The Journal of Open Source Software

Decomposing Gaifman Structures

José Luis Balcázar [1¶], **Marie Ely Piceno** [2], **and Laura Rodríguez-Navas** [3]

1 Universitat Politècnica de Catalunya, Catalonia **2** Algorithia, Mexico City, Mexico **3** BETA Tech Center, Universitat de Vic - Universitat Central de Catalunya, Catalonia ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

We describe here our tool `degais` for computing and visualizing decompositions of Gaifman structures. These are a generalization of undirected graphs whose tree-like decomposition often helps visualizing specific properties of data.

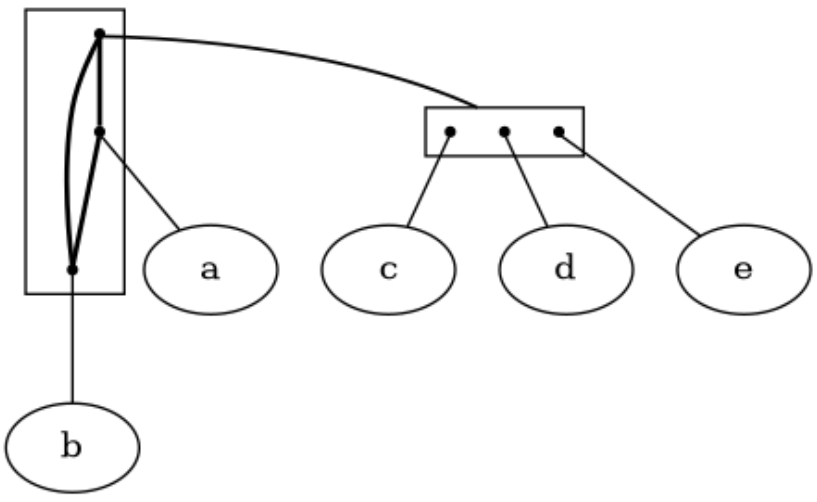
Gaifman structures are obtained from transactional data by generalizing the existing notion of a Gaifman graph. In (Balcázar et al., 2018) (alternative, open access version here) we proposed to employ these generalized Gaifman graphs as a method of visual data analysis and, specifically, as a way of displaying co-occurrence patterns in data. Such a process allows the user to understand better a given dataset, by showing visually the roles of each item in the data in terms of their co-occurrence patterns with other items.

Subsequently, we demonstrated in (Piceno et al., 2021) how the visualization of various sorts of Gaifman structures leads to better understanding of a fully anonymized dataset of patient diagnostics and conditions from a local hospital.

As a very simple example, assume that the dataset `e5.td` contains three tuples (or: transactions), namely,

```
a b c
a b d
a b e
```

Then, a call `degais e5` will produce the following diagram, indicating that items `c`, `d`, and `e` never co-occur, but all three do co-occur with both `a` and `b` which, also, do co-occur:



The Journal of Open Source Software

Statement of need

The software used in (Balcázar et al., 2018) and, also, the implementations employed for related documents (Piceno, 2020; Piceno et al., 2021; Rodríguez-Navas, 2017) were not really usable by other people. These predecessors of `degais`, closed-source in (Rodríguez-Navas, 2017) and several variants of open-source implementations for (Piceno, 2020; Piceno et al., 2021), get replaced now by `degais`, an implementation with which we hope to offer the possibility of using this data analysis tool to any interested person.

Model foundations

Gaifman graphs are mathematical structures introduced several decades ago as a means to study limitations of the expressivity of logical languages (Libkin, 2004) (also available here). Their basic notion is pretty simple: given a dataset consisting of observations, each observation being a set of *items* (e. g., attribute-value pairs or simply categorical values), for each transaction, and for each pair of different items x and y in that transaction, we ensure that the edge (x, y) is present in the graph, and that only such edges are: thus, items coincide with vertices in the graph, and they remain disconnected exactly when they don't appear jointly in any transaction.

Sometimes, we may be interested in keeping track of quantitative information that the standard Gaifman graph lacks. Alternative versions were introduced in (Balcázar et al., 2018):

- In thresholded graphs, the difference between having the edge or not, instead of being zero joint occurrences versus 1 or more, resorts to a threshold possibly different from 1: graph connections represent frequencies of co-occurrence higher than the threshold.
- In the simplest version of labeled Gaifman graphs, the edges are labeled by their multiplicities, that is, the number of tuples containing both of the vertices that they connect. This strategy, in practice, most often leads nowhere.
- In more evolved versions, edge labels are obtained from these same multiplicities via some sort of discretization process. In linear Gaifman graphs, labels correspond to frequencies of co-occurrence falling in intervals of some fixed width while in the exponential Gaifman graphs a log function is applied first, then the fixed-width intervals are applied.

We represent visually the labels with different *colors*.

The data analysis approach proposed in (Balcázar et al., 2018) consists in applying a decomposition process to the (possibly labeled) Gaifman graphs; in some cases the so-called modular graph decomposition suffices but, in general, one must resort to clans on 2-structures (Ehrenfeucht et al., 1999), explained next. This is a concept that generalizes graphs and, specifically, symmetric 2-structures generalize undirected graphs such as Gaifman graphs. We call Gaifman Structures the symmetric 2-structures obtained as the labeled extensions of Gaifman graphs enumerated above.

Then, the decomposition procedure is based on the so-called "clans". They extend the intuitive concept of when a vertex "sees in different ways" (or: "distinguishes") two other vertices. We say that an item x distinguishes two other items if the edges that connect x with these two items have different labels. Then, a clan is a set of items such that any two members of the clan cannot be distinguished by any item outside the clan, an idea that motivated the choice of the term.

Clans can be collapsed into single vertices without any ambiguity about how the edges look like after the collapse: from the perspective of an outside vertex, or indeed of a disjoint second clan, all the edges connecting to nodes inside the clan are of the same equivalence class, so that this class can be chosen for edges upon collapsing disjoint clans into vertices. Thus, we consider *strong clans*, those that don't intersect any other clan, and by collapsing them into vertices we get a tree-like organization of smaller (hopefully *much* smaller and clearer)

2-structures. More precisely, each maximal strong clan is collapsed to a single vertex; then, each of these vertices is "opened" to find that its internal components form also a 2-structure, which is decomposed recursively.

The decomposition recursion reaches a limit in two cases. In a *complete clan*, all the edges are in the same equivalence class, every subset is a clan, and there are no strong subclans; while in a primitive clan there are no nontrivial subclans at all. It is a theorem of the theory of 2-structures that the nodes of the clan decomposition of a symmetric 2-structure are all primitive or complete clans.

Tool functionality

The tool runs as a CLI call like: `degais dataset` to which some tuning options can be added. Extension `.td` (standing for transactional dataset) is assumed for the file name if it is not present. The data file is expected to contain a transactional dataset: a sequence of transactions, one per line, each consisting of a set of items. Items are (almost) arbitrary strings separated by spaces; however, characters ':' and '-' should not appear in items. Available options include `--coloring`, which selects the way labels ("colors") are decided among the options described above; `--param` that provides a value needed by some coloring strategies; and `--freq_thr` that applies a frequency threshold so as to display only some top frequent items.

Installation, option details and some example runs are provided in the current documentation; let's just mention here that the standard `pipx install degais` is expected to work, if the external dependencies (just Python 3 and GraphViz) are ready. Once installed, just use CLI calls.

Additional explanations, connections to other data analysis approaches, and details of the sophisticate algorithmics behind this tool will be published elsewhere; a preliminary version can be found here.

Examples

To start with, we advise to run first the example given in the Summary section above. File `e5.td` can be found in the `testdata` folder of the GitHub repository. The indicated image will be shown.

Then, try the call `degais e5 --coloring thresh --param 2`: it will show how the more frequently occurring a and b are correspondingly distinguished. One image shows the decomposition whereas the second one provides a legend allowing the user to connect the colors to actual frequency co-occurrence intervals.

By default, the outcome is shown as a graph, that is, non-co-occurring items are shown disconnected. If one wants to show a bona-fide 2-structure decomposition, that is, treat zero as one more possibility of its corresponding interval, the way to do it is to add the option `--complete` to the call: for instance, added to the thresholded call with parameter 2 on e5 as in the previous paragraph, the outcome will differ substantially, because 0 co-occurrences will not receive anymore a different color than 1 co-occurrence.

The case described in (Balcázar et al., 2018), based on one of the variants of the famous Titanic dataset, can be replicated easily on an appropriate, transactional version (also available in the `testdata` folder of the GitHub repository) by calling `degais` on it without any additional options.

Furthermore, the documentation includes several additional examples on additional datasets.

Acknowledgements

Supported by Ministerio de Ciencia e Innovación MCIN/AEI/10.13039/501100011033 under grant PID2020-112581GB-C21 (MOTION).

References

Balcázar, J. L., Piceno, M. E., & Rodríguez-Navas, L. (2018). Decomposition of quantitative Gaifman graphs as a data analysis tool. In W. Duivesteijn, A. Siebes, & A. Ukkonen (Eds.), *Advances in intelligent data analysis XVII - 17th international symposium, IDA 2018, 's-Hertogenbosch, The Netherlands, october 24-26, 2018, proceedings* (Vol. 11191, pp. 238–250). Springer. https://doi.org/10.1007/978-3-030-01768-2_20

Ehrenfeucht, A., Harju, T., & Rozenberg, G. (1999). *The theory of 2-structures - A framework for decomposition and transformation of graphs*. World Scientific. ISBN: 978-981-02-4042-4

Libkin, L. (2004). *Elements of finite model theory*. Springer. https://doi.org/10.1007/978-3-662-07003-1

Piceno, M. E. (2020). *Data analysis through graph decomposition* [PhD thesis, Universitat Politècnica de Catalunya]. http://hdl.handle.net/10803/669758

Piceno, M. E., Rodríguez-Navas, L., & Balcázar, J. L. (2021). Co-occurrence patterns in diagnostic data. *Computational Intelligence*, *37*(4), 1499–1514. https://doi.org/10.1111/coin.12317

Rodríguez-Navas, L. (2017). *Estructures de grafs amb equivalències d'arestes aplicades a l'anàlisi de dades relacionals (graduation project)*. Universitat Politècnica de Catalunya.