilistic model for the data, the aim of our method is to provide the user with subjectively interesting insights into the data. To do that, intelligible pattern syntaxes need to be designed to represent the data's local or global information. Secondly, the found patterns must be contrasted with a model of the user's belief state about the data (called *the background distribution*) to quantify their interestingness to the user (this makes our approach a subjective one). A further distinction from ERGMs is that our

method is naturally an iterative method, allowing the user to gain new insights from one or a few patterns at a time.

4.3 Pattern syntaxes for graphs

In this section we introduce both single subgroup and bi-subgroup patterns along with summaries for graphs. Here, we first introduce some notation.

An attributed graph is denoted as a triplet $G = (V, E, A)$ where V is a set of $n = |V|$ vertices, $E \subseteq \binom{V}{2}$ is a set of $m = |E|$ undirected edges,[3] and A is a set of attributes $a \in A$ defined as functions $a : V \to \text{Dom}_a$, where Dom_a is the set of values the attribute can take over V. For each attribute $a \in A$ with categorical Dom_a and for each $y \in \text{Dom}_a$, we introduce a Boolean function $s_{a,y} : V \to \{\text{true}, \text{false}\}$, with $s_{a,y}(v) \triangleq$ true for $v \in V$ iff $a(v) = y$. Analogously, for each $a \in A$ with $\text{Dom}_a \subseteq \mathbb{R}$ and for each $l, u \in \text{Dom}_a$ such that $l < u$, we define $s_{a,[l,u]} : V \to \{\text{true}, \text{false}\}$, with $s_{a,[l,u]}(v) \triangleq$ true iff $a(v) \in [l, u]$. We call these Boolean functions *selectors*, and denote the set of all selectors as S. A *description* or *rule* W is a conjunction of a subset of selectors: $W = s_1 \wedge s_2 \ldots \wedge s_{|W|}$. The *extension* $\varepsilon(W)$ of a rule W is defined as the subset of vertices that satisfy it: $\varepsilon(W) \triangleq \{v \in V | W(v) = \text{true}\}$. We also informally refer to the extension as the *subgroup*. Now a *description-induced subgraph* can be formally defined as:

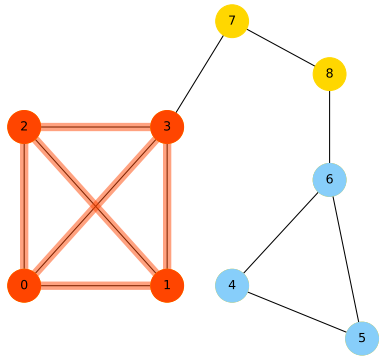
Definition 3. (Description-induced-subgraph) *Given an attributed graph $G = (V, E, A)$, and a description W, we say that a subgraph $G[W] = (V_W, E_W, A)$ where $V_W \subseteq V, E_W \subseteq E$, is induced by W if the following two properties hold,*

(i) $V_W = \varepsilon(W)$, i.e., the set of vertices from V that is the extension of the description W;

(ii) $E_W = \binom{V_W}{2} \cap E$, i.e., the set of edges from E that have both endpoints in V_W.

Example 3. *Fig. 4.1 displays an example attributed graph $G = (V, E, A)$ with $n = 9$ vertices, $m = 12$ edges (Graph in Fig. 4.1(a), vertex attributes in Fig. 4.1(b)). Each vertex is annotated with one real-valued attribute (i.e., a) and three nominal (or for simplicity, binary) attributes (i.e., b,c,d). Consider a description $W = s_{a,[0,3]} \wedge s_{b,1}$. The extension of this description is the set of vertices with attribute a value from 0 to 3 and attribute b as 1, i.e., $\varepsilon(W) = \{0, 1, 2, 3\}$. The subgraph induced by W is formed from $\varepsilon(W)$ and all the edges connecting pairs of vertices in that set (highlighted with red in Fig. 4.1(a)).*

[3] We consider undirected graphs without self-edges for the sake of presentation and consistency with most literature. However, we note that all our results can be easily extended to directed graphs and graphs with self-edges.



(a) Graph.

Vertex	a	b	c	d
0	0.6	1	0	1
1	1.5	1	0	0
2	2.2	1	0	1
3	1.4	1	1	1
4	2.5	0	0	0
5	4.7	0	0	0
6	4.3	0	0	0
7	3.6	1	1	1
8	3.8	1	1	0

(b) Vertex attributes.

Figure 4.1: Example attributed graph with 9 vertices (0-8) and 4 associated attributes (a-d). The subgraph induced by the description ($W = s_{a,[0,3]} \wedge s_{b,1}$) is highlighted in red.

4.3.1 Local pattern

4.3.1.1 Single-subgroup pattern

A first pattern syntax we consider, and which has already been studied in prior work, informs the user about the density of a description-induced subgraph $G[W]$. We assume the user is satisfied by knowing whether the density is unusually small, or unusually large, and given this does not expect to know the precise density. It thus suffices for the pattern syntax to indicate whether the density is either smaller than, or larger than, a specified value. We thus formally define the *single-subgroup* pattern syntax as a triplet (W, I, k_W), where W is a description and $I \in \{0, 1\}$ indicates whether the number of edges E_W in subgraph $G[W]$ induced by W is greater (or less) than k_W. Thus, $I = 0$ indicates the induced subgraph is dense, whereas $I = 1$ characterizes a sparse subgraph. The maximum number of edges in $G[W]$ is denoted by n_W, equal to $\frac{1}{2}|\varepsilon(W)|(|\varepsilon(W)| - 1)$ for undirected graphs without self-edges. One example of a single-subgroup pattern in Fig. 4.1 can be $(s_{a,[0,3]} \wedge s_{b,1}, 0, 6)$, corresponding to the dense subgraph highlighted in red.

Remark 3. (Difference to dense subgraph pattern in [100]) *Though the syntax for our single-subgroup pattern seems similar to that of the dense subgraph pattern (i.e., (W, k_W)) proposed by [100], they are essentially different definitions serving for different data mining tasks. In [100], the aim is to identify subjectively interesting subgraphs based on merely link information. For this aim, W in the dense subgraph pattern syntax represents the set of vertices in the subgraph, which has no association with node attributes. Moreover, an indicator I is included in our pattern syntax. This allows to regard not only surprisingly dense subgraphs but also surprisingly sparse ones as interesting. In contrast, [100] focuses on those*

surprisingly dense subgraphs. Because of these differences in W and I, k_W is different accordingly.

4.3.1.2 Bi-subgroup pattern

We also define a pattern syntax informing the user about the edge density between two potentially different subgroups. More formally, we define a *bi-subgroup pattern* as a quadruplet (W_1, W_2, I, k_W), where W_1 and W_2 are two descriptions, and $I \in \{0, 1\}$ indicates whether the number of connections between $\varepsilon(W_1)$ and $\varepsilon(W_2)$ is upper bounded (1) or lower bounded (0) by the threshold k_W. The maximum number of connections between the extensions $\varepsilon(W_1)$ and $\varepsilon(W_2)$ is denoted by $n_W \triangleq |\varepsilon(W_1)||\varepsilon(W_2)| - \frac{1}{2}|\varepsilon(W_1 \wedge W_2)|(|\varepsilon(W_1 \wedge W_2)| + 1)$ for undirected graphs without self-edges. For example, the bi-subgroup pattern $(s_{a,[0,3]} \wedge s_{b,1}, s_{b,0}, 1, 0)$ in Fig. 4.1, expresses sparse (or more precisely, zero) connection between the red vertex group (i.e., $\{0, 1, 2, 3\}$) and the blue one (i.e., $\{4, 5, 6\}$) . Note that single-subgroup patterns are a special case of bi-subgroup patterns when $W_1 \equiv W_2$.

Remark 4. (Setting of k_W) *Although k_W for a pattern (W_1, W_2, I, k_W) can be any value with which the number of connections between $\varepsilon(W_1)$ and $\varepsilon(W_2)$ (or within $\varepsilon(W_1)$ when $W_1 \equiv W_2$) are bounded, our work focuses on identifying patterns whose k_W is the actual number of connections between these two subgroups (or within this single subgroup when $W_1 \equiv W_2$), as such patterns are maximally informative.*

4.3.2 Global pattern: summarization for graphs

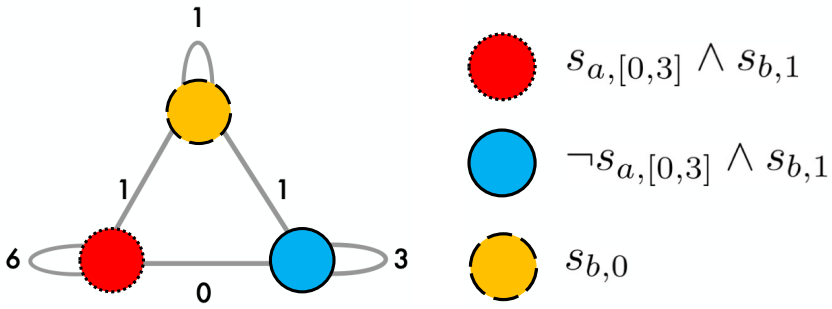
Here we define a global pattern syntax, which describes the edge density between any pair of subgroups selected from a set of subgroups that form a partition of the vertices. We first define the notion of a *summarization rule*, before introducing the global pattern syntax itself.

Definition 4. (Summarization rule for an attributed graph) *Given an attributed graph $G = (V, E, A)$,* the summarization rule $\mathbb{S}$ *of G is a set of descriptions such that their extensions are vertex-clusters that form a partition of the whole vertex set. That is, $\mathbb{S} = \{W_i | i = 1, 2, \ldots, c\}$ where $c \in \mathbb{N}$ is the number of disjoint vertex-clusters, where $\cup_{i=1}^{c} \varepsilon(W_i) = V$, $\forall W_i \in \mathbb{S}$ it holds that $\varepsilon(W_i) \neq \emptyset$, and $\forall W_i, W_j \in \mathbb{S}, i \neq j$ it holds that $\varepsilon(W_i) \cap \varepsilon(W_j) = \emptyset$.*

Definition 5. (Summary for an attributed graph based on a summarization rule) *A summary $\mathcal{S}$ for an attributed graph $G = (V, E, A)$ based on a summarization rule $\mathbb{S} = \{W_i | i = 1, 2, \ldots, c\}$ is* a complete weighted graph $\mathcal{S} = (V^{\mathbb{S}}, E^{\mathbb{S}}, w)$ *with weight function $w : E^{\mathbb{S}} \to \mathbb{R}$, whereby $V^{\mathbb{S}} = \{\varepsilon(W) | W \in \mathbb{S}\}$ is the set of*

vertices (referred to as supervertices *of the original graph G, i.e. each vertex from $\mathcal{S}$ is a set of vertices from G), $E^{\mathbb{S}} = \binom{V^{\mathbb{S}}}{2} \cup V^{\mathbb{S}}$ is the set of edges (to which we refer as* superedges*; the superedges in $\binom{V^{\mathbb{S}}}{2}$ represent the undirected edges between distinct supervertices, and the superedges in $V^{\mathbb{S}}$ represent the self-loops). The weight $w(\{\varepsilon(W_i), \varepsilon(W_j)\})$ for each superedge $\{\varepsilon(W_i), \varepsilon(W_j)\} \in E^{\mathbb{S}}$ will be denoted shorthand by $d_{i,j}$, and is defined as the number of edges between vertices from $\varepsilon(W_i)$ and those from $\varepsilon(W_j)$.*

We define a global pattern syntax informing the user about the summarization for an attributed graph $G = (V, E, A)$ with c disjoint vertex-clusters. More formally, we define a *summarization pattern* as a tuple $(\mathbb{S}, \mathcal{S})$ where $\mathbb{S}$ is the summarization rule, and $\mathcal{S}$ is the corresponding summary. Note that when revealing a summarization pattern $(\mathbb{S}, \mathcal{S})$ to a user, she or he gets access to its related local subgroup patterns: c single-subgroup patterns and $c(c-1)/2$ bi-subgroup patterns. An example of the global pattern for Fig. 4.1 can be $(\{s_{a,[0,3]} \wedge s_{b,1}, \neg s_{a,[0,3]} \wedge s_{b,1}, s_{b,0}\}, \mathcal{S}^*)$ where $\mathcal{S}^*$ represents the corresponding summary (see Fig. 4.2a).



(a) Summary $\mathcal{S}^*$.

(b) Descriptions in the summarization rule.

Figure 4.2: Example summarization pattern for Fig. 4.1 with the summarization rule $\{s_{a,[0,3]} \wedge s_{b,1}, \neg s_{a,[0,3]} \wedge s_{b,1}, s_{b,0}\}$ and the summary $\mathcal{S}^*$. This summary $\mathcal{S}^*$ is composed of three supervertices each of which corresponds to a set of vertices satisfying $s_{a,[0,3]} \wedge s_{b,1}$ (red circle with dotted line), $\neg s_{a,[0,3]} \wedge s_{b,1}$ (blue circle with solid line), $s_{b,0}$ (yellow circle with dashed line) respectively, and superedges each of which connects one supervetex to the other with a weight representing the number of edges between them.

4.4 Formalizing the subjective interestingness

4.4.1 General approach

We follow the approach as outlined by [101] to quantify the subjective interestingness of a pattern, which enables us to account for prior beliefs a user may

hold about the data. In this framework, the user's belief state is modeled by a *background distribution* P over the data space. This background distribution represents any prior beliefs the user may have by assigning a probability (density) to each possible value for the data according to how plausible the user thinks this value is. As such, the background distribution also makes it possible to evaluate the probability for any given pattern to be present in the data, and thus to assess the surprise of the user when informed about its presence. It was argued that a good choice for the background distribution is the maximum entropy distribution subject to some particular constraints that represent the user's prior beliefs about the data. As the user is informed about a pattern, the knowledge about the data will increase, and the background distribution will change. For details see Sec. 4.4.2.

Given a background distribution, the *Subjective Interestingness* (SI) of a pattern can be quantified as the ratio of the *Information Content* (IC) and the *Description Length* (DL) of the pattern. The IC is defined as the amount of information gained when informed about the pattern's presence, which can be computed as the negative log probability of the pattern w.r.t. the background distribution P. The DL is quantified as the length of the code needed to communicate the pattern to the user. These are discussed in more detail in Sec. 4.4.3, but first we further explain the background distribution (Sec. 4.4.2).

Remark 5. (Positioning with respect to directly related literature) *Here we clarify how previous work is leveraged, and what concepts are newly introduced in our work. We define single/bi-subgroup patterns and global patterns in an attributed graph. To quantify the SI measure for such patterns, we follow the framework outlined by [101]. As mentioned above, in this framework, the SI is computed as the ratio of the IC and the DL w.r.t. the background distribution which models the user's belief state. This framework also provides the general idea for deriving the initial background distribution and updating it to reflect newly acquired knowledge. [102] later introduced a new type of graph-related prior that the background distribution can incorporate, and this prior is considered in our work. In [100], this framework was used to identify subjectively interesting dense subgraphs, merely based on link information. In our work, we leverage some computational results from [100] (i.e., in updating the background distribution, approximating the IC), and made further adaptions such that the framework proposed by [101] can serve for our newly proposed patterns based on attribute information (i.e., single-subgroup patterns, bi-subgroup patterns and global patterns).*

4.4.2 The background distribution

4.4.2.1 The initial background distribution

To derive the initial background distribution, we need to assume what prior beliefs the user may have. Here we discuss three types of prior beliefs which are common

in practice: (1) on individual vertex degrees; (2) on the overall graph density; (3) on densities between bins (particular subsets of vertices).

(1–2) *Prior beliefs on individual vertex degrees and on the overall graph density.* Given the user's prior beliefs about the degree of each vertex, [101] showed that the maximum entropy distribution is a product of independent Bernoulli distributions, one for each of the random variable $b_{u,v}$, which equals to 1 if $(u, v) \in E$ and 0 otherwise. Denoting the probability that $b_{u,v} = 1$ by $p_{u,v}$, this distribution is of the form:

$$P(E) = \prod_{u,v} {p_{u,v}}^{b_{u,v}} \cdot (1 - p_{u,v})^{1-b_{u,v}},$$
$$\text{where} \quad p_{u,v} = \frac{\exp(\lambda_u^r + \lambda_v^c)}{1 + \exp(\lambda_u^r + \lambda_v^c)}.$$

This can be conveniently expressed as:

$$P(E) = \prod_{u,v} \frac{\exp((\lambda_u^r + \lambda_v^c) \cdot b_{u,v})}{1 + \exp(\lambda_u^r + \lambda_v^c)}.$$

The parameters λ_u^r and λ_v^c can be computed efficiently. For a prior belief on the overall density, every edge probability $p_{u,v}$ simply equals the assumed density.

(3) *Additional prior beliefs on densities between bins.* We can partition vertices in an attributed graph into bins according to their value for a particular attribute. For example, vertices representing people in a university social network can be partitioned by class year. Then expressing prior beliefs regarding the edge density between two bins is possible. This would allow the user to express, for example, an expectation about the probability that people in class year y_1 are connected to those in class year y_2. If the user believes that people in different class years are less likely to connect with each other, a discovered pattern would be more informative if it contrasts more with this kind of belief, i.e. if it reveals a high density between two sets of people from different class years. As shown in [102], the resulting background distribution is also a product of Bernoulli distributions, one for each of the random variables $b_{u,v} \in \{0, 1\}$:

$$P(E) = \prod_{u,v} \frac{\exp((\lambda_u^r + \lambda_v^c + \gamma_{k_{u,v}}) \cdot b_{u,v})}{1 + \exp(\lambda_u^r + \lambda_v^c + \gamma_{k_{u,v}})}, \tag{4.1}$$

where $k_{u,v}$ is the index for the block corresponding to the intersecting part of two bins which vertex u and vertex v belongs to correspondingly. λ_u^r, λ_v^c and $\gamma_{k_{u,v}}$ are parameters and can be computed efficiently. Note our model is not

limited to incorporate this type of belief related to a single attribute. Vertices can be partitioned differently by another attribute. Our model can consider multiple attributes so that users could express prior beliefs regarding the edge densities between bins resulting from multiple partitions[4].

4.4.2.2 Updating the background distribution

Upon being represented with a pattern, the background distribution should be updated to reflect the user's newly acquired knowledge. The beliefs attached to any value for the data that does not contain the pattern should become zero. In the present context, once we present a subgroup pattern (W_1, W_2, I, k_W) to the user, the updated background distribution P' should be such that $\phi_W(E) \geq k_W$ (if $I = 0$) or $\phi_W(E) \leq k_W$ (if $I = 1$) holds with probability one, where $\phi_W(E)$ denotes a function counting the number of edges between $\varepsilon(W_1)$ and $\varepsilon(W_2)$. [16] presented an argumentation for choosing P' as the *I-projection* of the previous background distribution onto the set of distributions consistent with the presented pattern. Then [100] showed that the resulting P' is again a product of Bernoulli distributions:

$$P'(E) = \prod_{u,v} {p'_{u,v}}^{b_{u,v}} \cdot (1 - p'_{u,v})^{1-b_{u,v}}$$

$$\text{where} \qquad p'_{u,v} = \begin{cases} p_{u,v} & \text{if} \quad \neg\big(u \in \varepsilon(W_1), v \in \varepsilon(W_2)\big), \\ \frac{p_{u,v} \cdot \exp(\lambda_W)}{1 - p_{u,v} + p_{u,v} \cdot \exp(\lambda_W)} & \text{otherwise.} \end{cases}$$

How to compute λ_W is also given in [100].

Remark 6. (Updating P if a summarization pattern is presented) *In the case that a summarization pattern $(\mathbb{S}, \mathcal{S})$ is presented to the user, we simply update the background distribution as if all the subgroup patterns related to $(\mathbb{S}, \mathcal{S})$ were presented, and we denote such updated background distribution by $P_{(\mathbb{S},\mathcal{S})}$.*

4.4.3 The subjective interestingness measure

We now discuss how the SI measure can be formalized by relying on the background distribution, first for local and then for global patterns.

4.4.3.1 The SI measure for a local pattern

The information content *(IC)*. Given a pattern (W_1, W_2, I, k_W), and a background distribution defined by P, the probability of the presence of the pattern is the probability of getting more than k_W (for $I = 0$) or $n_W - k_W$ (for $I = 1$)

[4] simply by replacing $\gamma_{k_{u,v}}$ in Eq. 4.1 with $\sum_{i=1}^{i=h} \gamma_{k^i_{u,v}}$ where h is the number of attributes considered (also the number of partitions).

successes in n_W trials with possibly different success probability $p_{u,v}$ (for $I = 0$) or $1-p_{u,v}$ (for $I = 1$). More specifically, we consider a success for the case $I = 0$ to be the presence of an edge between some pair of vertices (u, v) for $u \in \varepsilon(W_1)$, $v \in \varepsilon(W_2)$, and $p_{u,v}$ is the corresponding success probability. In contrast, the absence of an edge between some vertices (u, v) is deemed to be a success for the case $I = 1$, with the probability as $1 - p_{u,v}$. The work of [100] proposed to tightly upper bound the probability of a similar dense subgraph pattern by applying the general Chernoff/Hoeffding bound [103, 104]. Here, we can use the same approach, which gives:

$$\mathbf{Pr}[(W_1, W_2, I = 0, k_W)] \leq \exp\Big(- n_W \mathbf{KL}\Big(\frac{k_W}{n_W} \parallel p_W\Big)\Big),$$

$$\mathbf{Pr}[(W_1, W_2, I = 1, k_W)] \leq \exp\Big(- n_W \mathbf{KL}\Big(1 - \frac{k_W}{n_W} \parallel 1 - p_W\Big)\Big),$$

where

$$p_W = \frac{1}{n_W} \sum_{u \in \varepsilon(W_1), v \in \varepsilon(W_2)} p_{u,v}. \tag{4.2}$$

$\mathbf{KL}\Big(\frac{k_W}{n_W} \parallel p_W\Big)$ is the Kullback-Leibler divergence between two Bernoulli distributions with success probabilities $\frac{k_W}{n_W}$ and p_W respectively. Note that:

$$\begin{aligned}\mathbf{KL}\big(\frac{k_W}{n_W} \parallel p_W\big) &= \mathbf{KL}\big(1 - \frac{k_W}{n_W} \parallel 1 - p_W\big), \\ &= \frac{k_W}{n_W} \log\big(\frac{k_W/n_W}{p_W}\big) + \big(1 - \frac{k_W}{n_W}\big) \log\big(\frac{1 - k_W/n_W}{1 - p_W}\big).\end{aligned}$$

We can thus write, regardless of I:

$$\mathbf{Pr}[(W_1, W_2, I, k_W)] \leq \exp\Big(- n_W \mathbf{KL}\Big(\frac{k_W}{n_W} \parallel p_W\Big)\Big).$$

The information content is the negative log probability of the pattern being present under the background distribution. Thus, using the above:

$$\begin{aligned}\mathrm{IC}[(W_1, W_2, I, k_W)] &= -\log(\mathbf{Pr}[(W_1, W_2, I, k_W)]), \\ &\geq n_W \mathbf{KL}\Big(\frac{k_W}{n_W} \parallel p_W\Big).\end{aligned} \tag{4.3}$$

The description length *(DL)*. A pattern with larger IC is more informative. Yet, sometimes it is harder for the user to assimilate as its description is more complex. A good SI measure should trade off IC with DL. The DL should capture the length of the description needed to communicate a pattern. Intuitively, the cost for the user to assimilate a description W depends on the number of selectors in W, i.e.,

$|W|$. Let us assume communicating each selector in a description W has a constant cost of α, and the cost for I and k_W is fixed as β. The total description length of a pattern (W_1, W_2, I, k_W) can then be written as

$$\mathrm{DL}[(W_1, W_2, I, k_W)] = \alpha(|W_1| + |W_2|) + \beta. \tag{4.4}$$

The subjective interestingness *(SI)*. In summary, we obtain:

$$\begin{aligned} \mathrm{SI}[(\mathbb{S}, \mathcal{S})] &= \frac{\mathrm{IC}[(W_1, W_2, I, k_W)]}{\mathrm{DL}[(W_1, W_2, I, k_W)]}, \\ &= \frac{n_W \mathbf{KL}\left(\frac{k_W}{n_W} \parallel p_W\right)}{\alpha(|W_1| + |W_2|) + \beta}. \end{aligned} \tag{4.5}$$

Remark 7. (Justification about choices of α and β) *In all our experiments for use cases, we apply $\alpha = 0.6, \beta = 1$. We here state the reason for this choice.*

In practice, the absolute value of the SI from Eqs. 4.5 is largely irrelevant, as it is only used for ranking the patterns, or even just for finding a single pattern (i.e., the most interesting one to the user). Thus, we can set $\beta = 1$ without losing generality, such that the only remaining parameters is α.

Tuning α biases the results toward more or fewer selectors to describe the subgroup pattern. Notice an optimal extent of such kind of bias cannot be determined by doing model selection in the statistical sense, but rather should be chosen based on aspects of human cognition (e.g., larger α should be used when the user prefers patterns in a more succinct form). In this work, we set $\alpha = 0.6$ throughout all use cases which gives qualitative results. However, α can be flexibly tuned for adapting to the user' preferences.

4.4.3.2 The SI measure for a global pattern

The information content *(IC)*. The probability of a global summarization pattern turns out to be harder to formulate analytically, and thus also the negative log probability of the pattern – which is the subjective amount of information gained by observing the pattern. However, it is relatively straightforward to quantify the (subjective) amount of information in the connectivity in the graph prior to observing the pattern, and after observing the pattern. The difference between these two is thus the information gained. More formally, we thus mathematically define the IC of a summarization pattern $(\mathbb{S}, \mathcal{S})$ as the difference between the log probability for the connectivity in the graph (i.e., the edge set E) under $P_{(\mathbb{S},\mathcal{S})}$ and that under P:

$$\mathrm{IC}[(\mathbb{S}, \mathcal{S})] = \log P_{(\mathbb{S},\mathcal{S})}(E) - \log P(E). \tag{4.6}$$

This quantity is straightforward to compute where $P_{(\mathbb{S},\mathcal{S})}$ is computed as the updated background distribution as if all the subgroup patterns related to $(\mathbb{S}, \mathcal{S})$ were presented (previously mentioned in Remark 6 in Sec. 4.4.2.2).

The description length *(DL)*. We search for optimal $\mathbb{S}$ by a strategy that is based on splitting a binary search tree (for details see Sec. 4.5.2.1). Thus, the cost for the user to assimilate $\mathbb{S}$ is linear to the number of descriptions in $\mathbb{S}$, i.e. c. As for $\mathcal{S}$, assimilating it costs quadratically to c, because $\mathcal{S}$ is essentially a complete graph with c vertices and $c(c+1)/2$ edges. The total description length of a pattern $(\mathbb{S}, \mathcal{S})$ can be written as

$$\mathrm{DL}[(\mathbb{S}, \mathcal{S})] = \zeta \cdot c(c+1)/2 + \eta \cdot c + \theta. \tag{4.7}$$

where θ is a constant term for mitigating the quadratically increasing drop in SI value given by an increasing c, and this helps to avoid early stopping.
The subjective interestingness *(SI)*. In summary, we obtain:

$$\begin{aligned} \mathrm{SI}[(\mathbb{S}, \mathcal{S})] &= \frac{\mathrm{IC}[(\mathbb{S}, \mathcal{S})]}{\mathrm{DL}[(\mathbb{S}, \mathcal{S})]}, \\ &= \frac{\log P_{(\mathbb{S},\mathcal{S})}(E) - \log P(E)}{\zeta \cdot c(c+1)/2 + \eta \cdot c + \theta.}. \end{aligned} \tag{4.8}$$

Remark 8. (Justification about choices of ζ, η and θ) *In all our experiments, we use $\zeta = 0.02, \eta = 0.02, \theta = 1$. As stated in Remark 7 in Sec. 4.4.3.1, parameters of the DL indicate how much the user prefers patterns that can be described succinctly, and thus should be determined based on aspects of human cognition instead of statistical model selection. We here follow the similar sense to choose the DL parameters for global patterns (i.e.,ζ, η and θ in Eq. 4.8). Notice we set a high value for θ (i.e., 1) in comparison with ζ (i.e., 0.02) and η (i.e., 0.02). This is a safe choice to avoid early stopping (i.e., the iterating stops before the user observes a suitable global pattern).*

4.5 Algorithms

This section describes the algorithms for mining interesting patterns locally and globally, in Sec. 4.5.1 and Sec. 4.5.2 respectively, followed by an outline to the implementation in Sec. 4.5.3.

4.5.1 Local pattern mining

Since the proposed SI interestingness measure is more complex than most objective measures, we consider applying some heuristic search strategies to help maintain the tractability. For searching single-subgroup patterns, we used beam search (see Sec. 4.5.1.1). To search for the bi-subgroup patterns, however, a traditional beam over both W_1 and W_2 simultaneously turned out to be more difficult to apply effectively. We thus propose a nested beam search strategy to handle this case. More details about this strategy are covered by Sec. 4.5.1.2.

4.5.1.1 Beam search

In the case of mining single-subgroup patterns, we applied a classical heuristic search strategy over the space of descriptions—the beam search. The general idea is to only store a certain number (called the *beam width*) of best partial description candidates of a certain length (number of selectors) according to the SI measure, and to expand those next with a new selector. This is then iterated. This approach is standard practice in subgroup discovery, being the search algorithm implemented in popular packages such as Cortana [105], One Click Miner [106], and pysubgroup [107].

4.5.1.2 Nested beam search

Table 4.1: Notations for Algorithm 4.1

Notation	Description
OuterBeam	The outer beam storing best description pairs (W_1, W_2) during the search.
InnerBeam	The inner beam only storing best descriptions W_2.
x_1	The outer beam width (i.e., the minimum number of different descriptions W_1 contained in the outer beam.
x_2	The inner beam width.
D	The search depth (i.e., maximum number of selectors combined in a description).

The basic idea of this approach is to nest one beam search into the other one where the outer search branches based on a 'beam' of promising selector candidates for the description W_1 , and the inner search expands those for W_2. The detailed procedure for this nested beam search is shown in Algorithm 4.1, and related notation displayed in Table 4.1.

The total number of interesting patterns identified by Algorithm 4.1 is $x_1 \cdot x_2$. Note that we deliberately constrain the beam to contain at least x_1 different W_1 descriptions so that a sufficient diversity among all the discovered patterns is guaranteed (see lines 22-23 in Algorithm 4.1).

4.5.2 Global pattern mining

To identify the most interesting global (or summarization) pattern, a greedy search strategy (see Sec. 4.5.2.1) equipped with some speedup strategies (see Sec. 4.5.2.2) are adopted.

Algorithm 4.1: Subjectively Interesting BiSubgroup Pattern Mining

input : Graph $G = \{V, E, A\}$, x_1, x_2, D
output: Top $x_1 \cdot x_2$ bi-subgroup patterns contained in OuterBeam

$S \leftarrow$ the set of all selectors to build descriptions from;
OuterBeam $\leftarrow \{\emptyset\}$;
$d_1 \leftarrow 0$;
$d_2 \leftarrow 0$;
while $d_1 < D$ **do** `// The outer search`
  $\mathbb{C}_1 \leftarrow$ all the W_1 candidates in OuterBeam;
  for $C_1 \in \mathbb{C}_1$ **do** `// Expand on` W_1 `candidates`
    for $s_1 \in S$ **do**
      $Z_1 \leftarrow C_1 \wedge s_1$;
      InnerBeam $\leftarrow \{\emptyset\}$;
      while $d_2 < D$ **do** `// The inner search`
        $\mathbb{C}_2 \leftarrow$ all the W_2 candidates in InnerBeam;
        for $C_2 \in \mathbb{C}_2$ **do** `// Expand` W_2 `candidates`
          for $s_2 \in S$ **do**
            $Z_2 \leftarrow C_2 \wedge s_2$;
            $k_W \leftarrow$ the number of edges between vertices $\varepsilon(Z_1)$ and $\varepsilon(Z_2)$;
            `// compute SI of the pattern` (Z_1, Z_2, I, k_W) `using Eq. 4.5`
            si$' \leftarrow$ SI$[(Z_1, Z_2, I, k_W)]$;
            `// Add (si`$', Z_2$`) to the InnerBeam if InnerBeam contains less than` x_2 `elements or replace the tuple with the smallest SI in InnerBeam if si`$'$ `is larger than that value`
            InnerBeam $\leftarrow$ `UpdateBeam` (InnerBeam, (si$', Z_2$), x_2) ;
        $d_2 \leftarrow d_2 + 1$
      for (si$, Z$) $\in$ InnerBeam **do**
        `// Add (si`$, Z_1, Z$`) to the OuterBeam if the number of various` W_1 `descriptions in OuterBeam is less than` x_1 `or replace the tuple with the smallest SI if si is larger than that value`
        OuterBeam $\leftarrow$ `UpdateBeam` (OuterBeam, (si, Z_1, Z), x_1) ;
  $d_1 \leftarrow d_1 + 1$

4.5.2.1 The basic search strategy

The algorithm begins by checking each possible summarization rule only containing a single-selector description and its negation. Applying such a rule at the beginning means cutting the whole vertex set into two non-overlapping clusters, each of which satisfies a description in this rule correspondingly. The rule whose corresponding summarizaiton pattern has the maximal SI value is selected as a seed set for $\mathbb{S}$. Then the algorithm iterates in the following way to greedily grow that set: for each existing description in the set, the algorithm again checks the application of an additional single-selector description and its negation. This further separates a particular vertex cluster into two sub-clusters, one of which additionally satisfies this description and the other does not. The optimal combination of the existing description to further specify and the additional single-selector description are selected. The search stops when reaching some search budget (e.g. the maximum number of iterations). The detailed procedure for this search is displayed in Algorithm 4.2.

4.5.2.2 Speedup strategies

Parallel Processing. Our search strategy is trivially parallelizable. To gain some speedup, the search process for each attribute and its related selectors (lines 10-24 in Algorithm 4.2) is executed simultaneously in multiple processors.

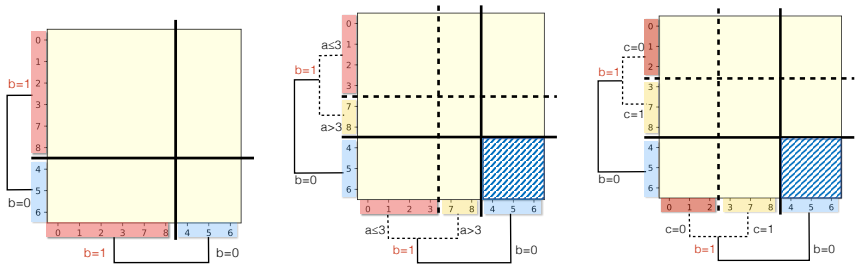
Reusing some computations. We further speedup the search by circumventing some redundant computations when computing the SI for each candidate of summarization pattern. As mentioned above in Sec. 4.4.2.2, $P_{(\mathbb{S},\mathcal{S})}$ is computed as an updated background distribution as if all the subgroup patterns related to $(\mathbb{S}, \mathcal{S})$ were presented, which requires to determine λ_W for each related subgroup pattern. Nevertheless, when branching in different ways during the search (i.e., using different pairs of a selector and its negation to extend a given description), extensions do not interfere with subgroup patterns whose descriptions are not extended. Hence, their λ_W do not need to be recomputed, providing a speed up.

Here we illustrate that, by taking the attributed network in Fig. 4.1 as the example (see Fig. 4.3 which visualizes the corresponding adjacency matrix with arranged vertex indices in left and in bottom; Entries are not indicated for simplicity). Assume the network is currently divided into two vertex subgroups each respectively satisfying $b = 1$ and $b = 0$, and the search is in the step of finding the optimal selector to specify the description $b = 1$ (indices of corresponding vertices are highlighed in red in Fig. 4.3 (a)). Though the adjacency matrix is cut in two different ways, refining the description $b = 1$ into two more specific ones by adding $a \leq 3$ and $a > 3$ (in Fig. 4.3 (b)), or adding $c = 0$ and $c = 1$ (in Fig. 4.3 (c)), both do not interfere with the subgroup satisfying $b = 0$ (the blue striped area).

Algorithm 4.2: Interesting Summarization Pattern Mining

input : Graph $G = \{V, E, A\}$, Search Iteration budget D
output: $(\mathbb{S}, \mathcal{S})$

$\mathbb{S} \leftarrow \{\emptyset\}$;
VertexClusters $\leftarrow \{V\}$// A set of vertex-clusters each of which is formed by the extension of a description in $\mathbb{S}$. Initially, it is a set only containing one member, the whole vertex set;
$i \leftarrow 0$ // The number tracking the iteration round;
while $i < D$ **do**
    si$\leftarrow -\infty$;
    $\mathbb{S}' \leftarrow \mathbb{S}$;
    $\mathbb{S}'' \leftarrow \mathbb{S}$;
    VertexClusters$' \leftarrow$ VertexClusters;
    for $W \in \mathbb{S}$ **do** // Iterate over each description rule currently in $\mathbb{S}$
        for $a \in A$ **do** // Iterate over each attribute
            $S_a \leftarrow$ the set of all selectors associated with the attribute a;
            for $s \in S_a$ **do** // Iterate over each selector of the attribute a
                // Update $\mathbb{S}'$ by replacing W with two more specific descriptions such that one additionally satisfies s, and the other does not
                $\mathbb{S}' \leftarrow \mathbb{S}' \setminus \{W\} \cup \{W \wedge s, W \wedge \neg s\}$;
                // Update VertexClusters$'$ correspondingly
                VertexClusters$' \leftarrow$ VertexClusters$' \setminus \{\varepsilon(W)\} \cup \{\varepsilon(W \wedge s), \varepsilon(W \wedge \neg s)\}$;
                $\mathcal{S}' \leftarrow$ A summary of $G = \{V, E, A\}$ based on the summarization rule $\mathbb{S}$;
                si$' \leftarrow$ SI$[(\mathbb{S}', \mathcal{S}')]$;
                if si$'$ >si **then**
                    si$\leftarrow$si$'$;
                    $\mathbb{S} \leftarrow \mathbb{S}'$;
                    VertexClusters$\leftarrow$VertexClusters$'$;
                    $\mathcal{S} \leftarrow \mathcal{S}'$
                $\mathbb{S}' \leftarrow \mathbb{S}''$// Revert to $\mathbb{S}''$;
    $i++$;



(a) The adjacency matrix before branching.

(b) Branching from $b = 1$ in a way.

(c) Branching from $b = 1$ in another way.

Figure 4.3: Illustration of the existence of a common subgroup pattern when branching in two different ways.

4.5.3 Implementation

For mining patterns locally, *Pysubgroup* [107], a Python package for subgroup discovery implementation written by Florian Lemmerich, was used as a base to be built upon. We integrated our nested beam search algorithm and SI measure (along with other state-of-the-art interestingness measures for comparison) into this original interface. A Python implementation of all the algorithms and the experiments is available at https://bitbucket.org/ghentdatascience/globalessd_public. All experiments were conducted on a PC with Ubuntu OS, Intel(R) Core(TM) i7-7700K 4.20GHz CPUs, and 32 GB of RAM.

4.6 Experiments

We evaluate our methods on six real-world networks. In the following, we first describe the datasets (Sec. 4.6.1). Then we present the conducted experiments and discuss the results with a purpose to address the following questions:

RQ1 Are our local pattern mining algorithms sensitive to the beam width? (Sec. 4.6.2)

RQ2 Does our SI measure outperform state-of-the-art objective interestingness measures? (Sec. 4.6.3)

RQ3 Is the SI truly subjective, in the sense of being able to consider a user's prior beliefs? (Sec. 4.6.4)

RQ4 How can optimizing SI help avoid redundancy between iteratively mined patterns? (Sec. 4.6.5)

RQ5 Is our global pattern mining approach able to summarize the whole graph in a meaningful way such that all the interesting patterns can be revealed? (Sec. 4.6.6)

RQ6 How do the algorithms scale? (Sec. 4.6.7)

4.6.1 Data

Basic data information is summarized in Table 4.2.

***Caltech36* and *Reed98*.** Two Facebook social networks from the Facebook100 [108] data set, gathered in September 2005: one for Caltech Facebook users, and one for Reed University. Vertex attributes describe the person's status (faculty or student), gender, major, minor, dorm/house, graduation year, and high school.

***Lastfm*.** A social network of friendships between `Lastfm.com` users, generated from the publicly available dataset [109] in the HetRec 2011 workshop. In this dataset, tag assignments of a list of most-listened musical artists provided by each user are given in [user, tag, artist] tuples, where those tags are unstructured text labels that users used to express songs of artists. We then took tags that a user ever assigned to any artist and assigned those to the user as binary attributes expressing a user's music interests. This dataset has been used in many publications to evaluate local pattern mining methods [11, 14, 15].

***DBLPtopics*.** A citation network generated from the DBLP citation data V11[5] [110, 111] by choosing a random subset of publications from 20 conferences[6] selected to cover 4 research areas: Machine Learning, Database, Information Retrieval, and Data Mining. Vertices represent publications, and directed edges represent citation relationships. Each publication is annotated with 50 attributes (denoted by $a_1, a_2, \ldots, a_{50}$) whose value indicates the relevance of this paper to a certain topic. These attributes are obtained by computing the first 50 *latent semantic indexing (LSI)* components for the original paper-topic matrix (of size 10837×9074) where each entry value indicates the relevance of a paper (represented by row) to a field of study (represented by column) and this value is provided by the original DBLP data. In our work, the selector space on which the search is carried does not include every attribute value pair. A discretization is applied here: values for each attribute are sorted and discretized into 4 partitions of equal size by 3 quartiles. This gives $3 \times 2 = 6$ selectors for each attribute ($6 \times 50 = 300$ selectors in total) three of which respectively assign *true* to vertices with value smaller than the first, second, third quartile of the total values for this

[5]This citation dataset are extracted from DBLP website: https://dblp.uni-trier.de/, containing 4107340 publications (from unknown year till May 2019) and 36624464 citation relationships. It can be accessed by: https://aminer.org/citation

[6]AAAI, CIKM, ECIR, ECML-PKDD, ICDE, ICDM, ICDT, ICLR, ICML, IJCAI, KDD, NIPS, PAKDD, PODS, SDM, SIGIR, SIGMOD, VLDB, WSDM, WWW

Table 4.2: Dataset statistics summary.

Dataset	Type	$\|V\|$	$\|E\|$	Attribute type	#Attributes	$\|S\|$
Caltech36	undirected	762	16651	nominal	7	602
Reed98	undirected	962	18812	nominal	7	748
Lastfm	undirected	1892	12717	binary	11946	21695
DBLPtopics	directed	10837	6883	numerical	50	300
DBLPaffs	directed	6472	3066	binary	116	232
MPvotes	undirected	650	49631	binary	39	78

attribute, and the other three are the corresponding negations. We denote the i-th quartile of values for the attribute a by Q_i^a.

DBLPaffs. A DBLP citation network based on a random subset of publications same as the one for the above task. Only papers for which the authors' country (or state, in the USA) of affiliation is available are included as vertices. The resulting 116 countries/states are included as binary vertex attributes, set to 1 iff one of the paper's authors is affiliated to an institute in that country/state.

MPvotes. The Twitter social network generated from friendships between Members of Parliament (MPs) in UK [112]. Their voting records on Brexit from 12th June 2018 to 3rd April 2019 are included as 39 binary vertex attributes, set to be 1, or -1 iff this MP vote for/abstain or, against/abstain respectively. Note we include abstain on both positive and negative sides rather than make abstain (or not abstain) alone being a value, because a selector that describes a subgroup of MPs abstaining (or not abstaining) in a particular vote is not very meaningful in practice.

4.6.2 Parameter sensitivity (RQ1)

For mining local patterns, we used the standard beam search for single-subgroup patterns, and the nested beam search for bi-subgroup patterns. In all experiments, we set the search depth $D = 2$ (because patterns that are described by more than 2 selectors often appear less interesting in practice, and they would add unnecessary difficulty for interpretation). Then the performance of those beam search methods ultimately depends on the beam width.

4.6.2.1 Experimental setup

Choice of datasets. We used *Lastfm* to investigate the effect of the beam width on the performance of single-subgroup pattern mining, as it involves the largest search space (given by the largest number of selectors i.e., 21695). With regard to that on bi-subgroup pattern mining, because the search is more time-consuming, we used *Lastfm* while only considering 100 most frequently used tags as attributes

(i.e., giving 200 selectors as the search space). We also used *Reed98* as it involves the largest search space among datasets that were used in our experiments on bi-subgroup pattern mining.

Other settings. Though we applied the SI measure with $\alpha = 0.6$, $\beta = 1$ in all use cases of local pattern mining (as previously mentioned in Remark 7 in Sec. 4.4.3.1), to more meaningfully investigate the parameter sensitivity in this experiment, we set α to be smaller, i.e., $\alpha = 0.1$.[7]

4.6.2.2 Results

Effect of the beam width on single-subgroup pattern mining. First, we analyze the sensitivity of the standard beam search w.r.t. the beam width for single-subgroup pattern mining. How the search performance changes with the beam width (denoted by x) is illustrated below (see Fig. 4.4 (a) for the SI value of the identified best pattern and Fig. 4.4 (b) for the run time).

Clearly, increasing x from 1 to 40 results in the same best pattern (with the SI value as 258.7, the description as 'IDM $= 1$') along with a gentle increase in the run time. Though it shows a greedy search (i.e., $x = 1$) can already perform well, this is not guaranteed.

As indicated in a further investigation, increasing the beam width is rendered useless by the existence of a dominant pattern with a single selector (i.e., 'IDM $= 1$') such that there are no other patterns that have higher SI value than it and its children. Once our method incorporates this dominant pattern into the background distribution for one subsequent iteration to reflect the user's newly acquired knowledge, the advantage of a larger beam width appears as the best pattern is identified when x increases to be 3 (see Fig. 4.5 (a)). The run time grows linearly as x increases (see Fig. 4.5 (b)).

Effect of the beam width on bi-subgroup pattern mining. To study the effects of the beam width, we implemented all cases with x_1 and x_2 being $1, 2, 3, 4$, or 7.

In *Lastfm*, clearly from Fig. 4.6(a), small beam widths (e.g., when $x_1 = 1$ with $x_2 = 3$) are sufficient for our algorithm to identify the best bi-subgroup pattern (i.e., the one with SI as 194.8). This is even more the case for *Reed98* network, as our method of bi-subgroup pattern mining always identify the same best bi-subgroup pattern (i.e., the one with SI as 728) when gradually increasing x_1 and x_2.

[7] In this sensitivity investigation, applying a relatively larger α (e.g., $\alpha = 0.6$) can more possibly lead to positive results (i.e., showing our algorithms are insensitive to the beam width, as the same best pattern is always identified while varying the beam width) but by a fluke: setting α *larger* in the SI measure penalizes more complex patterns *more heavily*, and this makes the best pattern found before further branching in a beam search more easily dominate, giving less credible positive results. We thus safely chose α to be 0.1 in this experiment.

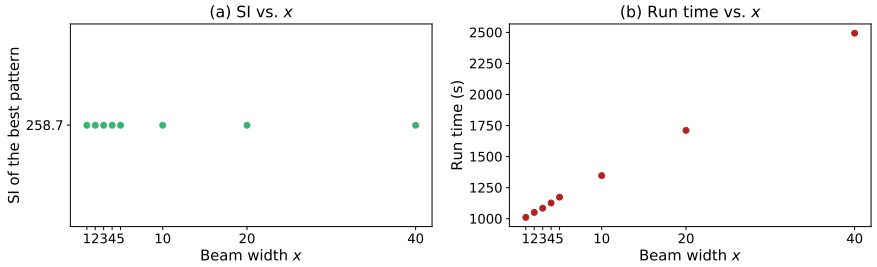


Figure 4.4: Varying the beam width x in the search for single-subgroup patterns in Lastfm.

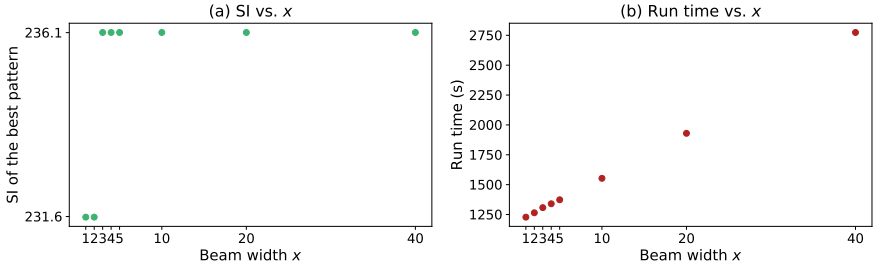


Figure 4.5: Varying the beam width x in the search for single-subgroup patterns in Lastfm after incorporating the dominant pattern described by 'IDM $= 1$'.

For bi-subgroup pattern mining in either *Lastfm* or *Reed98*, the run time experiences an approximately linear growth as x_1 or x_2 increases with the other beam width is fixed (see Fig. 4.6(b) and Fig. 4.6(c) for *Lastfm*, Fig. 4.7(b) and Fig. 4.7(c) for *Reed98*).

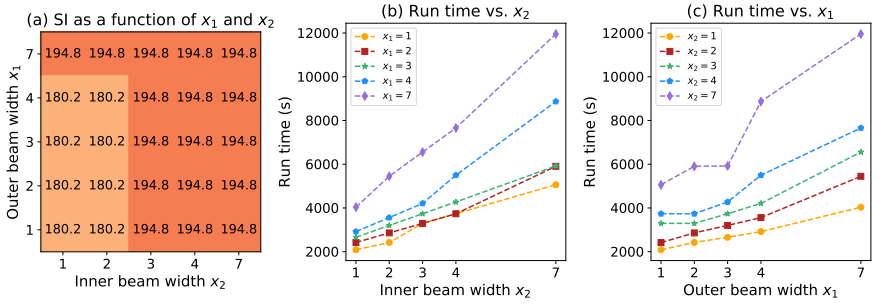


Figure 4.6: Varying the outer/inner beam width x_1/x_2 in the search for bi-subgroup patterns in Lastfm.

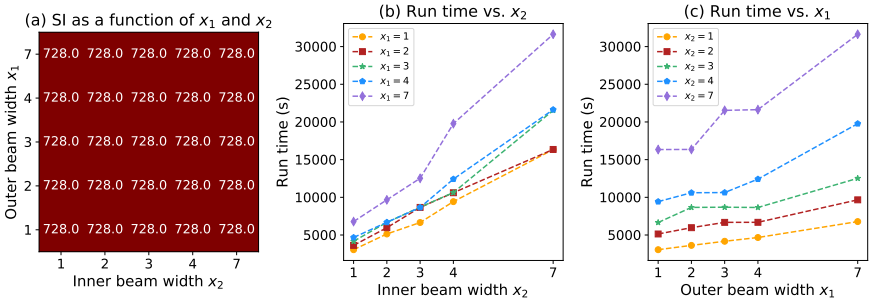


Figure 4.7: Varying the outer/inner beam width x_1/x_2 in the search for bi-subgroup patterns in Reed98.

Summary. This empirical analysis suggests that overall our algorithms are not sensitive to the beam width. A small beam width is usually sufficient, particularly if there is a dominant pattern. When that is not the case, slightly increasing the beam width was sufficient in our experiments.

We recommend an initial setting with $x = 5$ for single-subgroup pattern discovery and $x_1 = 2, x_2 = 3$ for bi-subgroup pattern discovery, which is usually more than sufficient. If it is not sufficient, the user can increment x, either x_1 or x_2 by 1 iteratively until satisfying results are yielded.

4.6.3 Comparative evaluation (RQ2)

4.6.3.1 Experimental setup

A comparison between the SI and other objective interestingness measures can only be made on their performances on single-subgroup pattern discovery (or more precisely, dense subgraph mining), because those existing objective measures are limited to quantify the interestingness of a dense subgraph community.

Choice of datasets and prior beliefs. To constrain the search that uses our SI measure to only identify dense subgraphs, we applied individual vertex degrees as the prior beliefs, and chose sparse networks (i.e, *Lastfm* and *DBLPaffs*) for this comparative task. When using the individual vertex degree as priors, single-subgroup patterns' density will not be explainable merely from the individual degrees of the constituent vertices. For real-world networks, given its sparsity (which is common), incorporating this prior leads to a background distribution with a low average connection probability. In this case, our algorithm identify mostly dense clusters (i.e. $I = 0$), as these are more informative in the sense of strongly contrasting with the expectation which is towards sparsity. *Lastfm*, *DBLPtopics* and *DBLPaffs* are all evidently sparse networks. Among them, *Lastfm* and *DBLPaffs* were chosen as their attributes and the discovered patterns are more readily understood.

Baselines. For this comparative evaluation, we consider the following baselines:

- *Edge density.* The number of edges divided by the maximal number of edges.
- *Average degree.* The degree sum for all vertices divided by the number of vertices.
- *Pool's community score [14].* The reduction in the number of erroneous links between treating each vertex as a single community and treating all vertices as a whole.
- *Edge surplus [35].* The number of edges exceeding the expected number of edges assuming each edge is present at the same probability α.
- *Segregation index [113].* The difference between the number of expected inter-edges to the number of observed inter-edges, normalized by the expectation.
- *Modularity of a single community [114, 115].* The modularity measure of a single community based on transforming the definition of modularity to a local measure.
- *Inverse average-ODF (out-degree fraction) [116].* 1 minus the average fraction of vertices' out-degrees to degrees.
- *Inverse conductance.* The number of edges inside the cluster divided by the number of edges leaving the cluster.

More detailed descriptions along with mathematical definitions for these baselines can be found in Table 4.11 in the Appendix 4.A.

Other settings. For single-subgroup pattern discovery on both *Lastfm* and *DBLPaffs* networks, we use beam search with beam width 5 and search depth 2.

4.6.3.2 Results

Four most interesting patterns w.r.t. the SI and these baseline measures on *Lastfm* are presented in Table 4.3 and Table 4.4 respectively. For each pattern, we display values for elements that constitute the pattern syntax including W, I, k_W, and also other statistics including its rank, $|\varepsilon(W)|$, and #inter-edges. #inter-edges is the number of connections between $\varepsilon(W)$ and $V \setminus \varepsilon(W)$, telling how isolated a particular group of members is. Particularly for patterns discovered using the SI, we also display $p_W \cdot n_W$, the expected number of connections within $\varepsilon(W)$ w.r.t. the background distribution. Comparing $p_W \cdot n_W$ to k_W gives a direct sense of how much the user's expectation differs from the truth (Recall p_W from Eqs. 4.2).

Table 4.3: Top 4 single-subgroup patterns w.r.t. the SI in Lastfm network. For each pattern (each row), we display values for elements that constitute the pattern syntax including W, I, k_W, and also other statistics including its rank, $|\varepsilon(W)|$, $p_w \cdot n_W$ and #inter-edges (each column). k_W is the number of observed edges within $\varepsilon(W)$ (i.e., the set of vertices satisfying the description W), and $p_W \cdot n_W$ is the expected number of edges within $\varepsilon(W)$ w.r.t. the background distribution. I is the indicator equal to 0 if the observed pattern is dense for the user (i.e., $k_W > p_W \cdot n_W$) or 1 otherwise (i.e., $k_W < p_W \cdot n_W$). #inter-edges is the number of connections between $\varepsilon(W)$ and $V \setminus \varepsilon(W)$.

Rank	W	I	k_W	$\lvert\varepsilon(W)\rvert$	$p_W \cdot n_W$	#inter-edges
1	idm $= 1$	0	96	78	8.93	496
2	heavy metal $= 1$	0	220	165	60.04	1322
3	synthpop $= 1$	0	208	131	57.32	1307
4	new wave $= 1$	0	292	191	104.01	1731

Here, we summarize the main findings.

Using baselines. Each of those objective measures exhibits a particular bias that arguably makes the obtained patterns less useful in practice. The edge density is easily maximized to a value of 1 simply by considering very small subgraphs. That's why the patterns identified by using this measure are all those composed of only 2 vertices with 1 connecting edge. In contrast, using the average degree tends to find very large communities, because in a large community there are many other vertices for each vertex to be possibly connected to. Although Pool argued that their measure may be larger for larger communities than for smaller ones, in their own experiments on the *Lastfm* network as well as in our own results, it yields relatively small communities [14]. As they explained, the reason was *Lastfm*'s attribute data is extremely sparse with a density of merely 0.15%. Note that patterns with the top 10 edge surplus values are the same as those for the Pool's measure. Although these two measures are defined in different ways, Pool's measure can be further simplified to a form essentially the same as the edge surplus. Pursuing a larger segregation index essentially targets communities which have much less cross-community links than expected. This measure emphasizes more strongly the number of cross-community links, and yields extremely small or large communities with few inter-edges on *Lastfm*. Using the modularity of a single community tends to find rather large communities representing audiences of mainstream music. The results for the inverse average-ODF and the inverse conductance are not displayed in the supplement, because the largest values for these two measures can be easily achieved by a community with no edges leaving this community, for which a trivial example is the whole network.

Using the SI. We argue that the patterns extracted using our SI measure are most insightful, striking the right balance between coverage (sufficiently large) and specificity (not conveying too generic or trivial information). The top one

Table 4.4: Top 4 single-subgroup patterns w.r.t. baselines in Lastfm network. For each pattern, we display values for elements that constitute the pattern syntax including W, I, k_W, and also other statistics including $|\varepsilon(W)|$, and #inter-edges. k_W is the number of observed connections within $\varepsilon(W)$ (i.e., the set of vertices satisfying the description W). As all other measures are only for quantifying the interestingness of dense subgraphs, the indicator I is always equal to 0. #inter-edges is the number of connections between $\varepsilon(W)$ and $V \setminus \varepsilon(W)$.

Measure	W	I	k_W	$\lvert\varepsilon(W)\rvert$	#inter-edges
Edge Density	1981 songs $= 1$	0	1	2	21
	africa $= 1$	0	1	2	76
	40s $= 1$	0	1	2	22
	early reggae $= 1$	0	1	2	10
Average Degree	post rock $= 0 \wedge$ post-rock $= 0$	0	12181	1783	498
	post-rock $= 0 \wedge$ dark ambient $= 0$	0	12092	1770	573
	post-rock $= 0 \wedge$ grindcore $= 0$	0	12032	1762	634
	post-rock $= 0 \wedge$ technical death metal $= 0$	0	12106	1773	560
Pool's community score or Edge surplus	bionic $= 1 \wedge$ 30 seconds to mars $= 0$	0	8	6	343
	bionic $= 1 \wedge$ taylor swift $= 0$	0	8	6	343
	bionic $= 1 \wedge$ latin $= 0$	0	8	6	343
	bionic $= 1 \wedge$ spanish $= 0$	0	8	6	343
Segregation Index	gluhie 90e$= 0 \wedge$ lithuanian black metal $= 1$	0	3	3	1
	goddesses$= 0 \wedge$ pagan black metal $= 1$	0	3	3	1
	gluhie 90e$= 0 \wedge$ pagan black metal$= 1$	0	3	3	1
	heartbroke$= 0 \wedge$ lithuanian black metal $= 1$	0	3	3	1
Modularity of a single community	pop $= 1 \wedge$ new wave $= 0$	0	2689	475	4913
	pop $= 1 \wedge$ progressive rock $= 0$	0	2943	514	5083
	pop $= 1 \wedge$ experimental $= 0$	0	2844	497	5083
	pop $= 1 \wedge$ metal $= 0$	0	2761	496	5067

characterises a group of 78 IDM (i.e., intelligent dance music) fans. Audiences in this group are connected more frequently than expected (96 vs. 8.93), and they altogether only have 496 connections to those people not into IDM, which is much sparser than connections within the IDM group (as the connectivity density across the group and that within the group are respectively $496/(78 \times 1814) \approx 0.0035$ and $96/(78 \times (78-1)/2) \approx 0.0320$).

Remark 9. (Results on *DBLPaffs*) *For DBLPaffs, the same conclusion as above can also be reached. See top 4 single-subgroup patterns on DBLPaffs w.r.t. our SI and other measures in Table 4.12 and Table 4.13 respectively in the Appendix 4.A.*

Summary. Unlike state-of-the-art objective interestingness measures, each of which exhibits a particular bias, the proposed SI measure achieves a natural balance between coverage and specificity, arguably leading to more insightful patterns.

4.6.4 The effects of different prior beliefs: a subjective evaluation (RQ3)

4.6.4.1 Experimental setup

To demonstrate the SI's subjectiveness, we consider different prior beliefs, in search for patterns w.r.t. the SI. We deliberately perform this evaluation on bi-subgroup pattern discovery for a more generic and interesting setting.

Choice of datasets. In the following, we analyze results on *Caltech36* and *Reed98*. These two networks are chosen, because their straightforward domain knowledge provides us the ease for prior belief settings. People, even those that are not social scientists, normally hold prior beliefs about this sort of friendship network (e.g., they commonly believe that students of different class years are less likely to know each other than students from the same class year).

Other settings. For bi-subgroup pattern discovery, we applied the nested beam search with $x_1 = 2, x_2 = 3$, and $D = 2$. Moreover, we constrain the target descriptions W_1 and W_2 to include at least one common attribute but with various values, so that the corresponding pair of subgroups $\varepsilon(W_1)$ and $\varepsilon(W_2)$ do not overlap with each other. Under this setting, the obtained patterns are more explainable, and the results are easier to evaluate.

4.6.4.2 Results

The 4 most subjectively interesting patterns under each prior belief are presented in Table 4.6 (for *Caltech36*) and Table 4.7 (for *Reed98*), with their associated notations are summarized in Table 4.5.

Incorporating Prior 1. We first incorporated prior belief on the individual vertex degree (i.e. Prior 1). In general, the identified patterns belong to knowledge

Table 4.5: Notations in Table 4.6, 4.7 and 4.8

Notation	Description
W_1/W_2	The description of the first/second subgroup
$\|\varepsilon(W_1)\|/\|\varepsilon(W_2)\|$	The subgroup of vertices satisfying the description W_1/W_2
k_W	The number of observed edges between $\varepsilon(W_1)$ and $\varepsilon(W_2)$
$p_W \cdot n_W$	The expected number of edges between $\varepsilon(W_1)$ and $\varepsilon(W_2)$ w.r.t. the background distribution
I	The indicator equal to 0 if the observed pattern is dense for the user (i.e., $k_W > p_W \cdot n_W$) or 1 otherwise (i.e., $k_W < p_W \cdot n_W$)

commonly held by people, and are not useful. The top 4 patterns on *Caltech36* all reveal people graduating in different years rarely know each other (rows for Prior 1 in Table 4.6), in particular between ones in class of 2006 and ones in class of 2008 (indicated by the most interesting pattern). Although W_2 of the second pattern (i.e., *status* = *alumni*) does not contain the attribute graduation year, it implicitly represents people who had graduated in former year. For *Reed98*, the discovered patterns under Prior 1 also express the negative influence of different graduation years on connections (rows for Prior 1 in Table 4.7).

Incorporating Prior 1 and Prior 2. We then incorporated prior beliefs on the densities between bins for different graduation years (i.e., Prior 2). All the extracted top 4 patterns on *Caltech 36* indicate rare connections between people living in different dormitories, and this is also not surprising (rows for Prior 1 + Prior 2 in Table 4.6).

For *Reed98*, incorporating Prior 1 and Prior 2 provides interesting patterns (rows for Prior 1 + Prior 2 in Table 4.7). The top one indicates people living in dormitory 88 are friends with many in dormitory 89. In contrast, what people commonly believe is that people living in different dormitories are less likely to know each other. For a user who has such preconceived notion, this pattern is interesting. Both the fourth and the seventh patterns reveal a certain person knew more people in class of 2009 than expected.

Incorporating Prior 1, Prior 2 and Prior 3. For *Caltech 36*, by additionally incorporating prior beliefs on the dependency of the connectivity probability on the difference in dormitories (i.e., Prior 3), patterns characterizing some interesting dense connections are discovered (rows for Prior 1 + Prior 2 + Prior 3 in Table 4.7). For instance, the top pattern indicates three people in class of 2004 connect with many in class of 2008. In fact, these three people's graduation had been postponed, as their status is 'student' rather than 'alumni' in year 2005. Furthermore, the starting year for those 2008 cohort is exactly when these three people should have graduated. Therefore, these two groups had opportunities to become friends. The

Table 4.6: Varying prior beliefs in Caltech36 network. Prior 1 represents the prior belief on the individual vertex degree. Both Prior 2 and Prior 3 regard particular attribute knowledge. More specifically, Prior 2 expresses the user's knowledge on the edge densities between bins for different graduation years, and Prior 3 expresses that for different dormitories. For each pattern (each row), we display values for elements that constitute the pattern syntax including W_1, W_2, I, k_W, and also other statistics including its rank, $|\varepsilon(W_1)|$, $|\varepsilon(W_1)|$, and $p_w \cdot n_W$ (each column).

	Rank	W_1	W_2	I	k_W	$\lvert\varepsilon(W_1)\rvert$	$\lvert\varepsilon(W_2)\rvert$	$p_W \cdot n_W$
Prior 1	1	year $=$ 2006	year $=$ 2008	1	1346	153	173	2379.10
	2	status $=$ student $\wedge$ year $=$ 2008	status $=$ alumni	1	842	167	159	1783.26
	3	status $=$ student $\wedge$ year $=$ 2008	year $=$ 2006	1	1330	167	153	2367.96
	4	status $=$ student $\wedge$ year $=$ 2006	year $=$ 2008	1	1346	152	173	2377.53
Prior 1 + Prior 2	1	dorm/house $=$ 169	dorm/house $=$ 171	1	194	99	67	569.56
	2	dorm/house $=$ 169	dorm/house $=$ 166	1	237	99	70	620.42
	3	dorm/house $=$ 169	dorm/house $=$ 172	1	319	99	91	706.65
	4	dorm/house $=$ 169	dorm/house $=$ 170	1	300	99	87	646.04
Prior 1 + Prior 2 + Prior 3	1	status $=$ student $\wedge$ year $=$ 2004	year $=$ 2008	0	108	3	173	25.23
	2	status $=$ student $\wedge$ year $=$ 2004	year $=$ 2008 $\wedge$ minor $=$ 0	0	71	3	114	15.67
	3	status $=$ student $\wedge$ year $=$ 2004	year $=$ 2008 $\wedge$ gender $=$ male	0	71	3	116	16.97
	4	student status $=$ student $\wedge$ dorm/house $=$ 166	student status $=$ alumni $\wedge$ high school $=$ 19445	0	51	53	1	17.52

Table 4.7: Varying prior beliefs in Reed98 network. Prior 1 represents the prior belief on the individual vertex degree. Prior 2 is on the edge densities between bins for different graduation years. For each pattern (each row), we display values for elements that constitute the pattern syntax including W_1, W_2, I, k_W, and also other statistics including its rank, $|\varepsilon(W_1)|$, $|\varepsilon(W_1)|$, and $p_w \cdot n_W$ (each column).

	Rank	W_1	W_2	I	k_W	$\lvert\varepsilon(W_1)\rvert$	$\lvert\varepsilon(W_2)\rvert$	$p_W \cdot n_W$
Prior 1	1	year $=$ 2008	year $=$ 2005	1	495	209	117	1401.97
	2	year $=$ 2007	year $=$ 2009	1	112	165	158	661.41
	3	status $=$ student $\wedge$ year $=$ 2008	year $=$ 2005	1	495	209	117	1401.97
	4	year $=$ 2008	year $=$ 2006	1	765	209	131	1643.38
Prior 1 +Prior 2	1	dorm/house $=$ 89	dorm/house $=$ 88	0	188	23	37	68.80
	2	dorm/house $=$ 89 $\wedge$ status $=$ student	dorm/house $=$ 88	0	188	22	37	68.45
	3	dorm/house $=$ 88 $\wedge$ status $=$ student	dorm/house $=$ 89	0	183	36	23	65.47
	4	dorm/house $=$ 111 $\wedge$ year $=$ 0	year $=$ 2009	0	24	1	158	0.66
	7	dorm/house $=$ 96 $\wedge$ year $=$ 2005	year $=$ 2009	0	12	1	158	0.07

fourth pattern indicates an alumnus who had studied in a high school knew almost all the students living in a certain dormitory. The reason behind this pattern might be worth investigating, which could be for instance, this alumni worked in this dormitory.

Summary. As the results show, incorporating different prior beliefs leads to discovering different patterns that strongly contrast with these beliefs. The proposed SI measure thus succeeds in quantifying the interestingness in a subjective manner.

4.6.5 Evaluation on iterative pattern mining (RQ4)

4.6.5.1 Experimental setup

Our method is naturally suited for iterative pattern mining, in a way to incorporate the newly obtained pattern into the background distribution for subsequent iterations. We show this on searching for bi-subgroup patterns because they are more generic.

Choice of datasets. Dataset *DBLPaffs* and *Lastfm* are used, as the meanings of their attributes are clear and straightforward, giving an ease to explain the discovered patterns.

Other settings. Other settings for this task are the same as for addressing RQ2. The nested beam search with $x_1 = 2, x_2 = 3$, and $D = 2$ was applied. The target descriptions W_1 and W_2 are constrained to include at least one common attribute but with various values, making the corresponding pair of subgroups $\varepsilon(W_1)$ and $\varepsilon(W_2)$ not overlap with each other.

4.6.5.2 Results

Results for *Lastfm* are displayed and discussed in the Appendix 4.B. Here we only analyze the results on *DBLPaffs*. Table 4.8 displays top 3 patterns found in each of the four iterations on *DBLPaffs*.

Iteration 1. Initially, we incorporated prior on the overall graph density. The resulting top pattern indicates papers from institutes in USA seldom cite those from other countries.

Iteration 2. After incorporating the top pattern in iteration 1, a set of dense patterns were identified. All the top 3 patterns reveal a highly-cited subgroup of papers whose authors are affiliated to institutes in California and New Jersey. This agrees with fact that many of the world's largest high-tech corporations and reputable universities are located in these regions. Examples include Silicon valley, Stanford university in CA, NEC Laboratories, AT&T Laboratories in NJ, among others.

Iteration 3. The top 3 patterns in iteration 3 reveal that papers from authors with Chinese affiliations are rarely cited by papers with authors from other coun-

Table 4.8: Top 3 discovered bi-subgroup patterns of each iteration in DBLPaffs network. For each pattern (each row), we display values for elements that constitute the pattern syntax including W_1, W_2, I, k_W, and also other statistics including its rank, $|\varepsilon(W_1)|$, $|\varepsilon(W_1)|$, and $p_w \cdot n_W$ (each column). See Table 4.5 for descriptions of these statistics.

	Rank	W_1	W_2	I	k_W	$\lvert\varepsilon(W_1)\rvert$	$\lvert\varepsilon(W_2)\rvert$	$p_W \cdot n_W$
Iteration 1	1	USA $= 1$	USA $= 0$	1	335	3132	3340	765.83
	2	USA$= 1 \wedge$ China $= 0$	USA $= 0$	1	288	2969	3340	725.97
	3	USA$= 1 \wedge$ Australia $= 0$	USA $= 0$	1	320	3092	3340	756.05
Iteration 2	1	NJ (New Jersey) $= 0$	NJ $= 1 \wedge$ CA (California) $= 1$	0	93	6262	15	6.91
	2	CA $= 0$	NJ$= 1 \wedge$ CA $= 1$	0	86	5584	15	6.13
	3	NJ$= 1 \wedge$ Israel $= 0$	NJ$= 1 \wedge$ CA $= 1$	0	93	6153	15	6.76
Iteration 3	1	China $= 0$	China $= 1$	1	144	5599	873	271.02
	2	China $= 0$	China $= 1 \wedge$ IL (Illinois) $= 0$	1	128	5599	861	266.10
	3	China $= 0 \wedge$ USA $= 0$	China $= 1$	1	64	2630	873	168.09
Iteration 4	1	CA $= 1$	CA $= 0 \wedge$ WA $= 1$	0	55	888	184	11.73
	2	WA $= 0$	WA $= 1$	0	182	6254	218	97.78
	3	CA $= 1 \wedge$ TX (Texas) $= 0$	CA $= 0 \wedge$ WA $= 1$	0	55	876	184	11.57

tries. However, they are frequently cited by papers with Chinese authors, as indicated by our identified top single-subgroup pattern in *DBLPaffs* (see Table 4.12 in the Appendix 4.A). This indicates researchers with Chinese affiliations are surprisingly isolated, the reason of which might be interesting to investigate.

Iteration 4. The top patterns in iteration 4 reveal that papers from institutions in Washington state are highly cited by others, in particular by papers from California. Closer inspection revealed that the majority of these papers are written by authors from Microsoft Corporation and the University of Washington.

Summary. By incorporating the newly obtained patterns into the background distribution for subsequent iterations, our method can identify patterns which strongly contrast with this knowledge. This results in a set of patterns that are not redundant and highly surprising to the user. Note that the lack of redundancy arises naturally, without the need for explicitly constraining the overlap between the patterns in consecutive iterations. In fact, some amount of overlap may still occur, as long as the non-redundant part of the information is sufficiently large.

4.6.6 Empirical results on the discovered global patterns (RQ5)

To demonstrate the use of our method for mining interesting global patterns, we illustrate and analyze the experimental results on *DBLPaffs* (in Sec. 4.6.6.1), *DBLPtopics* (in Sec. 4.6.6.2) and *MP* (in the Appendix 4.C). Each of these datasets serves an interesting case study for us to evaluate our method on.

4.6.6.1 Case study on *DBLPaffs*

Task. Paper citations relate to authors' affiliations to some extent. For example, institutions in some particular countries or regions are reputable, and often produce highly-cited research. Also, collaborations and mutual citations may frequently occur in institutions from some certain countries or regions. Thus, of particular interest could be patterns that describe a subgroup of papers from affiliations A frequently (or rarely) cite papers in another subgroup from affiliations B. We show such patterns can be revealed by a summarization yielded by our approach.

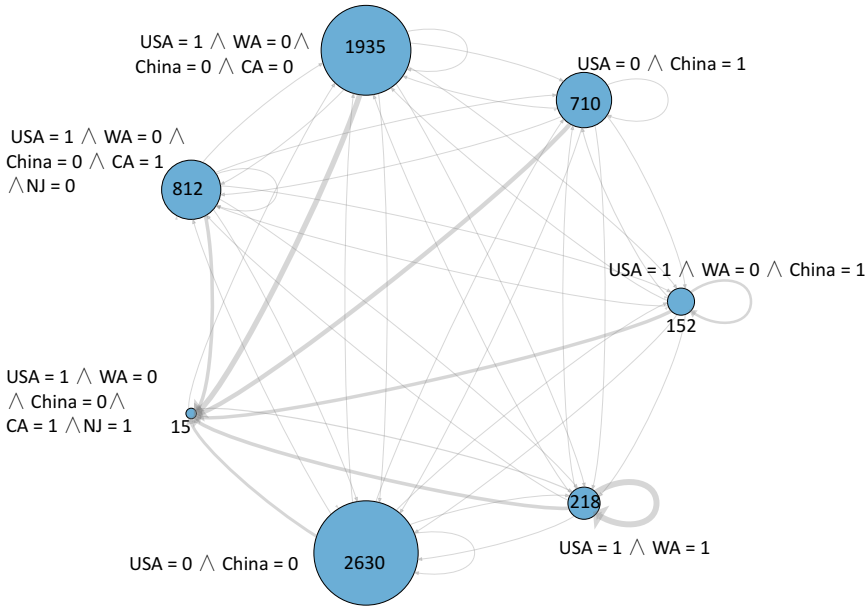


Figure 4.8: The resulting summary of DBLPaffs. Each supervertex (representing a paper subgroup) is labelled by its number of members (in the centre of the blue circle) and its description (near the blue circle). Each directed edge connects one supervertex to the other, and its linewidth indicates the connectivty density from a subgroup (e.g. $\varepsilon(W_1)$) to the other one (e.g., $\varepsilon(W_2)$). A thicker edge means the citations from $\varepsilon(W_1)$ to $\varepsilon(W_2)$ are more frequent).

The resulting summarization. By running our algorithm for 6 iterations, this citation network is summarized into 7 subgroups each consisting of papers satisfying a particular description about their authors' affiliations. These 7 subgroups are respectively defined by

1. USA $= 1$ and WA (Washington) $= 1$;
2. USA $= 1$ and WA $= 0$ and China $= 1$;

3. USA $=$ 1 and WA $=$ 0 and China $=$ 0 and CA (California) $=$ 1 and NJ (New Jersey) $=$ 1;
4. USA $=$ 1 and WA $=$ 0 and China $=$ 0 and CA $=$ 1 and NJ $=$ 0;
5. USA $=$ 1 and WA $=$ 0 and China $=$ 0 and CA $=$ 0;
6. USA $=$ 0 and China $=$ 1;
7. USA $=$ 0 and China $=$ 0.

The summary is displayed in Fig. 4.8. In the following, we discuss properties of local subgroup patterns revealed in our summarization to access its validity.

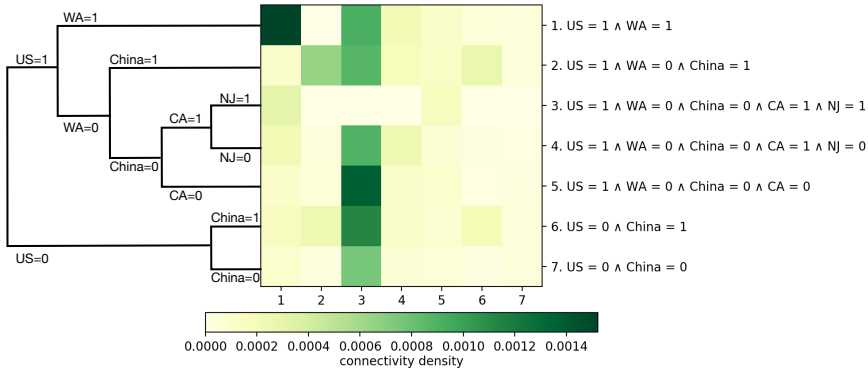


Figure 4.9: The heatmap representation of the density matrix for DBLPaffs, aligned with a dendrogram illustration of the splitting hierarchy on the left. A deeper color of each square indicates a higher connectivity density from a subgroup (represented by row) to another one (represented by column).

Remark 10. (Redundancy in the descriptions) *One may notice that some subgroup descriptions can be more concise. For example, the first subgroup pattern "USA = 1 and WA = 1" should induce the same extension as only"WA= 1". There is no mechanism in our approach for the global pattern mining that would prefer the alternative shorter description of the same subgroup. Yet, such redundancy can be easily identified and adjusted in post-processing. Moreover, this issue does not affect our single/bi-subgroup pattern mining approach where each iteration of the search essentially identifies an optimal pattern rather than a split (in global pattern mining approach), and shorter description of the same subgroup would have a larger SI value given by its smaller DL value.*

Discussion. A series of interesting local subgroup patterns emerge from the resulting summarization. The density matrix where its entry at the i-th row and the j-th column is the citation density from papers in the i-th subgroup to the j-th

is visualized by a heatmap, of which the left side is lined up with a dendrogram illustrating the splitting hierachy (see Fig. 4.9).

Obviously, the most cohesive subgroup are papers from institutions in Washington state in USA, as they cite those within this subgroup most frequently (indicated by the darkest green square in the top left). Closer inspection revealed that the majority of these papers are written by authors from Microsoft Corporation and the University of Washington.

The most highly-cited subgroup is the third one (indicated by the dark color of all the squares along the third column except the one in the third row). This subgroup only contains 15 papers, and their authors are affiliated to institutes in California and New Jersey, neither in Washington nor China. Note this also agrees with bi-subgroup patterns found in previous experiment for addressing RQ3 (Iteration 2 in Sec. 4.6.5). As already been pointed out, many of the world's largest high-tech corporations and reputable universities are located in this region. Examples include Silicon valley, Stanford university in CA, NEC Laboratories, AT&T Laboratories in NJ, among others.

Another interesting subgroup is the second one of which authors are with affiliations in China and USA (except Washington). Researchers related to this subgroup are surprisingly isolated, as their papers are seldom cited by those from other subgroups but very frequently (or to be more precise, the second most frequently) within this subgroup (indicated by the shallow color of all the squares along the second column except the one in the second row). In fact, Chinese affiliated with research organisations in China and Chinese affiliated with organisations in USA, have coauthored most papers in this subgroup. The reason of their isolation might be interesting for users to investigate. Again, this coincides with what we found in experiment for addressing RQ3 (Iteration 3 in Sec. 4.6.5). The difference is the identified subgroup here is more specified (i.e., also being with affiliation in USA except Washington).

A follow-up experiment. The rest subgroup defined by USA $= 0$ and China $= 0$ (i.e., the 7th one) contains a considerable number of members (indicated by the largest circle in Fig. 4.8). Continuing to run our algorithm for subsequent iterations tends to split this subgroup up such that some cohesive groups affiliated with organisations in other countries are revealed. For example, subgroups related to affiliations in Singapore, Canada, the Netherlands emerge respectively in the first 3 subsequent iterations (see the corresponding splitting hierarchy highlighted by red dashed lines in Fig. 4.10). They all cite papers within the same subgroup or those from the third subgroup (i.e., the overall most highly-cited one) very frequently (see rows $7, 8, 9$ of the heatmap in Fig 4.10).

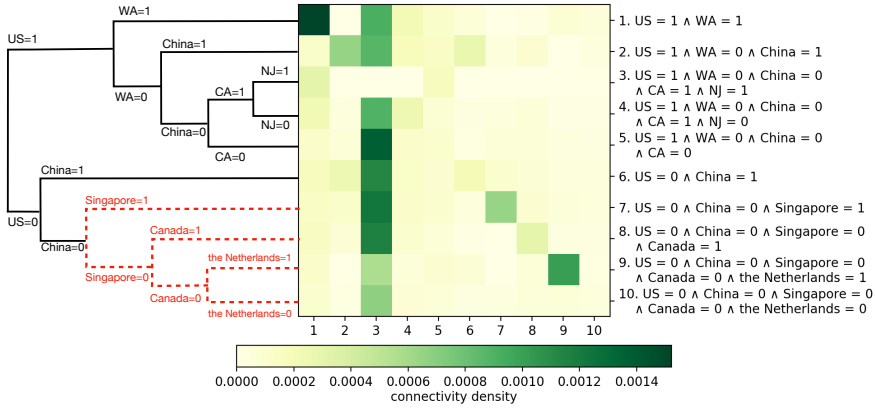


Figure 4.10: The heatmap representation of the density matrix among subgroups obtained by running our algorithm for another 3 subsequent iterations on DBLPaffs, with a dendrogram illustration of the splitting hierarchy on the left. A deeper color of each square indicates a higher connectivity density from a subgroup (represented by row) to another one (represented by column). The splitting hierarchy for the 3 new iterations are in red dashed lines.

4.6.6.2 Case study on *DBLPtopics*

Task. A user working for an academic organization may want to obtain a high-level view of citation vitality among different research fields. Given *DBLPtopics* dataset, we here show the global pattern identified by our summarization approach can provide such high-level view, revealing interesting local subgroup patterns of the form 'papers of study field A frequently (or rarely) cite those of field B'. We also show the obtained global pattern can provide the user further insights by linking with information about paper distribution among different conferences.

The resulting summarization. The summarization of *DBLPtopics* is generated by running our algorithm for 4 iterations, and the resulting summarization rule means to divide all papers into the following 5 subgroups:

1. $a_1 < Q_2^{a_1} \wedge a_8 \geq Q_1^{a_8}$ (Theoretical machine learning);
2. $a_1 < Q_2^{a_1} \wedge a_8 < Q_1^{a_8}$ (Practical machine learning);
3. $a_1 \geq Q_2^{a_1} \wedge a_5 < Q_3^{a_5} \wedge a_3 < Q_3^{a_3}$ (Data mining);
4. $a_1 \geq Q_2^{a_1} \wedge a_5 < Q_3^{a_5} \wedge a_3 \geq Q_3^{a_3}$ (Information retrieval);
5. $a_1 \geq Q_2^{a_1} \wedge a_5 \geq Q_3^{a_5}$ (Database).

For each subgroup, we list its original description and a corresponding short interpretation (in brackets) based on summarizing attributes' meaning. As mentioned previously (in Sec. 4.6.1), an attribute is essentially one of the first 50 LSI

Table 4.9: The meaning of attributes related to the resulting summarization.

Attribute	Meaning (Top 5 most strongly associated fields of study by absolute weight)
a_1	Data mining (0.55) Machine Learning (-0.49) Database (0.32) Computer Science (0.28) Information retrieval (0.25)
a_3	Data mining (0.41) Computer science (-0.40) Mathematics (0.39) Information retrieval (0.30) Pattern recognition (0.24)
a_5	Database (0.61) Information retrieval (-0.49) Query optimization (0.21) World Wide Web (-0.18) Mathematics (0.15)
a_8	Mathematical optimization (0.45) Information retrieval (0.44) Database (0.37) Data mining (-0.25) Computer science (0.22)

components for the original paper-topic matrix. Its meaning can thus be described by its 5 subcomponents with highest absolute weights (shown in Table. 4.9). A higher weight means this attribute's meaning is closer (positive sign) or more contrasting (negative sign) to this research field. We will use these short interpretations rather than original descriptions in the following part, because these are more straightforward. Generally, this summarization not only successfully captures those 4 research areas that publications in *DBLPtopics* are intended to cover (i.e., Machine Learning, Database, Information Retrieval, and Data Mining), but also identifies a deeper-level structure (i.e., the partition of machine leaning papers into two subgroups according to different aspects they emphasize: more practical or more theoretical).

The summary of *DBLPtopics* based on the resulting summarization rule is displayed in Fig. 4.11. To highlight the citation vitality between each pair of subgroups, the corresponding citation density matrix is visualized by a heatmap, lined up with a dendrogram on the left illustrating the splitting hierarchy (see Fig. 4.12).

Discussion. As shown in Fig. 4.12, the citation density within the same subgroup is often high, indicating papers of similar research field often cite each other.

Exceptions are the second (practical machine learning) subgroup and the third

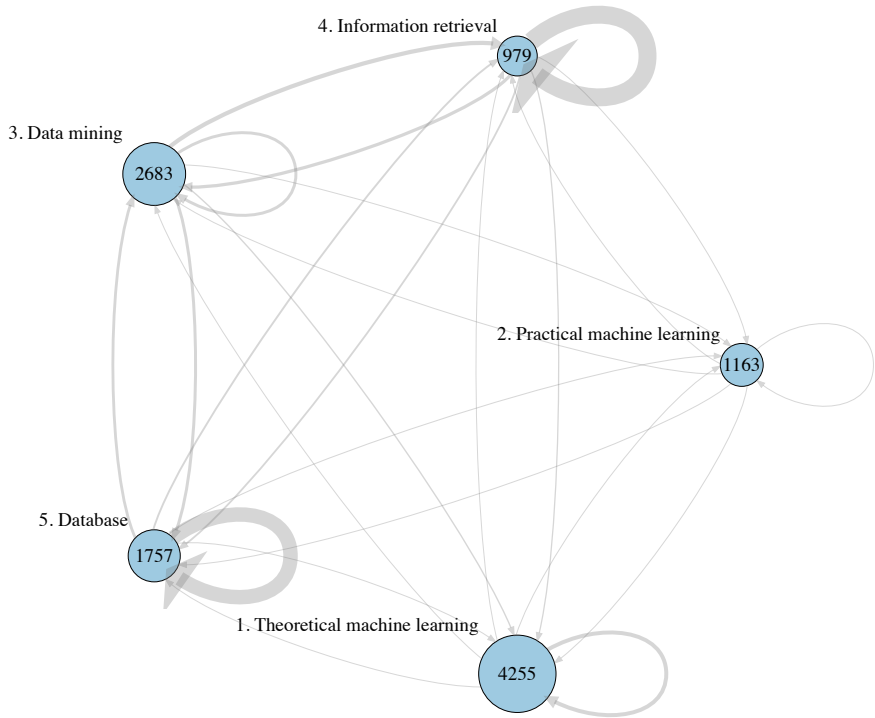


Figure 4.11: The resulting summary of DBLPtopics. Each supervertex (representing a paper subgroup) is labelled by its number of members (in the centre of the blue circle) and its description (near the blue circle). Each directed edge connects one supervertex to the other, and its linewidth indicates the connectivty density from a subgroup (e.g. $\varepsilon(W_1)$) to the other one (e.g., $\varepsilon(W_2)$). A thicker edge means the citations from $\varepsilon(W_1)$ to $\varepsilon(W_2)$ are more frequent).

one (data mining) which respectively cite the fifth (database) and the fourth (information retrieval) most frequently. This accords with the fact that solving data mining or practical machine learning research questions often necessitates database techniques or information retrieval to solve some subtasks.

Clearly, the fourth and the fifth subgroup are most cohesive (indicated by those two evidently dark green squares in the fourth and the fifth place of the diagonal). Also, these two groups cite each other and the data mining subgroup very frequently.

One downstream task: knowing more about conferences. The summarization generated by our approach can be useful in some downstream analysis tasks. Here we show an example of utilizing it to know more about conferences, simply by linking with the distribution of publications in those 20 selected conferences within each subgroup (displayed in Fig. 4.13).

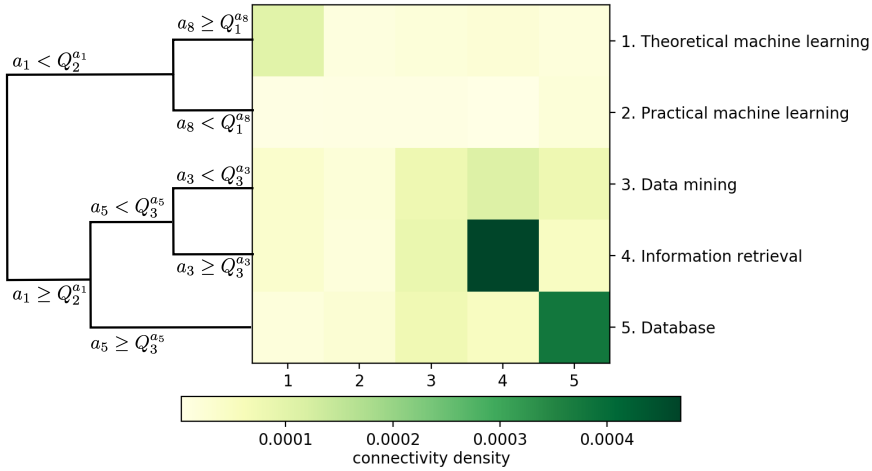


Figure 4.12: The heatmap representation of the density matrix for DBLPtopics, aligned with a dendrogram illustration of the splitting hierarchy on the left (Recall Q_i^a denotes the i-th quartile of values for the attribute a). A deeper color of each square indicates a higher connectivity density from a subgroup (represented by row) to another one (represented by column).

First, by merely looking at the distribution for each subgroup, the users can learn the relationship between research fields and conferences, e.g., answering questions like which research field is dominated by which conference. As can be seen, a noticeable large proportion of publications in regard to the information retrieval (the fourth subgroup) are in SIGIR and CIKM. and the database publications (the fifth subgroup) are mostly in ICDE, VLDB, SIGMOD. The data mining subgroup (the third one) is special in a sense that their publications are distributed quite evenly. WWW only holds a slim majority, and publications from KDD, AAAI, ICDM, CIKM are a little bit more than those from another venue (except WWW). Moreover, it is interesting to notice KDD and ICDM appear to be more interdisciplinary, accepting papers surprisingly evenly from these research areas compared to other conferences (as there is no noticeably longer dark brown or light green rectangular in either one of these 5 horizontal bins in Fig. 4.13).

Also, the user can combine Fig. 4.12 and Fig. 4.13 to deduce the citation vitality among different conferences. For example, publications in SIGIR and CIKM often cite those also in these two conferences (as the fourth subgroup is very cohesive), and they also often cite publications in WWW, AAAI, KDD,CIKM (those dominating the third subgroup).

Summary. As shown by these case studies on different datasets, global patterns identified by our method can not only directly provide insights by revealing a series of interesting single-subgroup and bi-subgroup patterns, but also be utilized to

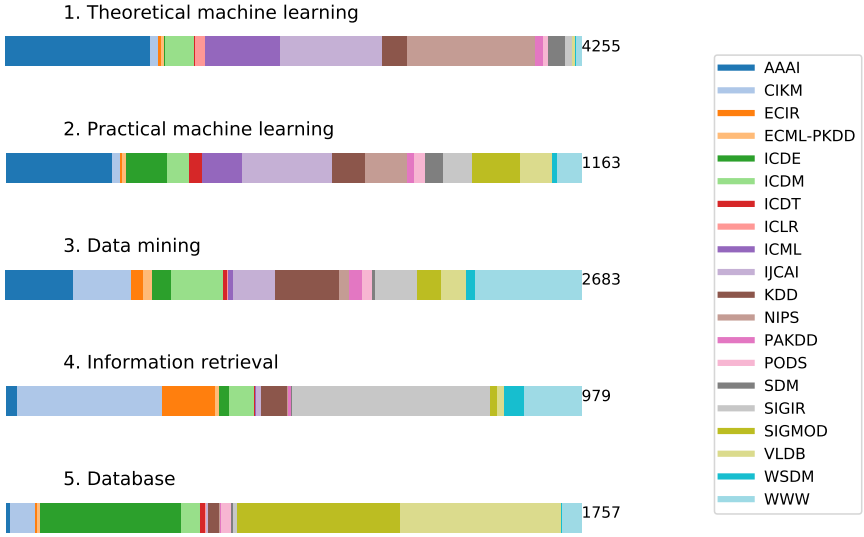


Figure 4.13: The distribution publications in 20 selected conferences within each subgroup. For each bin representing a subgroup, the subgroup description is placed on the top, and the number of papers in this subgroup is placed on the right end. The length of a rectangular in a certain color and hatch inside a bin is proportional to the percentage of publications in a certain conference in a subgroup. Conferences are in alphabetical order.

facilitate some downstream analysis tasks.

4.6.7 Scalability evaluation (RQ6)

4.6.7.1 Experimental setup

Choice of datasets. We used *Lastfm* to investigate the scalability to the number of selectors, because it can give a largest number of selectors (i.e., 21695) as the search space.

Other settings. Same as for other experiments, in the scalability evaluation, we applied the beam search with $x = 5$ (for single-subgroup pattern discovery), the nested beam search with $x_1 = 2$, $x_2 = 3$, and $D = 2$ (for bi-subgroup pattern discovery), 8 processors running in parallel (for global pattern mining).

4.6.7.2 Results

Effect of $|S|$. Fig. 4.14 displays run time on *Lastfm* w.r.t. the number of selectors in the search space (i.e., $|S|$). It is clear that, in either single-subgroup or global pattern mining, the run time experiences a linear growth as we gradually double the $|S|$ (from 10 to 20480), whereas the run time for bi-subgroup pattern mining

increases more than linearly, and exceeds 1 day when $|S|$ is larger than 2560.

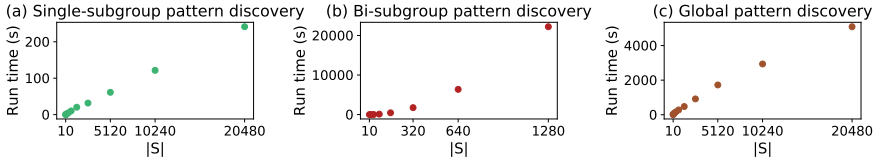


Figure 4.14: Run time (s) parametrized by $|S|$ on Lastfm.

Run time. The run time of our experiments for addressing RQ2 to RQ5, as well as the $|S|$ and $|V|$ statistics are listed in Table 4.10. The influence of the $|S|$ and $|V|$ on the run time is evident.

Table 4.10: Run time.

	Dataset	$\|S\|$	$\|V\|$	Run time (s)
Single-subgroup pattern mining (RQ2)	*Lastfm*	21695	1892	278.49
	DBLPaffs	232	6472	32.40
Bi-subgroup pattern mining (RQ3 and RQ4)	*Caltech36*	602	762	1312.57
	Reed98	748	962	1965.41
	Lastfm	200	1892	679.85
	DBLPaffs	232	6472	3114.78
Global pattern mining (RQ5)	*DBLPaffs*	232	6472	830.69
	DBLPtopics	150	10837	1570.90
	MPvotes	78	650	12.73

Summary. The run time grows linearly in the number of attributes in both single-subgroup and global pattern mining, whereas it grows faster than linearly in bi-subgroup pattern mining.

4.7 Conclusions

Prior work of pattern mining in attributed graphs typically only search for dense subgraphs ('communities') with homogenous attributes. We generalized this type of pattern to densities within this subgraph (no matter whether dense or sparse, which we refer as *single-subgroup pattern*), between a pair of different subgroups (which we refer as *bi-subgroup pattern*), as well as between all pairs from a set of subgroups that partition the whole vertex set (which we refer as *global pattern*).

We developed a novel information-theoretic approach for quantifying interestingness of such patterns in a subjective manner, with respect to a flexible type of

prior knowledge the user may have about the graph, including insights gained from previous patterns.

The empirical results show that our method can efficiently find interesting patterns of these new different types. In the standard problem of dense subgraph mining, our method can yield results that are superior to the state-of-the-art. We also demonstrated empirically that our method succeeds in taking in account prior knowledge in a meaningful way.

The proposed SI interestingness measure has considerable advantages, but a price to pay for this is in terms of computational time. To help maintain the tractability, we succumb to some accurate heuristic search strategies. It would be useful for the future work to discover a search strategy with performance guarantee and to further speed up the search (e.g., by branch and bounds).

acknowledgements

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, from the FWO (project no. G091017N, G0F9816N, 3G042220), and from the European Union's Horizon 2020 research and innovation programme and the FWO under the Marie Sklodowska-Curie Grant Agreement no. 665501.

Appendices

4.A For Section 4.6.3: A comparative evaluation on *DBLPaffs* network (RQ2)

Some objective interestingness measures we used for comparison, as well as their explanations are listed in Table 4.11.

We consider undirected graphs for the sake of presentation and consistency with most literature. However, we note that the generalization to directed graphs is straightforward.

4.B For Section 4.6.5: Evaluation on the iterative pattern mining on *Lastfm* Dataset (RQ4)

Table 4.14 displays the top 3 patterns found in each of the five iterations on the *Lastfm*. The description search space is built based on only 100 most frequently

Table 4.11: Existing measures for a comparison. For a given attributed graph $G = \{V, E, \Lambda\}$, and a community induced by a description W such that $\varepsilon(W) \in V$, $d(u)$ denotes the degree of vertex $u \in V$; $\overline{d}_W(u)$ denotes the inter-degree of vertex $u \in \varepsilon(W)$, specfically, $\overline{d}_W(u) := |\{(u, v) \in E : v \in V \setminus \varepsilon(W)\}|$; and #inter-edges denotes the number of connections between $\varepsilon(W)$ and $V \setminus \varepsilon(W)$.

Measure	Description	Mathematical definition
Edge density	the ratio of the number of edges to the number of possible edges in the cluster	$\frac{2*k_W}{\lvert\varepsilon(W)\rvert*(\lvert\varepsilon(W)\rvert-1)}$
Average degree	the ratio of the degree sum for all vertices to the number of vertices in the cluster	$\frac{2*k_W}{\lvert\varepsilon(W)\rvert}$
Pool's measure [14]	the reduction in the number of erroneous links between treating each vertex as a single community and treating all the vertices as a whole	$\sum_{u\in\varepsilon(W)} d(u) - \left(\frac{\lvert\varepsilon(W)\rvert*(\lvert\varepsilon(W)\rvert-1)}{2} - k_W\right)$ $-\#\text{inter-edges} = -\frac{\lvert\varepsilon(W)\rvert*(\lvert\varepsilon(W)\rvert-1)}{2} + 3 * k_W$
Edge Surplus [35]	the number of edges exceeding the expected number of edges within the cluster assuming each edge is present with the same probability α	$k_W - \alpha * \lvert\varepsilon(W)\rvert * (\lvert\varepsilon(W)\rvert - 1)$
Segregation index [113]	the difference between the number of expected inter-edges to the number of the observed inter-edges, normalized by the expectation	$1 - \frac{\#\text{inter-edges}*\lvert V\rvert*(\lvert V\rvert-1)}{2*\lvert E\rvert*\lvert\varepsilon(W)\rvert*(\lvert V\rvert-\lvert\varepsilon(W)\rvert)}$
Modularity of a single community [114, 115]	the measure quantifying the modularity contribution of a single community based on transforming the definition of modularity to a local measure	$\frac{1}{2*\lvert E\rvert}\sum_{u,v\in\varepsilon(W)}\left(a_{u,v} - \frac{d(u)*d(v)}{2*\lvert E\rvert}\right)$
Inverse Average-ODF (out-degree fraction) [116]	the inverse of the Average-ODF which is based on averaging the fraction of inter-degree and the degree for each vertex in the cluster	$1 - \frac{1}{\lvert\varepsilon(W)\rvert}\sum_{u\in\varepsilon(W)}\frac{\overline{d}_W(u)}{d(u)}$
Inverse Conductance	the ratio of the number of edges inside the cluster to the number of edges leaving the cluster	$\frac{k_W}{\#\text{inter-edges}}$

Table 4.12: Top 4 single-subgroup patterns w.r.t. our SI in DBLPaffs network. For each pattern (each row), we display values for elements that constitute the pattern syntax including W, I, k_W and also other statistics including its rank, $|\varepsilon(W)|$, $p_w \cdot n_W$ and #inter-edges (each column). k_W is the number of observed edges within $\varepsilon(W)$ (i.e., the set of vertices satisfying the description W), and $p_W \cdot n_W$ is the expected number of edges within $\varepsilon(W)$ w.r.t. the background distribution. I is the indicator equal to 0 if the observed pattern is dense for the user (i.e., $k_W > p_W \cdot n_W$) or 1 otherwise (i.e., $k_W < p_W \cdot n_W$). #inter-edges is the number of connections between $\varepsilon(W)$ and $V \setminus \varepsilon(W)$.

Rank	W	I	k_W	$\lvert\varepsilon(W)\rvert$	$p_W \cdot n_W$	#inter-edges
1	China $= 1$	0	179	873	63.20	566
2	China $= 1 \wedge$ IN (Indiana) $= 0$	0	179	869	62.58	561
3	China $= 1 \wedge$ Italy $= 0$	0	179	870	62.67	561
4	China $= 1 \wedge$ Denmark $= 0$	0	179	870	62.69	562

Table 4.13: Top 4 single-subgroup patterns w.r.t. other measures in DBLPaffs network. For each pattern (each row), we display values for elements that constitute the pattern syntax including W, I, k_W and also other statistics including $|\varepsilon(W)|$ and #inter-edges (each column). k_W is the number of observed edges within $\varepsilon(W)$ (i.e., the set of vertices satisfying the description W), and $p_W \cdot n_W$ is the expected number of edges within $\varepsilon(W)$ w.r.t. the background distribution. $I = 0$ in all cases as other measures can only quantify the interestingness of dense subgraphs. #inter-edges is the number of connections between $\varepsilon(W)$ and $V \setminus \varepsilon(W)$.

Measure	W	I	k_W	$\lvert\varepsilon(W)\rvert$	#inter-edges
Edge Density	DE (Delaware) $= 1 \wedge$ MD(Maryland) $= 1$	0	1	2	2
	DC (District of Columbia) $= 1 \wedge$ TX (Texas) $= 1$	0	1	2	6
	Netherlands $= 1 \wedge$ MA(Massachusetts) $= 1$	0	1	2	3
	Netherlands $= 1 \wedge$ WA $= 1$	0	1	2	5
Average Degree	UK $= 0 \wedge$ Japan $= 0$	0	2882	6038	161
	UK $= 0 \wedge$ Ireland $= 0$	0	2975	6234	79
	Japan $= 0 \wedge$ Ireland $= 0$	0	2952	6191	106
	Sweden $= 0 \wedge$ Ireland $= 0$	0	3044	6391	22
Pool's community score or Edge surplus	DE$= 1 \wedge$ MD$= 1$	0	1	2	2
	DC $= 1 \wedge$ TX $= 1$	0	1	2	6
	Netherlands $= 1 \wedge$ MA $= 1$	0	1	2	3
	Netherlands $= 1 \wedge$ WA $= 1$	0	1	2	5
Segregation Index	AL (Alabama)$= 0$	0	3066	6470	0
	AL $= 1$	0	0	2	0
	Bulgaria$= 0$	0	3066	6471	0
	AS (American Samoa)$= 0$	0	3066	6471	0
Modularity of a single community	China $= 0 \wedge$ United States $= 1$	0	1173	2969	1203
	NY(New York) $= 0 \wedge$ United States $= 1$	0	1067	2757	1224
	Singapore $= 0 \wedge$ United States $= 1$	0	1247	3088	1194
	Germany $= 0 \wedge$ United States $= 1$	0	1262	3077	1191

used tags, that means, $|S| = 100 \times 2$.

Iteration 1. Initially, we incorporate prior belief on individual vertex degree. The extracted most interesting pattern reflects a conflict between aggressive heavy metal fans and mainstream pop lovers who do not listen to heavy metal at all.

Iteration 2. After incorporating the top pattern identified in iteration 1, what comes top is the one expressing again a conflict between mainstream and non-mainstream music preference, but another kind (i.e., pop with no indie, and experimental with no pop). Also, we can notice only the second pattern for the iteration 1 is remained in the iteration 2 top list but with a lower rank as third. The interestingness of any sparse pattern associated with the newly incorporated one under the updated background distribution is expected to decrease, as the user's would not feel surprised about such pattern.

Iteration 3. In iteration 3, our method tends to identify some interesting dense patterns, mainly related to synth pop and new wave genres. The top one states synth pop fans frequently connect with many people listening to new wave but not synth pop. This pattern appears fallacious at the first glance. Nevertheless, synth pop is a subgenre of new wave music. Also, the latter group may listen to synth pop but they use a different tag 'synthpop' instead of 'synth pop', as there are even 102 audience only tag synth pop as 'synthpop' (see the third patten). Hence, this pattern makes sense as it describes dense connections between two groups which resemble each other.

Iteration 4. The top 3 patterns in iteration 4 all express negative associations between new wave and some sort of catchy mainstream music (eg. pop, rnb, or hip-hop, among several others).

Iteration 5. Once we incorporate the most interesting one, patterns characterizing some positively associated genres stand out. For example, the top one in iteration 5 indicates instrumental audience are friends with many ambient audience who doesn't listen to instrumental music. These two genres are not opposite concepts and share many in common (e.g., recordings for both do not include lyrics). Actually, ambient music can be regarded as a slow form of instrumental music.

Summary. By incorporating the newly obtained patterns into the background distribution for subsequent iterations, our method can identify patterns which strongly contrast to this knowledge. This results in a set of patterns that are not redundant and are highly surprising to the user. Note this does not means we restrict patterns in different iterations not to be associated with each other. In fact, overlapping could happen when this is informative.

Table 4.14: Top 3 discovered bi-subgroup patterns of each iteration in Lastfm network. For each pattern (each row), we display values for elements that constitute the pattern syntax including W_1, W_2, I, k_W, and also other statistics including its rank, $|\varepsilon(W_1)|$, $|\varepsilon(W_1)|$, and $p_w \cdot n_W$ (each column). k_W is the number of observed connections between $\varepsilon(W_1)$ (i.e., vertices satisfying the description W_1) and $\varepsilon(W_2)$ (i.e., vertices satisfying the description W_2), and $p_W \cdot n_W$ is the expected number of connections between $\varepsilon(W_1)$ to $\varepsilon(W_2)$ w.r.t. the background distribution. I is the indicator equal to 0 if the observed pattern is dense for the user (i.e., $k_W > p_W \cdot n_W$) or 1 otherwise (i.e., $k_W < p_W \cdot n_W$).

	Rank	W_1	W_2	I	k_W	$\lvert\varepsilon(W_1)\rvert$	$\lvert\varepsilon(W_2)\rvert$	$p_W \cdot n_W$
Iteration 1	1	heavy mental $= 1$	heavy mental $= 0 \wedge$ pop $= 1$	1	349	165	529	769.18
	2	pop$= 1 \wedge$experimental $= 0$	rnb $= 0 \wedge$experimental$= 1$	1	360	497	230	812.78
	3	pop$= 1 \wedge$experimental $= 0$	experimental $= 1$	1	495	497	247	943.96
Iteration 2	1	pop $= 1 \wedge$ indie $= 0$	pop $= 0 \wedge$experimental $= 1$	1	103	366	159	369.44
	2	pop $= 1 \wedge$ alternative $= 0$	pop $= 0 \wedge$ experimental $= 1$	1	84	325	159	334.77
	3	pop $= 1 \wedge$ experimental $= 0$	rnb $= 0 \wedge$ experimental $= 1$	1	360	497	230	750.77
Iteration 3	1	synth pop $= 1$	synth pop $= 0 \wedge$ new wave $= 1$	0	163	54	150	43.10
	2	synth pop $= 1 \wedge$ british $= 1$	new wave $= 1 \wedge$british $= 0$	0	116	26	113	20.71
	3	synth pop $= 1$	synth pop $= 0 \wedge$ synthpop $= 1$	0	125	54	102	29.64
Iteration 4	1	new wave $= 1 \wedge$ hip-hop $= 0$	new wave $= 0 \wedge$ pop $= 1$	1	160	475	343	670.74
	2	new wave $= 1 \wedge$ rnb $= 0$	new wave $= 0 \wedge$ pop $= 1$	1	379	170	475	705.43
	3	new wave $= 1 \wedge$ soul $= 0$	new wave $= 0 \wedge$ pop $= 1$	1	323	150	475	624.41
Iteration 5	1	instrumental $= 1$	instrumental $= 0 \wedge$ ambient $= 1$	0	273	195	144	114.62
	2	electronic $= 1$	electronic $= 0 \wedge$ ambient $= 1$	0	268	167	160	113.66
	3	progressive metal $= 1$	progressive metal $= 0 \wedge$ heavy metal $= 1$	0	128	99	111	34.81

4.C For Section 4.6.6: One more case study on *MPvotes* for the evaluation of global pattern mining

Task. Brexit is a hot topic of debate in UK. MPs' voting behaviours on Brexit might affect the likelihood of their connections. Using this information to summarize MPs friendship network is thus potential to provide insights on the Brexit saga. We here investigate whether our approach can achieve this.

The resulting summarization. The summarization of *MPvotes* generated from running our algorithm for 4 iterations splits all MPs into 5 subgroups, and they are respectively defined by

1. I1 $= -1$ or $0 \wedge$ I10 V3 $= -1$ or $0 \wedge$ I10 V4 $= -1$ or 0;
2. I1 $= -1$ or $0 \wedge$ I10 V3 $= -1$ or $0 \wedge$ I10 V4 $= 1$;
3. I1 $= -1$ or $0 \wedge$ I10 V3 $= 1$;
4. I1 $= 1 \wedge$ I7 V4 $= 1$ or 0;
5. I1 $= 1 \wedge$ I7 V4 $= -1$.

where 'Ii Vj' represents the j-th vote in the i-th issue. For an issue around which there exists only one vote, say the 1st issue, it is simply represented as I1. Detailed interpretation of all voting issues related to our summarization are displayed in Table 4.15. The summary of *MPvotes* is illustrated in Fig. 4.15. For a dedicated view of the connectivity density between each subgroup pair, the corresponding density matrix is visualized by a heatmap, aligned with an dendrogram illustration of the splitting hierachy on the left (see Fig. 4.16).

Table 4.15: The description of voting issues related to the resulting summarization in the order of spliting.

Vote Notation	Description
I1	Government in rejecting an amendment that would have given MPs the power to stop the UK from leaving the EU without a deal.
I10 V3	Labour's plan for a close economic relationship with the EU.
I10 V4	UK membership of the European Free Trade Association (Efta) and European Economic Area (EEA).
I7 V4	Government in contempt of parliament

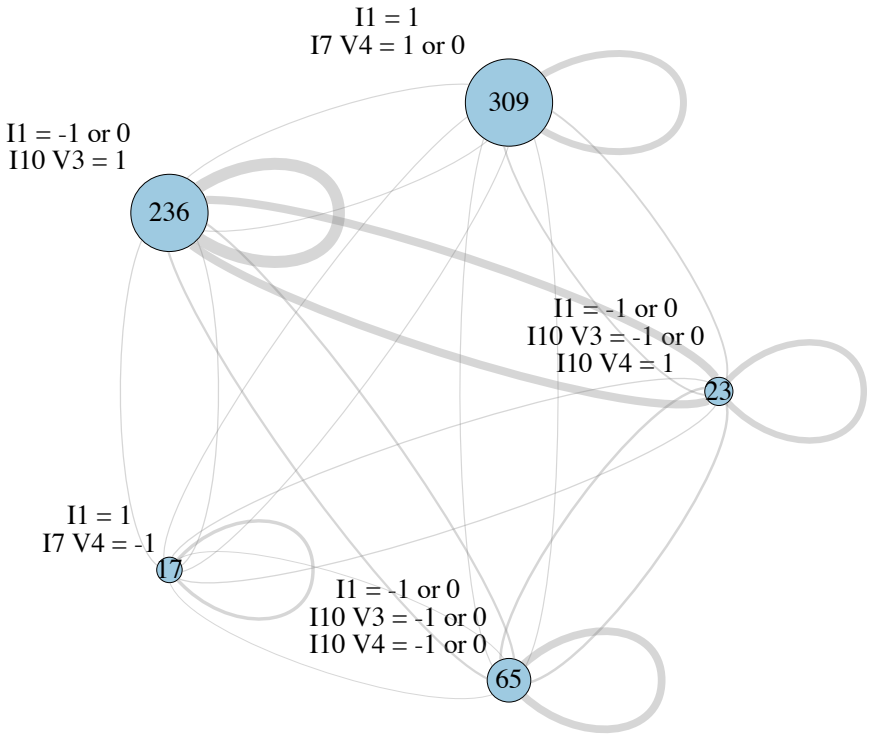


Figure 4.15: The resulting summary of MPvotes. Each supervertex (representing a subgroup of MPs) is labelled by its number of members (in the centre of the blue circle) and its description (near the blue circle). Each undirected edge connects between one supervertex and the other, with its linewidth indicating the connectivity density between these two corresponding subgroups (The thicker the edge, the higher the connectivity density).

Discussion. Clearly in Fig. 4.16, our summarization identifies several crucial votings that partition MPs into cohesive subgroups. That is, MPs taking the same sides in these votings connect more frequently to each other (i.e., those within the same subgroup) than MPs voting differently (i.e., those in other subgroups). The only exception is the 2nd subgroup who connect most frequently to the 3rd subgroup. More interpretations of these patterns are provided in the following.

Combining with political parties. The user can utilize our summarization of *MPvotes* to obtain insights about Brexit saga. Here, we provide one example. More specifically, we show, by combining with the distribution of MPs' party affiliations within each subgroup (illustrated in Fig. 4.17), our summarization can:

(a) reveal crucial voting issues over which MPs from different parties take different sides;

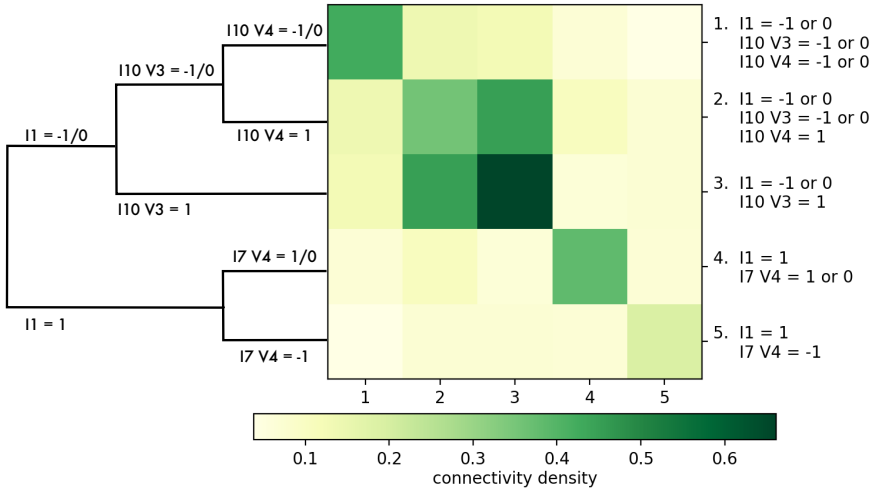


Figure 4.16: The heatmap representation of the density matrix among subgroups obtained by running our algorithm for 4 iterations on MPvotes, aligned with a dendrogram illustration of the splitting hierarchy on the left. A darker color of each square indicates a higher connectivity density between a subgroup (represented by row) and another one (represented by column).

(b) provide a high-level view of connectivity densities among different political parties.

Now we trace the partition process based on our summarization in order to show (a). The first split is a vote on I1 of which 'ayes' side with the government to keep no-deal Brexit on the table as a possibility (see the dendrogram in Fig. 4.17). A clear opinion conflict between different parties can be observed. More specifically, all the MPs from Scottish National Party (SNP), Liberal Democrat (LD), Sinn Fein (SF), Plaid Cymru (PC), Green (Grn) and the majority of MPs in Labour (Lab) voted against I1 or abstained (the aggregation of the first, second and third subgroup). All except two MPs from Conservative (Con) and all from Democratic Unionist Party (DUP) were in favour (the aggregation of the fourth and fifth subgroup). Then those 'Noes' and abstainers of I1 are divided according to their stances on Lab's plan for a close economic relationship with the EU (i.e., I10 V3). 'Ayes' of I10 V3 (i.e., the third subgroup) are dominated by most MPs from Lab. The others are further split over their votes on UK membership of Efta and Eea (i.e., I10 V4), in which MPs from some non-mainstream parties voted for or abstained (i.e., the firstst subgroup) and 15 MPs from Lab voted against. In the fourth split of vote on I7 V4, MPs affiliated with Con and those with DUP are clearly separated from each other, leading to the fourth and fifth subgroup respectively.

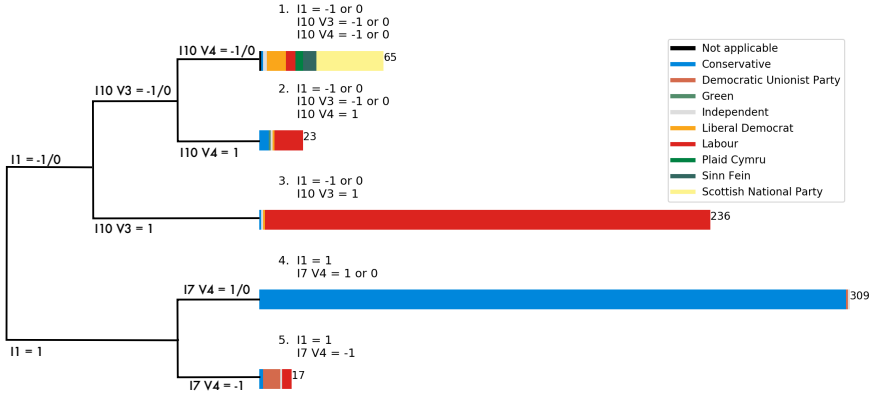


Figure 4.17: The distribution of party affiliations of MPs in each subgroup, aligned with a dendrogram illustrating the splitting hierarchy on the left. For each bin corresponding to a subgroup, the subgroup description is placed on the top, and the number of MPs in this subgroup is placed on the right end. The rectangular length of a particular color inside a bin is proportional to the number of MPs affiliated with a particular party in this subgroup.

Then we show (b) by combining our summarization (Fig. 4.16) and the party affiliation distribution (Fig. 4.17). Here we show some interesting findings. As mentioned previously, one bi-subgroup pattern reveals frequent connections between the second subgroup and the third one. The second subgroup can be interpreted as a group of unrepresentative Lab MPs, whereases the third subgroup corresponds to a representative group, as closer inspection shows MPs in either of these two subgroups are mostly affiliated with Lab, though the population of the second subgroup is much smaller. Also, MPs affiliated with some non-mainstream parties (e.g., SNP, LD,SF,PC) connect much more to those affiliated with Lab than those with Con, especially those with Lab belonging to the second subgroup. Although the fourth subgroup is almost made up with purely MPs that are from Con, its relatively small self-connectivity in comparison with that to the first and the third subgroup indicates not many MPs from Con build friendship with each other.

References

[1] T. L. Fond and J. Neville. *Randomization tests for distinguishing social influence and homophily effects*. In Proceedings of the 19th international conference on World wide web, pages 601–610, 2010.

[2] M. McPherson, L. Smith-Lovin, and J. M. Cook. *Birds of a Feather: Homophily in Social Networks*. Annual Review of Sociology, 27(1):415–444, 2001.

[3] S. Aral, L. Muchnik, and A. Sundararajan. *Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks*. Proceedings of the National Academy of Sciences, 106(51):21544–21549, 2009.

[4] J. Li, L. Wu, O. Zaïane, and H. Liu. *Toward personalized relational learning*. In Proceedings of the 2017 SIAM International Conference on Data Mining, pages 444–452, 2017.

[5] N. Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, and D. Song. *Joint link prediction and attribute inference using a social-attribute network*. ACM Transactions on Intelligent Systems and Technology, 5(2):1–20, 2014.

[6] Z. Yin, M. Gupta, T. Weninger, and J. Han. *A Unified Framework for Link Recommendation Using Random Walks*. In 2010 International Conference on Advances in Social Networks Analysis and Mining, pages 152–159, 2010.

[7] N. Barbieri, F. Bonchi, and G. Manco. *Who to follow and why: link prediction with explanations*. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1266–1275, 2014.

[8] X. Wei, L. Xu, B. Cao, and P. S. Yu. *Cross View Link Prediction by Learning Noise-Resilient Representation Consensus*. In Proceedings of the 26th International Conference on World Wide Web, page 1611–1619, 2017.

[9] F. Herrera, C. J. Carmona, P. González, and M. J. del Jesus. *An overview on subgroup discovery: foundations and applications*. Knowledge and Information Systems, 29(3):495–525, 2011.

[10] M. Atzmueller. *Subgroup discovery*. WIREs Data Mining and Knowledge Discovery, 5(1):35–49, 2015. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1144.

[11] M. Atzmueller, S. Doerfel, and F. Mitzlaff. *Description-oriented community detection using exhaustive subgroup discovery*. Information Sciences, 329:965 – 984, 2016. Special issue on Discovery Science.

[12] F. Moser, R. Colak, A. Rafiey, and M. Ester. *Mining Cohesive Patterns from Graphs with Feature Vectors*. In Proceedings of the 2009 SIAM International Conference on Data Mining, pages 593–604.

[13] P.-N. Mougel, M. Plantevit, C. Rigotti, O. Gandrillon, and J.-F. Boulicaut. *Constraint-Based Mining of Sets of Cliques Sharing Vertex Properties*. In Workshop on Analysis of Complex Networks ACNE'10 co-located with ECML PKDD 2010, pages 48–62, 2010.

[14] S. Pool, F. Bonchi, and M. v. Leeuwen. *Description-Driven Community Detection*. ACM Transactions on Intelligent Systems and Technology, 5(2):28:1–28:28, 2014.

[15] E. Galbrun, A. Gionis, and N. Tatti. *Overlapping Community Detection in Labeled Graphs*. Data Mining and Knowledge Discovery, 28(5-6):1586–1610, 2014.

[16] T. De Bie. *An Information Theoretic Framework for Data Mining*. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 564–572, 2011.

[17] T. De Bie. *Subjective Interestingness in Exploratory Data Mining*. In Proceedings of the 12th International Symposium on Advances in Intelligent Data Analysis, pages 19–31, 2013.

[18] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal. *A Survey of Algorithms for Dense Subgraph Discovery*, pages 303–336. 2010.

[19] D. Eppstein, M. Löffler, and D. Strash. *Listing All Maximal Cliques in Sparse Graphs in Near-Optimal Time*. In O. Cheong, K.-Y. Chwa, and K. Park, editors, Algorithms and Computation, pages 403–414, 2010.

[20] A. Gély, L. Nourine, and B. Sadi. *Enumeration aspects of maximal cliques and bicliques*. Discrete Applied Mathematics, 157(7):1447 – 1459, 2009.

[21] T. Uno. *An efficient algorithm for solving pseudo clique enumeration problem*. Algorithmica, 56(1):3–16, 2010.

[22] J. Abello, M. G. Resende, and S. Sudarsky. *Massive quasi-clique detection*. In Latin American symposium on theoretical informatics, pages 598–612, 2002.

[23] G. Liu and L. Wong. *Effective Pruning Techniques for Mining Quasi-Cliques*. In W. Daelemans, B. Goethals, and K. Morik, editors, Machine Learning and Knowledge Discovery in Databases, pages 33–49, 2008.

[24] J. Pei, D. Jiang, and A. Zhang. *On Mining Cross-Graph Quasi-Cliques*. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pages 228–238, 2005.

[25] Z. Zeng, J. Wang, L. Zhou, and G. Karypis. *Coherent Closed Quasi-Clique Discovery from Large Dense Graph Databases*. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 797–802, 2006.

[26] K. Shin, T. Eliassi-Rad, and C. Faloutsos. *Corescope: Graph mining using k-core analysis–patterns, anomalies and algorithms*. In 2016 IEEE 16th international conference on data mining (ICDM), pages 469–478, 2016.

[27] Y. Okubo, M. Matsudaira, and M. Haraguchi. *Detecting Maximum k-Plex with Iterative Proper l-Plex Search*. In International Conference on Discovery Science, pages 240–251, 2014.

[28] J. Gao, J. Chen, M. Yin, R. Chen, and Y. Wang. *An Exact Algorithm for Maximum k-Plexes in Massive Graphs*. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, pages 1449–1455, 2018.

[29] J.-M. Bourjolly, G. Laporte, and G. Pesant. *An exact algorithm for the maximum k-club problem in an undirected graph*. European Journal of Operational Research, 138(1):21 – 28, 2002.

[30] S. Shahinpour and S. Butenko. *Algorithms for the maximum k-club problem in graphs*. Journal of Combinatorial Optimization, 26(3):520–554, 2013.

[31] X. Ma, G. Zhou, J. Shang, J. Wang, J. Peng, and J. Han. *Detection of Complexes in Biological Networks Through Diversified Dense Subgraph Mining*. Journal of Computational Biology, 24(9):923–941, 2017.

[32] L. Qin, R.-H. Li, L. Chang, and C. Zhang. *Locally densest subgraph discovery*. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 965–974, 2015.

[33] S. Khuller and B. Saha. *On Finding Dense Subgraphs*. In S. Albers, A. Marchetti-Spaccamela, Y. Matias, S. Nikoletseas, and W. Thomas, editors, Automata, Languages and Programming, pages 597–608, 2009.

[34] M. E. J. Newman and M. Girvan. *Finding and evaluating community structure in networks*. Physical Review E, 69(2), 2004.

[35] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli. *Denser Than the Densest Subgraph: Extracting Optimal Quasi-cliques with Quality Guarantees*. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 104–112, 2013.

[36] A. Inokuchi, T. Washio, and H. Motoda. *An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data*. In Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, pages 13–23, 2000.

[37] L. B. Holder, D. J. Cook, and S. Djoko. *Substructure Discovery in the SUBDUE System*. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, pages 169–180, 1994.

[38] M. Kuramochi and G. Karypis. *An efficient algorithm for discovering frequent subgraphs*. IEEE Transactions on Knowledge and Data Engineering, 16(9):1038–1051, 2004.

[39] J. Huan, W. Wang, and J. Prins. *Efficient mining of frequent subgraphs in the presence of isomorphism*. In Third IEEE International Conference on Data Mining, pages 549–552, 2003.

[40] M. Kuramochi and G. Karypis. *Finding frequent patterns in a large sparse graph*. Data Mining and Knowledge Discovery, 11(3):243–271, 2005.

[41] Xifeng Yan and Jiawei Han. *gSpan: graph-based substructure pattern mining*. In 2002 IEEE International Conference on Data Mining, 2002. Proceedings., pages 721–724, 2002.

[42] X. Yan and J. Han. *CloseGraph: Mining Closed Frequent Graph Patterns*. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 286–295, 2003.

[43] T. Meinl, M. Wörlein, I. Fischer, and M. Philippsen. *Mining Molecular Datasets on Symmetric Multiprocessor Systems*. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pages 1269–1274, 2006.

[44] M. Berlingerio, F. Bonchi, B. Bringmann, and A. Gionis. *Mining Graph Evolution Rules*. In W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, editors, Machine Learning and Knowledge Discovery in Databases, pages 115–130, 2009.

[45] N. Vanetik, E. Gudes, and S. E. Shimony. *Computing Frequent Graph Patterns from Semistructured Data*. In Proceedings of the 2002 IEEE International Conference on Data Mining, page 458, 2002.

[46] I. Wegener and R. Pruim. *Complexity Theory: Exploring the Limits of Efficient Algorithms*. Springer Science & Business Media, 2005.

[47] J. Huan, W. Wang, J. Prins, and J. Yang. *SPIN: Mining Maximal Frequent Subgraphs from Graph Databases*. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 581–586, 2004.

[48] C. Borgelt and M. R. Berthold. *Mining Molecular Fragments: Finding Relevant Substructures of Molecules*. In Proceedings of the 2002 IEEE International Conference on Data Mining, page 51, 2002.

[49] S. Nijssen and J. N. Kok. *A Quickstart in Frequent Structure Mining Can Make a Difference*. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 647–652, 2004.

[50] B. Güvenoglu and B. E. Bostanoglu. *A qualitative survey on frequent subgraph mining*. Open Computer Science, 8:194 – 209, 2018.

[51] A. Silva, W. Meira Jr, and M. J. Zaki. *Mining Attribute-structure Correlated Patterns in Large Attributed Graphs*. Proceedings of the VLDB Endowment, 5(5), 2012.

[52] A. Khan, X. Yan, and K.-L. Wu. *Towards Proximity Pattern Mining in Large Graphs*. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pages 867–878, 2010.

[53] A. Bendimerad, M. Plantevit, and C. Robardet. *Mining exceptional closed patterns in attributed graphs*. Knowledge and Information Systems (KAIS), 56(1):1 – 25, 2018.

[54] A. Bendimerad, A. Mel, J. Lijffijt, M. Plantevit, C. Robardet, and T. De Bie. *SIAS-miner : mining subjectively interesting attributed subgraphs*. DATA MINING AND KNOWLEDGE DISCOVERY, 34:355–393, 2020.

[55] Y. Liu, T. Safavi, A. Dighe, and D. Koutra. *Graph Summarization Methods and Applications: A Survey*. ACM Computing Surveys (CSUR), 51(3):62:1–62:34, 2018.

[56] Y. Tian, R. A. Hankins, and J. M. Patel. *Efficient Aggregation for Graph Summarization*. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pages 567–580, 2008.

[57] E. F. Codd, S. B. Codd, and C. T. Salley. *Providing OLAP (on-line analytical processing) to user-analysts*. An IT mandate. Technical report, 230:230, 1993.

[58] S. Chaudhuri and U. Dayal. *An Overview of Data Warehousing and OLAP Technology*. ACM Sigmod record, 26(1):65–74, 1997.

[59] N. Zhang, Y. Tian, and J. M. Patel. *Discovery-driven graph summarization*. In 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), pages 880–891, 2010.

[60] F. Chen, X. Yan, F. Zhu, J. Han, and P. Yu. *Graph OLAP: Towards Online Analytical Processing on Graphs*. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, pages 103–112, 2008.

[61] C. Chen, C. X. Lin, M. Fredrikson, M. Christodorescu, X. Yan, and J. Han. *Mining Graph Patterns Efficiently via Randomized Summaries*. Proceedings of the VLDB Endowment, 2:742–753, 2009.

[62] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos. *Pics: Parameter-free identification of cohesive subgroups in large attributed graphs*. In Proceedings of the 2012 SIAM international conference on data mining, pages 439–450, 2012.

[63] Y. Wu, Z. Zhong, W. Xiong, and N. Jing. *Graph summarization for attributed graphs*. In 2014 International Conference on Information Science, Electronics and Electrical Engineering, volume 1, pages 503–507, 2014.

[64] K. U. Khan, W. Nawaz, and Y. Lee. *Set-Based Unified Approach for Attributed Graph Summarization*. In 2014 IEEE Fourth International Conference on Big Data and Cloud Computing, pages 378–385, 2014.

[65] P. D. Grünwald and A. Grunwald. *The minimum description length principle*. MIT press, 2007.

[66] N. Hassanlou, M. Shoaran, and A. Thomo. *Probabilistic Graph Summarization*. In J. Wang, H. Xiong, Y. Ishikawa, J. Xu, and J. Zhou, editors, Web-Age Information Management, pages 545–556, 2013.

[67] N. Ashrafi Payaman and M. Kangavari. *GSSC: Graph Summarization based on both Structure and Concepts*. International Journal of Information and Communication Technology Research, 9(1):33–44, 2017.

[68] Q. Song, Y. Wu, P. Lin, L. X. Dong, and H. Sun. *Mining Summaries for Knowledge Graph Search*. IEEE Transactions on Knowledge and Data Engineering, 30(10):1887–1900, 2018.

[69] L. Shi, H. Tong, J. Tang, and C. Lin. *VEGAS: Visual influEnce GrAph Summarization on Citation Networks*. IEEE Transactions on Knowledge and Data Engineering, 27(12):3417–3431, 2015.

[70] B. Adhikari, Y. Zhang, A. Bharadwaj, and B. A. Prakash. *Condensing Temporal Networks using Propagation*, pages 417–425.

[71] Y. Zhou, H. Cheng, and J. X. Yu. *Graph Clustering Based on Structural/Attribute Similarities*. Proceedings of the VLDB Endowment, 2(1):718–729, 2009.

[72] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng. *A Model-based Approach to Attributed Graph Clustering*. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pages 505–516, 2012.

[73] H. Cheng, Y. Zhou, and J. X. Yu. *Clustering Large Attributed Graphs: A Balance Between Structural and Attribute Similarities*. ACM Transactions on Knowledge Discovery from Data, 5:12:1–12:33, 2011.

[74] S. "Günnemann, I. Farber, B. Boden, and T. Seidl. *Subspace Clustering Meets Dense Subgraph Mining: A Synthesis of Two Paradigms*. In 2010 IEEE International Conference on Data Mining, pages 845–850, 2010.

[75] S. Günnemann, B. Boden, and T. Seidl. *DB-CSC: A Density-Based Approach for Subspace Clustering in Graphs with Feature Vectors*. In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, editors, Machine Learning and Knowledge Discovery in Databases, pages 565–580, 2011.

[76] L. Parsons, E. Haque, and H. Liu. *Subspace clustering for high dimensional data: a review*. ACM SIGKDD Explorations Newsletter, 6(1):90–105, 2004.

[77] B. Perozzi, L. Akoglu, P. Iglesias Sánchez, and E. Müller. *Focused Clustering and Outlier Detection in Large Attributed Graphs*. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1346–1355, 2014.

[78] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang. *Semantic Community Identification in Large Attribute Networks*. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pages 265–271, 2016.

[79] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang. *Attributed Graph Clustering: a Deep Attentional Embedding approach*. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pages 3670–3676, 2019.

[80] S. T. Roweis and L. K. Saul. *Nonlinear dimensionality reduction by locally linear embedding*. Science, 290(5500):2323–2326, 2000.

[81] M. Belkin and P. Niyogi. *Laplacian eigenmaps and spectral techniques for embedding and clustering*. Advances in Neural Information Processing Systems, 14:585–591, 2001.

[82] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola. *Distributed large-scale natural graph factorization*. In Proceedings of the 22nd International Conference on World Wide Web, pages 37–48, 2013.

[83] S. Cao, W. Lu, and Q. Xu. *Grarep: Learning graph representations with global structural information*. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pages 891–900, 2015.

[84] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu. *Asymmetric transitivity preserving graph embedding*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1105–1114, 2016.

[85] B. Perozzi, R. Al-Rfou, and S. Skiena. *Deepwalk: Online learning of social representations*. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 701–710, 2014.

[86] A. Grover and J. Leskovec. *node2vec: Scalable feature learning for networks*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 855–864, 2016.

[87] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo. *struc2vec: Learning node representations from structural identity*. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 385–394, 2017.

[88] H. Chen, B. Perozzi, Y. Hu, and S. Skiena. *Harp: Hierarchical representation learning for networks*. arXiv preprint arXiv:1706.07845, 2017.

[89] D. Wang, P. Cui, and W. Zhu. *Structural deep network embedding*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1225–1234, 2016.

[90] Y.-A. Lai, C.-C. Hsu, W. H. Chen, M.-Y. Yeh, and S.-D. Lin. *PRUNE: Preserving Proximity and Global Ranking for Network Embedding*. In Advances in Neural Information Processing Systems, volume 30, pages 5257–5266, 2017.

[91] A. Tsitsulin, D. Mottin, P. Karras, and E. Müller. *Verse: Versatile graph embeddings from similarity measures*. In Proceedings of the 2018 World Wide Web Conference, pages 539–548, 2018.

[92] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. *Line: Large-scale information network embedding*. In Proceedings of the 24th International Conference on World Wide Web, pages 1067–1077, 2015.

[93] B. Kang, J. Lijffijt, and T. De Bie. *Conditional network embeddings*. In 7th International Conference on Learning Representations, page 16, 2019.

[94] P. Goyal and E. Ferrara. *Graph embedding techniques, applications, and performance: A survey*. Knowledge-Based Systems, 151:78 – 94, 2018.

[95] A. C. Mara, J. Lijffijt, and T. De Bie. *Benchmarking network embedding models for link prediction: are we making progress?* In 7th IEEE International Conference on Data Science and Advanced Analytics, page 10, 2020.

[96] P. W. Holland and S. Leinhardt. *An Exponential Family of Probability Distributions for Directed Graphs*. Journal of the American Statistical Association, 76(373):33–50, 1981.

[97] J. K. Harris. *An introduction to exponential random graph modeling*, volume 173. Sage Publications, 2013.

[98] A. Fronczak. *Exponential random graph models*, 2012. arXiv:1210.7828.

[99] G. Casiraghi, V. Nanumyan, I. Scholtes, and F. Schweitzer. *Generalized Hypergeometric Ensembles: Statistical Hypothesis Testing in Complex Networks*, 2016. arXiv:1607.02441.

[100] M. Leeuwen, T. Bie, E. Spyropoulou, and C. Mesnage. *Subjective Interestingness of Subgraph Patterns*. Machine Learning, 105(1):41–75, 2016.

[101] T. De Bie. *Maximum Entropy Models and Subjective Interestingness: An Application to Tiles in Binary Databases*. Data Mining and Knowledge Discovery, 23(3):407–446, 2011.

[102] F. Adriaens, J. Lijffijt, and T. De Bie. *Subjectively interesting connecting trees*. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, volume 10535, pages 53–69, 2017.

[103] H. Chernoff. *A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations*. The Annals of Mathematical Statistics, 23(4):493–507, 1952.

[104] W. Hoeffding. *Probability Inequalities for Sums of Bounded Random Variables*. Journal of the American Statistical Association, 58(301):13–30, 1963.

[105] M. Meeng and A. Knobbe. *Flexible enrichment with Cortana–software demo*, 2011.

[106] M. Boley, M. Mampaey, B. Kang, P. Tokmakov, and S. Wrobel. *One click mining: interactive local pattern discovery through implicit preference and performance learning*. In IDEA '13 Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, pages 27–35, 2013.

[107] F. Lemmerich and M. Becker. *pysubgroup: Easy-to-use subgroup discovery in python*. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 658–662, 2018.

[108] A. L. Traud, P. J. Mucha, and M. A. Porter. *Social structure of Facebook networks*. Physica A: Statistical Mechanics and its Applications, 391(16):4165 – 4180, 2012.

[109] I. Cantador, P. Brusilovsky, and T. Kuflik. *2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011)*. In Proceedings of the 5th ACM conference on Recommender systems, 2011.

[110] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. *ArnetMiner: Extraction and Mining of Academic Social Networks*. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 990–998, 2008.

[111] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-j. P. Hsu, and K. Wang. *An overview of microsoft academic service (mas) and applications*. In Proceedings of the 24th International Conference on World Wide Web, pages 243–246, 2015.

[112] X. Chen, B. Kang, J. Lijffijt, and T. De Bie. *ALPINE: Active Link Prediction using Network Embedding*. arXiv e-prints, page arXiv:2002.01227, 2020. arXiv:2002.01227.

[113] L. C. Freeman. *Segregation in Social Networks*. Sociological Methods & Research, 6(4):411–429, 1978.

[114] M. E. J. Newman. *Modularity and community structure in networks*. Proceedings of the National Academy of Sciences, 103(23):8577–8582, 2006.

[115] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. *Extending the definition of modularity to directed graphs with overlapping communities*. Journal of Statistical Mechanics: Theory and Experiment, 2009(03):P03024, 2009.

[116] J. Yang and J. Leskovec. *Defining and Evaluating Network Communities Based on Ground-truth*. Knowledge and Information Systems, 42(1):181–213, 2015.

5 Conclusions

In this chapter, we conclude this thesis in a high-level view. Specific conclusions and future directions for each main contribution are provided at the end of Chapter 3 and Chapter 4.

5.1 General conclusions

This thesis tackles the problem of mining subjectively interesting patterns in two rich data types: time series and graphs. More specifically, we proposed novel pattern syntaxes, interestingness measures, and associated algorithms for mining motifs in time series, and for mining local and global rules implying subgraphs with surprising densities in graphs.

In general, what differentiates this thesis from prior work of mining rich data is its adopting of a *subjective interestingness* perspective—always keeping in mind that the mined patterns ultimately serve the interests of the user (e.g., for improving his or her understanding of the data or for benefiting the subsequent decision making). This perspective not only drove us to design the interestingness measure a subjective one in mining both time series and graphs, but also led us to concern a series of user-centric *what-if* questions when approaching the graph mining—*what if the user cannot understand the resulting patterns?* (explainability), *what if the user wishes to know some information that cannot fit in current pattern syntax* (generality), *what if there are too many similar ones in the output pattern set to get the user bored?* (non-redundancy). Attempts to address all these concerns fi-

nally made this research work a versatile one, enabling to tackle several well-posed tasks simultaneously including link rule discovery, dense (or sparse) subgraph detection, along with graph summarization. Thanks to the flexibility of De Bie's FORSIED (Formalizing Subjective Interestingness in Exploratory Data Mining) framework [1, 2], we generated solutions for all of these issues by building upon that.

Perspectives. Before embarking on limitations of the work reported in this thesis, let us reflect on what we think this thesis means. We can view this research work as instantiations of FORSIED framework on two interesting problems (i.e., discovering motifs in time series, as well as mining local and global subgraph patterns in attributed networks). Granted, our contribution is just a small part of the pattern mining research that are already (or will be) led by building upon FORSIED. By presenting our version of what FORSIED can achieve, we personally believe, this thesis also more or less reveals how the pattern mining research can hugely benefit from a theoretical framework of data mining like FORSIED (even though data mining is more an applied area). As has been pointed out in earlier time (in year 2000) by Mannilia in answering the question why look for a theoretical framework of data mining, *a theory in computer science can transform an area from hodgepodge of unconnected methods to an interesting and understandable whole, and at the same time enable an area of industry* [3]. Here, he gave a clear example (i.e., a relational model driving the development of the area of relational database) to support this statement. In this sense, we hope, this thesis would inspire the further development of FORSIED, or other (existing or incoming) theoretical data mining frameworks.

Moreover, for industrial applications that rely on mining informative patterns from data and want to include users in the loop, this work can benefit them as being a preliminary step towards ends-to-ends tools.

5.2 Future directions

Among the realm of rich data mining techniques, our thesis is situated at those based on developing building blocks (i.e., pattern syntaxes, interestingness measures, mining algorithms) that are dedicated to rich data. Though several previous limitations have been addressed, many still remain and some new ones have emerged, especially in aspects of interestingness measures and mining algorithms. In what follows, we will discuss these limitations that suggest further avenues of this research.

Interestingness measures. We first look at the aspect of interestingness measures.

- *More practical prior knowledge.* In our subjective interestingness framework, the user's prior knowledge is expressed as constraints for the maximum entropy optimisation problem which leads to the background distribution, the model for the user's prior beliefs. Nevertheless, all priors we have considered are data-dependent, and they are quite different from what the user usually holds in reality. Now we elaborate on this.

  The priors our model incorporates are in the format of property-value pair which expresses the user expects that a certain property (e.g., the mean, the variance or the first order difference of data points in the whole time series, the individual vertex degrees, the overall graph density, or densities between a pair of subgroups for the graph) should be in a certain value. However, in practice, the knowledge that the user holds is often a rough sense, which differs inherently from a very specific property-value pair. For example, the user may think students in a same university who love playing tennis often know each other, and this is, however, not equivalent to assert that the user knows the connectivity density among this subgroup of students—which is what our model can incorporate. Hence, such data-dependent property-value pairs usually overfit the user's prior knowledge in practice.

  Though this overfitting would normally not be a hinderance for generating qualitative results, we have to admit there are some cases where it can make the subjective interestingness backfire. For example, consider a university social network where students with same hobbies do not often know each other, which contradicts with the prior knowledge of the user who believes they should do. Because our model interprets such prior knowledge as connectivity densities within subgroups of students with the same hobbies—note which is actually sparse from the data, the resulting patterns will be ones indicating other kinds of information, or ones still related to a hobby but an 'outlier' such that students sharing this hobby are often friends. Clearly in this case, truly interesting patterns are identified as uninteresting and are submerged. Therefore, in the future, it would be useful to make our Subjective Interestingness (SI) measure able to circumvent this backfiring case, or tailor priors into more practical formats.

- *Subjective Description Length (DL).* The DL, one essential component of our Subjective Interestingness (SI) that quantifies the descriptional complexity of the pattern or the cost for a user to assimilate it, is not subjective in its current form (the source of the subjectiveness of our SI is merely the other component: the Information content (IC)). Consider parameters in the DL that represent how much the user prefers patterns in a more succinct form. We have argued in chapter 4 that these parameters should be determined from aspects of human cognition instead of statistical model selection. Nev-

ertheless, they were specified only based on an experimental testing. Clearly, different users own different cognitive capability or may prefer succinct patterns to different degrees, and a merely experimental testing cannot represent human cognition well. The variation among users should thus be taken into account through a theoretically rigorous study like the way IC is formalized.

- *More rigorous evaluation.* In our graph mining work, we provided experimental evaluation which demonstrates our SI measure is subjective and optimizing it can avoid redundancy between iteratively mined patterns. Nevertheless, the evaluation of the interestingness of the resulting patterns—this is to investigate another aspect: whether optimizing our SI can lead to patterns interesting to the user—is performed through a series of case studies (for both time series and graph mining work) with an imagined user and some imagined prior beliefs in mind. Though we consciously chose datasets with straightforward domain knowledge (e.g., social networks, citation networks) so that the quality of the resulting patterns can be easily justified, a more rigorous and convincing evaluation should be done through real user studies, as subjective interestingness depends on the user.

Mining algorithms. All our proposed algorithms are heuristics based on strategies such as greedy search and beam search. Though they produced qualitative results while helped to maintain the tractability, it would be useful for the future work to discover a heuristic strategy with theoretical guarantee for the quality of results, or develop algorithms that are anytime (i.e., it can be stopped at any point of time to supply patterns whose quality gradually improves over time). In regard with pursuing the anytime feature, the use of Monte Carlo Tree Search (MCTS) for pattern mining proposed by Bosc et al. [4] appears promising to be alternatives to our mining algorithms.

Pattern syntaxes. Now we give future directions with regard to the pattern syntax. Pattern mining is more for the case where the user has a clear sense about what format of information is valuable for him or her—this is captured by pattern syntax. We believe a general pattern syntax is more useful, in a sense it agrees with the user's expected format, but also poses least assumptions or constraints designed for the ease of model feasibility or computability. For example, a motif of length l proposed in the time series work is defined as a set of subsequences that are similar and of the length l. There is much room to expand this pattern syntax such that, e.g., enabling it to involve subsequences that are with similar shape in general but of slightly different lengths, or to be multivariate. Achieving such versatility necessitates ingenuity in interestingness measures and algorithms, but also in where everything begins—the pattern syntax.

Beyond users and data. Lastly, we want to talk about a concern for current data mining techniques. When we assert *user is king* in this thesis, a key argument is that the ultimate goal of data mining is to provide insights that can either improve the user's understanding about the data or boost his or her performance on a downstream task. Nevertheless, if contemplating the goal of data mining harder, we may find that the user is still the king, but data mining essentially is not about providing the user with valuable information in the *data*, but rather, in the *reality*. A problem is data, this digital format giant thing, cannot represent (even a piece of) our inherently complex reality without any loss of information. No matter how big the data is, it is almost never complete. Worse of all, data can easily be distorted, and is always biased. Those biases come from the source of data, the collection method, the person who performs the collection, the availability of objects being gathered as part of data, and so forth. Unfortunately, there appears no magic to make the data *unbiased* in the present and in the foreseeable future. Though much progress has been made by today's data mining research community in grappling with the biases, most of them are effective on reducing the type of biases that we can immediately see (e.g., discrimination on gender, race, or other sensitive traits). For some kind of unknown biases that are secretly happening, we still seem at a loss about what to do. Therefore, when we are too eager to sharpen our tools to mine gorgeous patterns from data, perhaps, we should turn around to also mine *what is not in the data*.

For making data mining closer to reality mining, in our beliefs, data mining needs to become more multidisciplinary. Today's data mining is multidisciplinary in a sense it involves different fields such as machine learning, statistics, database technology, expert systems and data visualization. In the future, data mining may need to span further to incorporate perspectives from a broad cross-section of humanity—people who take different roles in the society and hold different points of views such as social scientist, ethicist, criminologists, politicians, different end-users, and so on. Let us take a humble attitude and understand the world together!

References

[1] T. De Bie. *Subjective Interestingness in Exploratory Data Mining*. In Proceedings of the 12th International Symposium on Advances in Intelligent Data Analysis, pages 19–31, 2013.

[2] T. De Bie. *An Information Theoretic Framework for Data Mining*. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 564–572, 2011.

[3] H. Mannila. *Theoretical frameworks for data mining*. ACM SIGKDD Explorations Newsletter, 1(2):30–32, 2000.

[4] G. Bosc, J.-F. Boulicaut, C. Raïssi, and M. Kaytoue. *Anytime discovery of a diverse set of patterns with monte carlo tree search*. Data Mining and Knowledge Discovery, 32(3):604–650, 2018.