# The Data Problem in Data Mining

Albrecht Zimmermann
LIRIS, INSA Lyon, France
albrecht.zimmermann@insa-lyon.fr

## ABSTRACT

Computer science is essentially an *applied* or *engineering* science, creating tools. In Data Mining, those tools are supposed to help humans understand large amounts of data. In this position paper, I argue that for all the progress that has been made in Data Mining, in particular Pattern Mining, we are lacking insight into three key aspects: 1) How pattern mining algorithms perform quantitatively, 2) How to choose parameter settings, and 3) How to relate found patterns to the processes that generated the data. I illustrate the issue by surveying existing work in light of these concerns and pointing to the (relatively few) papers that *have* attempted to fill in the gaps. I argue further that progress regarding those questions is held back by a lack of data with varying, controlled properties, and that this lack is unlikely to be remedied by the ever increasing collection of real-life data. Instead, I am convinced that we will need to make a science of digital data generation, and use it to develop guidance to data practitioners.

## 1. INTRODUCTION

Computer science is basically an *applied* or *engineering* science. By this, I do not mean that all work done in our field does or should happen only in relation with a concretely defined real-life application. But rather that we use the results of other disciplines, be they mathematics, physics, or others, and develop what should be understood as *tools*, devices and algorithms that make it easier for humans to perform certain tasks.

In Data Mining, those tools come mainly in two forms: 1) *supervised* methods that learn from labeled data how to *predict* labels for unseen data or how to *characterize* predefined subsets, and 2) *unsupervised* methods. A second dimension along which to characterize them has to do with the scope of their results: a) they apply either to (almost) the entire data set – they are *global* in nature, such as classification models or clusterings, or b) being *local*, they refer only to a (non-predefined) subset of the data.

The setting where the lack of supervision and local results intersect, i.e. 2b), it often referred to as "Pattern Mining" (PM), which is the term I will use hereafter. It holds a great promise: given large amounts of data and little or no supervision, PM can find interesting, hitherto undiscovered, relationships – *patterns* – in the data. Those patterns can in turn be exploited in the domains whence the data were generated to, for instance, further research, improve logistics, or increase sales. To see why this promise is so great, one only has to consider that supervised modeling already knows one side of relationship it seeks to establish. Supervised modeling seeks explanations, unsupervised modeling hypotheses.

Pattern Mining has been an active research field at least since the publication of the seminal paper introducing the Apriori algorithm [3] for frequent itemset mining (FIM). The twenty years since have brought an ever-widening scope of the field, extending the original itemset/association rule setting to semi-structured and structured data, the transactional setting to single-instance settings such as episode and network mining, and the application of PM techniques to a variety of different fields. This widening of scope also led to a plethora of techniques, published in a number of journals and conferences.

As I will argue in detail, however, the increase in number of topics and algorithms has not been paralleled by an equal increase in understanding the strengths and in particular limitations of developed techniques, or by guidelines for their employment. And creating tools (or rather in many cases blueprints) is not enough: to fulfill PM's promise and make the most of the developed techniques, it is necessary to give potential users an idea how to actually employ those tools.

In particular, there are three large gaps in our understanding of pattern mining:

1. **We do not know how most pattern mining algorithms actually perform quantitatively!** Pattern mining algorithms are *rarely*, if ever, evaluated on additional data after they have been published. Additionally, they are rarely extensively compared against each other.
2. **We do not know how to choose good parameter settings for pattern mining algorithms!** The relationships between parameter settings and running times/memory consumption are not well-established, let alone the relationship to the interestingness of patterns.
3. **We do not know how mined patterns relate to the generative processes underlying the data!** The current working interpretations of "interesting" recur to (objective or subjective) unexpectedness, or summarization/compression of the data. This undermines the interpretation of patterns and the applicability of derived knowledge to the real-life setting whence they originated.

In the following three sections, I will illustrate the gaps in our knowledge in detail. In particular, I will pay attention to the work that *did* attempt to fill those gaps, and proposed ways of researching these issues, but also why it falls short.
These problems are not limited to PM – evaluations, comparisons, and the exploration of good parameter settings also often fall short of best practices for supervised settings, and techniques that result in global models. But the problem is particularly pronounced in PM, mainly because the local results require the evaluation of individual patterns and because quality criteria are harder to define.
As I will argue, an important contributing factor is the lack of data in general, and of data with controlled, diverse characteristics and known ground truth in particular:

- Lack of data necessarily limits experimental evaluations and if no new data is added to the portfolio, no reevaluations can be performed.
- Lack of diverse characteristics means that we have seen algorithmic behavior only over a relatively narrow range of settings, and that we lack understanding of how small changes in such characteristics affects behavior.
- The lack of data of which the ground truth is know, finally, is the main factor that makes it so hard to fill in the third gap: supervised data contains a certain amount of ground truth in the labels, while this is missing for unsupervised settings, and global approaches can be evaluated by assessing the data partition, for instance.

There is a potential solution to this problem – artificial data generation – but as I will show, the track record of the PM community w.r.t. data generation is rather weak so far. This has to be remedied if we want to do more than propose tools that might or might not work. While developing generators that lead to data with differing characteristics could turn out to be relatively easy, knowing what kind of generative processes can be expected to occur in real-life data will be more challenging and such knowledge will often not be found with data miners but with real-world practitioners. Hence, I argue that we need to add a deeper understanding of data – "data science" so-to-say – to the PM research portfolio, a task that will require the collaboration with researchers and practitioners of other fields that currently is too often more statement than fact.

## 2. (RE)EVALUATION/COMPARISON

The first glaring problem has to do with the lack of extensive evaluation of data mining algorithms, whether in the original papers, on additional data, or in comparisons with other techniques from the field. As a result thereof, we do not have a clear idea how different algorithms will behave on certain data. We *do* know some things: dense data sets will result in more patterns and more computational effort, strict formal concept analysis on real-life data will probably not result in good output compression. But apart from that, the body of knowledge is weak.
Often, the paper that introduces a new algorithm contains the most comprehensive published experimental evaluation of that algorithm in terms of running times, memory consumption etc. Even those initial evaluations are often not very extensive, though. In the following, I will give an overview of this phenomenon in different subfields.
The seminal FIM paper used an artificial data generator to evaluate their approach on more than 45 data sets, generated by varying parameters. Notably, all those data shared a common characteristic – they were sparse. A similarly systematic evaluation can still be found in [61], yet most other early work [22; 64; 44] used far fewer data sets. FIM papers since have followed this trend with few exceptions.
Sequence mining was first formalized and proposed at roughly the same time as FIM by the same authors and in [51] twelve data sets are used for evaluation, nine of which artificial, with a similarly declining trend for follow-up work [62; 21; 45]. Sequence mining was transferred from the transactional setting to finding recurrent patterns – *episodes* – in single large data sequences in [37] and algorithms from this subfield have typically only been evaluated on a few data sets.
With a few exceptions, papers on the generalizations of sequences – trees (introduced in [63]) and graphs [25; 30; 26; 60; 24] – have followed the same trajectory, with two of my own papers [11; 68] among the worst offenders in this regard.
Graph mining has also been extended to the "single large" setting and different papers [17; 31; 28] have used less than twenty data sets each from various sources.
Rarely have those algorithms been *reevaluated* on additional data beyond that used in the original papers, apart from some comparisons in the papers that proposed improvements. Even in the latter case, transitivity has often been assumed – if algorithm B has been reported to perform better than algorithm A, comparing to B is considered enough. But obviously, this only holds for the data on which that evaluation has been performed, either locking future evaluations into the same restricted data or leading to unjustified generalizations about algorithmic behavior.
The paper that most incisively demonstrated this problem is arguably the one by Zheng *et al.* [65]. As described above, the data on which Apriori was evaluated were artificially generated and follow-up techniques mainly used subsets of those data. When comparing the artificial data to real-life data at their disposal, Zheng *et al.* noticed that the latter had different characteristics. An experimental comparison of the follow-up algorithms showed that claimed improvements in the literature did not transfer to the real-life data – the improvements had been an artifact of the data generation process.
Unfortunately, that paper has remained one of a handful of exceptions. With the exception of the Frequent Itemset Mining Implementations (FIMI) workshops [19; 5], little additional work has been done on FIM. While FIMI was undoubtedly important, there were notable short-comings: the workshop took place only twice, the focus was more on the practical aspects of implementing abstract algorithms than on an assessment of the algorithms themselves, and was limited to the, relatively small, collection of data sets available.
In graph mining, [59] compared four depth-first search graph miners on new data, notably reporting results that contradict those in [24]. The authors of [40] generated and manipulated data, and most notably find that there is no strong relation between run times and the efficiency of the graph codes, as had been claimed in the literature.
Episode mining is an area in which evaluations and compar-

isons of algorithms are particularly rare. I am not aware of any other work except my own [67], in which I used a data generator to generate data having a range of characteristics and reported results that indicate temporal constraints have more impact than pattern semantics, contrary to claims in the literature.

Those papers show how important it is to use data with new characteristics, and to pay attention to questions of implementations and hardware when assessing the usefulness of algorithmic approaches. Yet, compared to the body of work describing algorithmic solutions, the body of work comprehensively evaluating those solutions is rather small.

# 3. IDENTIFYING PARAMETER SETTINGS

A second problem, somewhat related to the first, is that, for the majority of techniques, it is unclear how parameter settings should be chosen. As a rule of thumb, more lenient parameter settings will lead to longer running times but not even this relationship has been established concretely. At first sight, the situation w.r.t. this problem is better but this impression is deceiving.

Several papers established a relationship between the distribution of mined frequent, closed, and maximal itemsets and algorithmic running times [64; 49; 20; 14]. Apart from the fact that such research does not exist for structured data and patterns, those approaches are faced with the obvious problem that extensive mining, at potentially extensive running times, is needed before the relationship can be established.

An improvement consists of estimating support distributions and running times based on partial mining results, or sampled patterns, as been the approach of [34; 43; 16; 10; 9]. Those studies only traverse half the distance though – in predicting running times and output sizes – even though [9] proposes setting a frequency threshold high enough to avoid the exponential explosion of the output set. Whether a threshold setting that allows the operation to finish in a reasonable amount of time will lead to *interesting* patterns, is largely unexplored.

A notable exception concerned with mining sequences under regular expression constraints can be found in [6]. By sampling patterns fulfilling data-independent constraints under assumptions about the symbol distribution, they derive a model of background noise, and identify thresholds expected to lead to interesting results. A similar idea can be found in [38], which uses sampling and regression to arrive at *pattern frequency spectra* for FIM. By comparing analytical expressions for spectra of random data to the actually derived spectra, they also identify deviating regions, proposing to explore those. Those two papers come closest to actually giving guidance for parameter selection but are limited to frequency thresholds. Yet, the *blue print* for building up knowledge about the interplay of data characteristics and parameter settings – for instance for storage in experiment databases [54] – is there, held back by the relatively limited supply of available data sets.

Apart from the fact that it would be attractive to fix parameter settings before an expensive mining operation, this is an instance in which the unsupervised nature of pattern mining makes the task harder than for supervised settings. In the latter, validation sets can be used to assess the quality of resulting models, and parameters adjusted accordingly. There is work based on related ideas, which derive *null models* directly from the data, already found patterns, or user knowledge and use statistical testing to remove all those patterns that are not unexpected w.r.t. the null hypothesis [7; 18; 39], or uses multiple comparisons correction and hold-out sets [57]. While such approaches promise to remove all statistically unsurprising patterns, unexpected patterns are not necessarily useful ones.

# 4. MATCHING PATTERNS TO REALITY

This last remark hints at an important issue in PM: which patterns to consider "interesting". The field has moved on from the FIM view of (relatively) frequent and incuding (relatively) strong implications. Yet what has taken its place are (objectively or subjectively) surprisingness (see above), or effective summarization/compression (e.g. [56]). Such patterns are undoubtedly interesting and useful but they leave us with the third and in my opinion most important gap in our knowledge: it is currently mostly unclear how the patterns that are being mined by PM techniques relate to the patterns actually occurring in the data.

To be clear about this: we know what types of patterns we have defined, and we have the algorithms to find them according to specific criteria (at least in the case of complete techniques). In FIM, for instance, if there is set of items that is often bought together, it will be found. Yet so will all of its subsets, and maybe intersections with other itemsets, and we are currently lacking the knowledge to decide which of these patterns are relevant. Hence, in many cases we do not know whether we capture meaningful relationships.

Furthermore, even if we could identify the actual patterns, we would not know how those relate to the processes that generated the data in the first place. To continue with the FIM example, papers will often give the motivation behind performing such mining by stating that supermarkets can group items that are bought together close to each other, to motivate those customers who did not buy them together to do so. An alternative proposal states supermarkets should group such items far apart to motivate customers to traverse the entire store space and potentially make additional purchases.

These strategies implicitly assume two different types of customer behavior, though: in the latter case, the co-purchases are systematic and can be leveraged to generate additional business. In the former, the co-purchases are somewhat opportunistic, which also means that using the first strategy could lead to loss of business. Maybe the two types of behavior actually lead to different expressions of the pattern in the data. And maybe the layout of the supermarket at the time of purchase has an effect on this expression, for instance because it enables the opportunistic behavior. But since there exist no studies relating mined itemsets to the shopping behavior itself, i.e. the generative process of the data, it is unclear, twenty years after FIM was proposed, which of the two assumptions holds.

This has grave implications. If it is unclear how patterns relate to the underlying processes, it is also unclear how to exploit patterns in the domains that the data originated from. Given that this is the rationale of data mining, the current state of the art in PM research is therefore failing the field's main purpose: supporting real-world decision making.

Again, there exist studies that have attempted to fill this gap in a more or less systematic manner [32; 53; 52; 36; 35;

66; 48; 58; 67]. The typical approach consists of embedding explicitly defined patterns in the data (with or without noise effects), and comparing mined patterns to them to understand the relationships. Of particular interest are experiments in which the data pattern generation takes a different form from the pattern definition, such as in [35] in which Bayes' Nets were used to generate itemsets. Yet, again, there are not many of them, and their focus has often been only on a single technique.

The problem of interpretation exists for supervised and/or global approaches as well, but arguably to a lesser degree. If the goal is prediction, high accuracy is an indicator of a good result – after all, there are "black box" classification techniques but no black box PM ones. Similarly, a clustering that exhibits high intra-cluster similarity and inter-cluster dissimilarity is probably a good one – incidentally an assessment that is related to good summarization. And even if the setting is supervised and local, as in subgroup discovery, patterns that correlate strongly with the subgroup can be expected to be meaningful.

# 5. THE DATA PROBLEM

The preceding sections should not be read as an indictment of all PM research. Quite contrary, many of the found solutions are ingenious and elegant, and the impressive tool kit that has been amassed should enable practitioners to address many real-world problems more effectively. But faced with data, a typical practitioner will not know which tools to choose, how to set the parameters without extensive trial-and-error, and what conclusion to draw from the resulting patterns – unless we fill in the gaps.

There are several factors that influence the described situation. Some of those are related to what kind of research is rewarded, which in turn relates to publication policies. Instead of discussing those, the interpretation of which is necessarily subjective, I want to draw attention to an objective factor that I have also pointed to in each section:

**Despite what one would expect given the name "Data Mining", what we lack is data!**

In reaction to the work of Zheng *et al.*, the data sets they introduced were added to the benchmark data sets for FIM and reliance on the data generator from [3] was reduced. Several other data sets were added over time and the collection is currently downloadable at the FIMI website.[1] Yet the totality of this collection comprises only twelve data sets. This is a far cry from the large amount of data sets available at the UCI repository for machine learning [8] (used for predictive learning, clustering, and subgroup discovery), the UCR collection of data for time series classification and clustering [29], or even the data sets available for multi-label learning.[2]

Sequence mining is mainly performed on a small number of biological data sets. Most of the real-life data used in episode mining papers are covered by non-disclosure agreements and have therefore never entered the public domain. There are handful of tree mining data sets, mainly based on click streams or website traversal. Graph mining also makes heavy use of a small number of molecular data sets, and network mining data sets have ranged from graph encodings of UCI data to snapshots of social, citation, traffic, or biological networks.

Unless the current collections cover by pure accident all characteristics that can be encountered in the real world, even best-practice evaluations based on them will not give a complete picture of algorithms' strengths and weaknesses, making it difficult to address the first and second problem. Furthermore, even *if* we had large amounts of real-life data at our disposal, in most cases we would *not* know the ground truth of such data and therefore could not address the problem laid out in Section 4. After all, real-life data in supervised settings "only" need a label to assess whether relationships are relevant but evaluating patterns in a local unsupervised setting needs a much deeper understanding of the data. And even if we had data available of which we knew the ground truth, we would lack the knowledge about the generative processes leading to this ground truth, as described above.

Luckily for computer scientists, there is an alternative to assembling ever increasing collections of real-life data, or rather a complement: artificial data generation. This is the solution that has been chosen in exploring phenomena in the SAT solving community [47], for instance, or for evaluating another unsupervised setting, clustering [46]. The idea is also running like a thread through much of the work I have reviewed so far, whether it is artificial data used for systematically exploring the effects of data characteristics, for identifying where data deviates from random backgrounds, or for matching patterns to generating processes.

## 5.1 Data Generation in PM

The problem is, however, that the story of data generation in PM so far is arguably one of failure. The data generator used in [3] was discredited by Zheng *et al.*. This has repercussions since the data generators used in sequence and graph (and arguably tree) mining papers base on similar considerations. Cooper *et al.* [13], attempting to fix a second problem of that generator, ignored the first one, and introduced new artifacts. The survey undertaken in [12] lists 22 generators for network data, all of which attempt to reproduce certain numerical properties of real-life data, such as degree distribution, or clustering coefficient. With the exception of a few that attempt to model particular types of networks, the authors found that none of the proposals gets it fully right. The reference is admittedly somewhat dated but other work published since [33; 41; 15; 42] comes to the same conclusion.

Generators leading to data resembling the one already available suffer from the fact that they do not solve the problem of the data bottle neck. Approaches such as [49; 50; 55] take the output of an FIM operation and generate databases that will result in similar output. Thus, they take the existence of data as a given, as do the approaches that create null models based on the data. Furthermore, the data generated by the former is expected to result in the same mix of relevant and irrelevant patterns as the old one, and the latter mask the underlying processes.

While artificial data generation enables us to create data with a wide range of characteristics, assess the effects of different kinds of noise on the ability to recover patterns, and to simulate different generative processes, we have not used this ability to fill in the gaps in our understanding. This will need to change, and I am convinced that to do so

[1] `http://fimi.ua.ac.be/data/` – accessed 08/21/2014

[2] `http://en.sourceforge.jp/projects/sfnet_mulan/releases/` – accessed 08/21/2014

we, or at least some of us, have to become *data scientists*.

## 6. DATA SCIENCE

When media and non-academics refer to data miners or data analysts, the term "data scientist" is often used. But what this term implies, in my opinion, is that such a person understands the data, and **we do not**. Part of this is by design – as I wrote in the beginning, the promise of pattern mining is to find interesting patterns in a largely unsupervised manner. The naive interpretation of this promise, however, is similarly flawed as the claim that in the age of "Big Data", discovering correlation replaces understanding causation [4].[3]

To fill in the gaps in our understanding as PM researchers, data science needs to be added to our expertise. We need to develop data generators that produce varied characteristics in a controlled manner, to enable extensive experiments. Those data generators need to use generative processes to which we can map patterns back, so we start to understand how certain processes manifest in the output of the tools we develop. Concretely, this means exploring different distributions (and mixtures thereof) governing the data generation, instead of fixing a single one. It means adding varying degrees of noise to the data. And it means using generative processes that are different from the sought patterns – Bayes' Nets for itemset data, or interacting agents for network data, for instance. In fact, there exists already a tool that makes some of this possible, the KNIME data generator [2], which is however neither used widely nor systematically so far.

Once we have access to such generators, we can follow the approach of existing studies to fill in some of the gaps that currently exist. This means, for instance, evaluating and comparing algorithms over data of different density, pattern length, alphabet size, etc. It means establishing what proportion of individual patterns can be recovered under the effects of noise. It means assessing whether highly ranked itemsets represent fragments of embedded patterns, or represent subgraphs of Bayes' Nets. It also means understanding whether data that are generated by the same processes with the same parameters actually have the same characteristics, and whether they give rise to similar result sets, i.e. whether it is appropriate to transfer insights derived on one data set to another that "looks" similar. In other words, it should allow us to develop better ways of comparing data sets.

This is obviously not an exhaustive enumeration and it will take the creativity and effort of the community to get the field to that stage. But even that can only be a step on the way: we can learn how the patterns we mine relate to the patterns in the data, and in turn to the processes that generated them. How itemsets relate to agents that "shop" according to certain "behavior", for instance.

The knowledge about real-life behavior cannot come from inside our community, however. Instead, it will be found in physics [1], in engineering [27], in the social and life sciences. Once *we* have understood which method to use, how to set parameters, and how to select relevant patterns and interpret them, based on the data available, we can approach practitioners from those fields. Using their knowledge, we can generate data that are real life-like, and troubleshoot generators and evaluation methods. In all probability, some of the assumptions from those fields will turn out to be wrong but probably not more wrong than the assumptions we ourselves have made in generating data so far. And if, building on such assumptions, we find them to be wrong (or at least questionable), and feed this information back into the fields whence they originated, even better.

I have not come here to bury PM but to praise it. I am convinced that the potential of the tools that the community has developed over the last two decades is tremendous. I am, however, challenging the community to develop guidance for how to use those tools. Working closely with practitioners and giving them hands-on guidance, the modus operandi of many application papers, is a worthy endeavour but it is also time-consuming and allows for little generalization. We have to solve the data problem in data mining and we have to do it in a better-founded way than by trying to acquire additional real-life data sets. We need to make a science out of generating digital data.

[3]A claim that incidentally experiences tremendous pushback.

## Acknowledgments

I am grateful to Matthijs van Leeuwen, Arno Siebes, and Jilles Vreeken for reviewing preliminary versions of this article, giving feedback, sharpening the questions, and discussing possible answers. The author was supported by the FP7-PEOPLE-2013-IAPP project GRAISearch (Grant Agreement Number 612334) at the time of writing.

## 7. REFERENCES

[1] Corsika - an air shower simulation program,. https://web.ikp.kit.edu/corsika/.

[2] I. Adä and M. R. Berthold. The new iris data: modular data generators. In B. Rao, B. Krishnapuram, A. Tomkins, and Q. Yang, editors, *KDD*, pages 413–422. ACM, 2010.

[3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Databases*, pages 487–499, Santiago de Chile, Chile, Sept. 1994. Morgan Kaufmann.

[4] C. Anderson. The end of theory: The data deluge makes the scientific method obsolete. `http://archive.wired.com/science/discoveries/magazine/16-07/pb_theory`. Accessed 08/21/2014.

[5] R. J. Bayardo Jr., B. Goethals, and M. J. Zaki, editors. *FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004*, 2004.

[6] J. Besson, C. Rigotti, I. Mitasiunaite, and J.-F. Boulicaut. Parameter tuning for differential mining of string patterns. In *ICDM Workshops*, pages 77–86. IEEE Computer Society, 2008.

[7] T. D. Bie. Maximum entropy models and subjective interestingness: an application to tiles in binary databases. *Data Min. Knowl. Discov.*, 23(3):407–446, 2011.

[8] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.

[9] M. Boley, T. Gärtner, and H. Grosskreutz. Formal concept sampling for counting and threshold-free local pattern mining. In *SDM*, pages 177–188. SIAM, 2010.

[10] M. Boley and H. Grosskreutz. A randomized approach for approximating the number of frequent sets. In *ICDM*, pages 43–52. IEEE Computer Society, 2008.

[11] B. Bringmann and A. Zimmermann. Tree$^2$ - Decision trees for tree structured data. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, editors, *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 46–58. Springer, 2005.

[12] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1), 2006.

[13] C. Cooper and M. Zito. Realistic synthetic data for testing association rule mining algorithms for market basket databases. In J. N. Kok, J. Koronacki, R. L. de Mántaras, S. Matwin, D. Mladenic, and A. Skowron, editors, *PKDD*, volume 4702 of *Lecture Notes in Computer Science*, pages 398–405. Springer, 2007.

[14] F. Flouvat, F. D. Marchi, and J.-M. Petit. A new classification of datasets for frequent itemsets. *J. Intell. Inf. Syst.*, 34(1):1–19, 2010.

[15] A. Freno, M. Keller, and M. Tommasi. Fiedler random fields: A large-scale spectral approach to statistical network modeling. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *NIPS*, pages 1871–1879, 2012.

[16] F. Geerts, B. Goethals, and J. V. den Bussche. Tight upper bounds on the number of candidate patterns. *ACM Trans. Database Syst.*, 30(2):333–363, 2005.

[17] S. Ghazizadeh and S. S. Chawathe. Seus: Structure extraction using summaries. In S. Lange, K. Satoh, and C. H. Smith, editors, *Discovery Science*, volume 2534 of *Lecture Notes in Computer Science*, pages 71–85. Springer, 2002.

[18] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. *TKDD*, 1(3), 2007.

[19] B. Goethals and M. J. Zaki, editors. *FIMI '03, Frequent Itemset Mining Implementations, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA*, volume 90 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[20] K. Gouda and M. J. Zaki. Genmax: An efficient algorithm for mining maximal frequent itemsets. *Data Min. Knowl. Discov.*, 11(3):223–242, 2005.

[21] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu. Freespan: frequent pattern-projected sequential pattern mining. In R. Ramakrishnan, S. J. Stolfo, R. J. Bayardo, and I. Parsa, editors, *KDD*, pages 355–359. ACM, 2000.

[22] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. F. Naughton, and P. A. Bernstein, editors, *SIGMOD Conference*, pages 1–12. ACM, 2000.

[23] J. Han, B. W. Wah, V. Raghavan, X. Wu, and R. Rastogi, editors. *Fifth IEEE International Conference on Data Mining*, Houston, Texas, USA, Nov. 2005. IEEE.

[24] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. In *ICDM*, pages 549–552. IEEE Computer Society, 2003.

[25] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In D. A. Zighed, H. J. Komorowski, and J. M. Zytkow, editors, *PKDD*, volume 1910 of *Lecture Notes in Computer Science*, pages 13–23. Springer, 2000.

[26] A. Inokuchi, T. Washio, K. Nishimura, and H. Motoda. A fast algorithm for mining frequent connected subgraphs. Technical report, IBM Research, 2002.

[27] E. F. V. J. J. Down. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3):245–255, March 1993.

[28] U. Kang, C. E. Tsourakakis, and C. Faloutsos. Pegasus: A peta-scale graph mining system. In W. Wang, H. Kargupta, S. Ranka, P. S. Yu, and X. Wu, editors, *ICDM*, pages 229–238. IEEE Computer Society, 2009.

[29] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR time series classification/clustering homepage, 2011.

[30] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In N. Cercone, T. Y. Lin, and X. Wu, editors, *ICDM*, pages 313–320. IEEE Computer Society, 2001.

[31] M. Kuramochi and G. Karypis. Finding frequent patterns in a large sparse graph$^*$. *Data Min. Knowl. Discov.*, 11(3):243–271, 2005.

[32] S. Laxman, P. S. Sastry, and K. P. Unnikrishnan. Discovering frequent episodes and learning hidden markov models: A formal connection. *IEEE Trans. Knowl. Data Eng.*, 17(11):1505–1517, 2005.

[33] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In J. Huai, R. Chen, H.-W. Hon, Y. Liu, W.-Y. Ma, A. Tomkins, and X. Zhang, editors, *WWW*, pages 695–704. ACM, 2008.

[34] L. Lhote, F. Rioult, and A. Soulet. Average number of frequent (closed) patterns in bernouilli and markovian databases. In Han et al. [23], pages 713–716.

[35] M. Mampaey and J. Vreeken. Summarizing categorical data by clustering attributes. *Data Min. Knowl. Discov.*, 26(1):130–173, 2013.

[36] M. Mampaey, J. Vreeken, and N. Tatti. Summarizing data succinctly with the most informative itemsets. *TKDD*, 6(4):16, 2012.

[37] H. Mannila and H. Toivonen. Discovering frequent episodes in sequences. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pages 210–215. AAAI Press, 1995.

[38] A. U. Matthijs van Leeuwen. Fast estimation of the pattern frequency spectrum.

[39] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[40] S. Nijssen and J. Kok. Frequent subgraph miners: runtimes don't say everything. In T. Gärtner, G. Garriga, and T. Meinl, editors, *Proceedings of the Workshop on Mining and Learning with Graphs,*, pages 173–180, 2006.

[41] G. K. Orman, V. Labatut, and H. Cherifi. Qualitative comparison of community detection algorithms. In H. Cherifi, J. M. Zain, and E. El-Qawasmeh, editors, *DICTAP (2)*, volume 167 of *Communications in Computer and Information Science*, pages 265–279. Springer, 2011.

[42] G. K. Orman, V. Labatut, and H. Cherifi. Towards realistic artificial benchmark for community detection algorithms evaluation. *IJWBC*, 9(3):349–370, 2013.

[43] P. Palmerini, S. Orlando, and R. Perego. Statistical properties of transactional databases. In H. Haddad, A. Omicini, R. L. Wainwright, and L. M. Liebrock, editors, *SAC*, pages 515–519. ACM, 2004.

[44] 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*, pages 21–30, 2000.

[45] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Prefixspan: Mining sequential patterns by prefix-projected growth. In D. Georgakopoulos and A. Buchmann, editors, *ICDE*, pages 215–224. IEEE Computer Society, 2001.

[46] Y. Pei and O. Zaïane. A synthetic data generator for clustering and outlier analysis. Technical report, 2006.

[47] D. M. Pennock and Q. F. Stout. Exploiting a theory of phase transitions in three-satisfiability problems. In *AAAI/IAAI, Vol. 1*, pages 253–258, 1996.

[48] B. A. Prakash, J. Vreeken, and C. Faloutsos. Efficiently spotting the starting points of an epidemic in a large graph. *Knowl. Inf. Syst.*, 38(1):35–59, 2014.

[49] G. Ramesh, W. Maniatty, and M. J. Zaki. Feasible itemset distributions in data mining: theory and application. In *PODS*, pages 284–295. ACM, 2003.

[50] G. Ramesh, M. J. Zaki, and W. Maniatty. Distribution-based synthetic database generation techniques for itemset mining. In *IDEAS*, pages 307–316. IEEE Computer Society, 2005.

[51] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In P. M. G. Apers, M. Bouzeghoub, and G. Gardarin, editors, *EDBT*, volume 1057 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 1996.

[52] N. Tatti and J. Vreeken. Discovering descriptive tile trees - by mining optimal geometric subtiles. In P. A. Flach, T. D. Bie, and N. Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part I*, volume 7523 of *Lecture Notes in Computer Science*, pages 9–24. Springer, 2012.

[53] N. Tatti and J. Vreeken. The long and the short of it: summarising event sequences with serial episodes. In Q. Yang, D. Agarwal, and J. Pei, editors, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 462–470. ACM, 2012.

[54] J. Vanschoren, H. Blockeel, B. Pfahringer, and G. Holmes. Experiment databases - A new way to share, organize and learn from experiments. *Machine Learning*, 87(2):127–158, 2012.

[55] J. Vreeken, M. van Leeuwen, and A. Siebes. Preserving privacy through data generation. In N. Ramakrishnan and O. Zaïane, editors, *ICDM*, pages 685–690. IEEE Computer Society, 2007.

[56] J. Vreeken, M. van Leeuwen, and A. Siebes. Krimp: mining itemsets that compress. *Data Min. Knowl. Discov.*, 23(1):169–214, 2011.

[57] G. I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.

[58] G. I. Webb and J. Vreeken. Efficient discovery of the most interesting associations. *Transactions on Knowledge Discovery from Data*, 8(3):15–1, 2014.

[59] M. Wörlein, T. Meinl, I. Fischer, and M. Philippsen. A quantitative comparison of the subgraph miners mofa, gspan, ffsm, and gaston. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, editors, *PKDD*, pages 392–403. Springer, 2005.

[60] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *ICDM*, pages 721–724. IEEE Computer Society, 2002.

[61] M. J. Zaki. Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.*, 12(3):372–390, 2000.

[62] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.

[63] M. J. Zaki. Efficiently mining frequent trees in a forest. In *KDD*, pages 71–80. ACM, 2002.

[64] M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In R. L. Grossman, J. Han, V. Kumar, H. Mannila, and R. Motwani, editors, *SDM*. SIAM, 2002.

[65] Z. Zheng, R. Kohavi, and L. Mason. Real world performance of association rule algorithms. In *KDD*, pages 401–406, 2001.

[66] A. Zimmermann. Objectively evaluating condensed representations and interestingness measures for frequent itemset mining. *Journal of Intelligent Information Systems*, pages 1–19, 2013.

[67] A. Zimmermann. Understanding episode mining techniques: Benchmarking on diverse, realistic, artificial data. *Intell. Data Anal.*, 18(5):761–791, 2014.

[68] A. Zimmermann and B. Bringmann. Ctc - correlating tree patterns for classification. In Han et al. [23], pages 833–836.