Strategies for improving performance of evolutionary biclustering algorithm EBIC

Patryk Orzechowski*†
University of Pennsylvania
Philadelphia, PA 19104, USA
patryk.orzechowski@gmail.com

Jason H. Moore
University of Pennsylvania
Philadelphia, PA 19104
jhmoore@upenn.edu

ABSTRACT

Biclustering is a growing in popularity machine learning technique which searches for patterns in subsets of rows and subsets of columns. One of the recent advances in biclustering was the development of EBIC, a multi-GPU method based on evolutionary computation, which was demonstrated to outperform some of the leading methods in the field. In this short paper, we evaluate a couple of potential improvements to the method.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Computing methodologies** → **Cluster analysis**; *Search methodologies*; Bio-inspired approaches; • **Theory of computation** → *Massively parallel algorithms*;

KEYWORDS

biclustering, data mining, machine learning, evolutionary computation

ACM Reference Format:
Patryk Orzechowski and Jason H. Moore. 2019. Strategies for improving performance of evolutionary biclustering algorithm EBIC. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322046

1 INTRODUCTION

Biclustering is an increasingly popular data mining technique, which focuses on finding similarities between selected rows and selected columns of the input data. Biclustering methods return a series of biclusters – subsets of rows and subsets of columns with certain characteristics [1, 2], such as row-constant (the values in each row are exact, but the values between rows may differ), column-constant (similarly, but for columns), shift, scale, shift-scale (where each row is a shifted and/or scaled version of a base row), trend preserving patterns (monotonously increasing), or other.

Although multiple methods have been proposed so far, it wasn't until 2018 that any method accurately identified the majority of aforementioned patterns with sufficient accuracy [1, 7, 10, 12]. Such advance in the field was the development of Evolutionary search-based BIClustering (EBIC), which achieved very high recovery and relevance scores across different patterns. The method outperformed some of the leading biclustering methods by a large margin and was further optimized in order to handle big data [6, 9].

The main focus of this paper is indicating potential areas in which EBIC could be further improved. We evaluate different variants of the method that may potentially improve its performance.

2 METHODS

EBIC is a multi-GPU biclustering method based on evolutionary computation [6, 8, 9]. The method represents hybrid biclustering approaches for data mining [3–5]. EBIC facilitates multiple evolutionary strategies, including crowding, elitism or tabu list. Each bicluster in EBIC is represented as a series of columns which enforces a monotonously increasing ordering of rows. After initialization of biclusters and calculation of their fitnesses in parallel on GPUs, genetic operators (i.e. one of 4 different types of mutation or a crossover) are used in order to modify the set of biclusters. The main workflow of EBIC is presented in Figure 1.

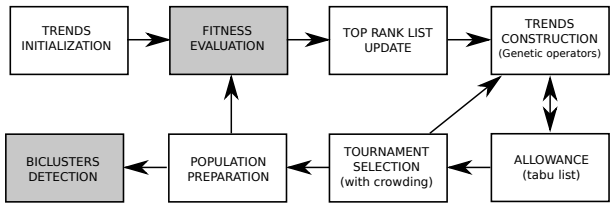

Figure 1: The basic scheme of EBIC. Darkened are parts of the method executed in parallel on GPUs.

For evaluation, a collection of 90 datasets with different square patterns from Wang et al. were used [12]. Narrow and overlapping patterns were beyond the scope of this paper. In each of the scenarios the methods were expected to detect from 3 to 5 biclusters of size 15x15, 20x20 and 25x25, which were implanted within the matrix of size 150x100, 200x150 and 300x200 respectively. Clustering Error (CE) was used as a quality measure. This subspace clustering evaluation metric proposed by Patrikainen et al. [11] allows for more objective evaluation of the performances, as it heavily

*corresponding author
†Patryk Orzechowski is also affiliated with Department of Automatics and Robotics, AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow, Poland

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322046

Table 1: The performance of variants of EBIC on different patterns according to Clustering Error (CE) metric. All the candidates were run for 5k iterations and compared with baseline version of the method at 5k and 20k iterations. The higher CE score, the better. The results outperforming a baseline version of EBIC with 5k iterations are in bold.

Algorithm\Patterns	Trend-preserving	Column-constant	Row-constant	Shift-scale	Shift	Scale
ebic-20k	*1.000000*	*0.996800*	*0.993442*	*0.796064*	*0.951548*	*0.717118*
ebic-5k	*0.961967*	*0.844396*	*0.977697*	*0.765907*	*0.846328*	*0.656557*
ebic-init	**0.967374**	**0.888879**	**0.993442**	0.741710	**0.854194**	0.596353
ebic-long	0.939329	**0.846344**	**0.978811**	**0.778567**	**0.849951**	0.612509
ebic-large	0.799447	0.792879	0.861758	0.544965	0.690673	0.452145
ebic-both	0.481338	0.516778	0.923864	0.265870	0.548694	0.234442
ebic-j2	0.875015	**0.848505**	0.976379	**0.841831**	0.806257	0.643479
ebic-post	**0.963538**	**0.870206**	0.977361	**0.772749**	**0.852978**	**0.705894**

penalizes incorrect assignments. Each of the candidates was run for 5k iterations (the default of EBIC). The recommended setting of EBIC for synthetic datasets (20k iterations) was included as the reference.

2.1 Candidates for improving EBIC

Based on the observation of EBIC performance, we have indicated a couple of areas that could potentially lead to improving the method. In this paper we focused on initialization of the first population, increasing the size of the population at cost of the number of iterations, modification of fitness function and postprocessing of the results. The following prototypes were evaluated in this paper:

Sensible initialization (ebic-init). Each generated trend is sorted according to the order of the randomly selected row.

Initialization with longer trends (ebic-long). The first population is initialized with trends of 4-8 columns, instead of 2-4 by default.

Increasing the size of the population at cost of a number of iterations (ebic-large). The total number of evaluations remains the same, but a larger population is used at each iteration.

Combination of crossover and mutation (ebic-both). In each iteration after crossover, one of the mutation operators is used.

Fitness function modification (ebic-j2). The algorithm is encouraged even more to find biclusters with a larger number of columns.

Postprocessing of the results (ebic-post). Additional filtering is performed at the final step to eliminate trends that overlap with each other by more than 50% (the default rate of overlap is 75%).

3 RESULTS

The averaged CE score from 15 datasets is presented in Table 1. It might be noticed that for some of the patterns three of the variations of EBIC (ebic-init, ebic-long and ebic-post) performed slightly better than the baseline method, but still not as good as the recommended setting. Three other candidates (ebic-large, ebic-both and ebic-j2) were worse than the baseline model in the majority of the scenarios.

4 CONCLUSIONS

In this paper we proposed and evaluated six different modifications of EBIC. Some of the prototyped variations of the method performed slightly better than the baseline method, but those difference weren't significant enough to offer a major improvement for the method. Our future effort will be focused on improving the speed of the convergence of EBIC as well as its performance for detection of scale and shift-scale patterns.

ACKNOWLEDGMENTS

This research was supported in part by PLGrid Infrastructure and by NIH grants LM010098, LM012601 and AI116794.

REFERENCES

[1] Kemal Eren, Mehmet Deveci, Onur Küçüktunç, and Ümit V Çatalyürek. 2013. A comparative analysis of biclustering algorithms for gene expression data. *Briefings in bioinformatics* 14, 3 (2013), 279–292.
[2] S. C. Madeira and A. L. Oliveira. 2004. Biclustering algorithms for biological data analysis: a survey. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on* 1, 1 (2004), 24–45.
[3] Patryk Orzechowski and Krzysztof Boryczko. 2016. Hybrid Biclustering Algorithms for Data Mining. In *Applications of Evolutionary Computation*, Giovanni Squillero and Paolo Burelli (Eds.). Springer International Publishing, Cham, 156–168.
[4] Patryk Orzechowski and Krzysztof Boryczko. 2016. Propagation-based biclustering algorithm for extracting inclusion-maximal motifs. *Computing and Informatics* 35, 2 (2016), 391–410.
[5] Patryk Orzechowski and Krzysztof Boryczko. 2016. Text Mining with Hybrid Biclustering Algorithms. In *Artificial Intelligence and Soft Computing*, Leszek Rutkowski, Marcin Korytkowski, Rafał Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada (Eds.). Springer International Publishing, Cham, 102–113.
[6] Patryk Orzechowski and Jason H. Moore. 2019. EBIC: an open source software for high-dimensional and big data analyses. *Bioinformatics* (2019), btz027. https://doi.org/10.1093/bioinformatics/btz027
[7] Patryk Orzechowski, Artur Pańszczyk, Xiuzhen Huang, and Jason H. Moore. 2018. runibic: a Bioconductor package for parallel row-based biclustering of gene expression data. *Bioinformatics* 34, 24 (2018), 4302–4304. https://doi.org/10.1093/bioinformatics/bty512
[8] Patryk Orzechowski, Moshe Sipper, Xiuzhen Huang, and Jason H Moore. 2018. EBIC: a next-generation evolutionary-based parallel biclustering method. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 59–60.
[9] Patryk Orzechowski, Moshe Sipper, Xiuzhen Huang, and Jason H. Moore. 2018. EBIC: an evolutionary-based parallel biclustering algorithm for pattern discovery. *Bioinformatics* 34, 21 (05 2018), 3719–3726. https://doi.org/10.1093/bioinformatics/bty401
[10] Victor A Padilha and Ricardo JGB Campello. 2017. A systematic comparative evaluation of biclustering techniques. *BMC bioinformatics* 18, 1 (2017), 55.
[11] Anne Patrikainen and Marina Meila. 2006. Comparing subspace clusterings. *IEEE Transactions on Knowledge and Data Engineering* 18, 7 (2006), 902–916.
[12] Zhenjia Wang, Guojun Li, Robert W Robinson, and Xiuzhen Huang. 2016. UniBic: Sequential row-based biclustering algorithm for analysis of gene expression data. *Scientific reports* 6 (2016).