

Multiobjective Optimization of Co-Clustering Ensembles

Francesco Gullo
Yahoo! Research Barcelona
gullo@yahoo-inc.com

AKM Khaled Ahsan Talukder
Department of Computer Science
George Mason University
atalukde@gmu.edu

Sean Luke
Department of Computer Science
George Mason University
sean@cs.gmu.edu

Carlotta Domeniconi
Department of Computer Science
George Mason University
carlotta@cs.gmu.edu

Andrea Tagarelli
Department of Electronics, Computer, and Systems Sciences
University of Calabria, Italy
tagarelli@deis.unical.it

ABSTRACT

Co-clustering is a machine learning task where the goal is to simultaneously develop clusters of the data and of their respective features. We address the use of *co-clustering ensembles* to establish a consensus co-clustering over the data. In this paper we develop a new preference-based multiobjective optimization algorithm to compete with a previous gradient ascent approach in finding optimal co-clustering ensembles. Unlike the gradient ascent algorithm, our approach once tackles the co-clustering problem with multiple heuristics, then applies the gradient ascent algorithm's joint heuristic as a preference selection procedure. We are able to significantly outperform the gradient ascent algorithm on feature clustering and on problems with smaller datasets.

Categories and Subject Descriptors

I.2.m [Artificial Intelligence]: Miscellaneous—*Evolutionary Computation*

General Terms

Algorithms, Experimentation

Keywords

Multiobjective Optimization, Co-Clustering, Ensembles

1. INTRODUCTION

This paper describes a novel application of multiobjective optimization to the problem of producing optimal *co-clustering ensembles*. Co-clustering is an unsupervised machine learning technique for identifying groups of *objects* related by their similar *feature values*. Co-clustering is distinguished from ordinary clustering in that it simultaneously discovers clusters of similar objects with regard to the values, and clusters of similar features with regard to the objects related by them. Often these cluster pairs do a better job of identifying the underlying structure in the data.

Ensemble learning is an approach to producing higher quality learned solutions by combining the results of several

different kinds of learning algorithms. These are fed into an ensemble learning algorithm to produce a solution (in our case, a *consensus co-clustering*) which ideally would be superior to any one member of the ensemble. Many such machine learning ensemble algorithms are essentially optimizers, hunting for sets of weights that produce a solution which is as optimal as possible according to one or more heuristic criteria.

In prior work [1, 2] we applied both a rudimentary multiobjective optimization algorithm (or MOEA) and a custom gradient ascent approaches to search for optimal consensus co-clusterings. The multiobjective optimization technique bred candidate consensus co-clusterings using two heuristic objective functions: one which examined how well the co-clustering clustered the features, and another which examined how well it clustered the objects. The gradient ascent approach instead used a single (and more discriminating) combined heuristic function, and performed better.

Here we revise the multiobjective optimization approach with a more advanced technique based on a customized version of the NSGA-II algorithm. This version uses the two heuristics, originally found in the earlier multiobjective optimization approach, to build a Pareto front (NSGA-II's "archive"), but then uses the advanced combined heuristic function to select parents from the front to breed new children. We provide a single solution at the end of the run by returning the member of the final Pareto front which maximized the combined heuristic function.

For a fuller description of this work, see [3].

2. PREVIOUS WORK

Previous literature has tackled clustering from an evolutionary computation perspective, and most commonly, from a multiobjective one. Kin et al [6] tackled the problem of optimizing the right model framework to fit to the data using a combination of multiobjective evolutionary optimization and local search (K-means, EM). Handl and Knowles [4] applied multiobjective optimization to examine the trade-off between the number of clusters and cluster solution quality. Oliveira et al examined cluster ensemble discovery by applying multiple objective optimization to cluster features [7]. Jin and Sendhoff [5] applied multiobjective optimization in a wide variety of machine learning contexts, including unsupervised clustering. However only our prior work [1] has examined co-clustering from an evolutionary optimization standpoint.

Data Set	Algorithm	Θ_f	P -value	Θ_o	P -value	Θ_{of}	P -value
Iris	MOEA-CB-PCE	0.5564	(> 99.8%)	0.0813		0.7348	(> 95.0%)
	CB-PCE	0.2332		0.4702	(> 99.8%)	0.6902	
Wine	MOEA-CB-PCE	0.5945	(> 99.8%)	0.1943		0.7748	(> 99.8%)
	CB-PCE	0.1142		0.4119	(> 99.0%)	0.3402	
Glass	MOEA-CB-PCE	0.7464	(> 99.8%)	0.0834		0.9048	(> 99.8%)
	CB-PCE	0.1302		0.4702	(> 99.8%)	0.1203	
E. Coli	MOEA-CB-PCE	-0.003		-0.0013		0.011	
	CB-PCE	0.0046	(> 99.8%)	0.0894	(> 99.8%)	0.0881	(> 99.8%)
TraceData	MOEA-CB-PCE	-0.00241		-0.0492		-0.0587	
	CB-PCE	0.018	(> 99.8%)	0.2347	(> 99.8%)	0.2493	(> 99.8%)

Table 1: Results on the different benchmark datasets. CB-PCE is compared against a MOEA employing the same objective function in its tournament selector, hill-climbing, and final Pareto front reduction. Bold faced results are statistically significantly superior.

3. APPROACH

We have three objective functions available: $\Psi_o(\mathcal{C}, \mathcal{E})$, a heuristic optimization function over object clusters, $\Psi_f(\mathcal{C}, \mathcal{E})$, a heuristic optimization function over feature clusters, and $\Psi_{of}(\mathcal{C}, \mathcal{E})$, a heuristic optimization function jointly over both object and feature clusters. We applied a “preference-based” version of NSGA-II to this problem, where the experimenter provides *a priori* information such as the region on the Pareto front which is preferred over others. $\Psi_{of}(\mathcal{C}, \mathcal{E})$ provided this preference information: at the end of the run, we returned the individual on the Pareto front which maximized $\Psi_{of}(\mathcal{C}, \mathcal{E})$. Additionally, while the NSGA-II algorithm used $\Psi_o(\mathcal{C}, \mathcal{E})$ and $\Psi_f(\mathcal{C}, \mathcal{E})$ as its two objective functions for purposes of building the archive, it used $\Psi_{of}(\mathcal{C}, \mathcal{E})$ as the objective function for assembling the population from the archive via tournament selection. Finally, every 25 generations we added a bit of hill-climbing based on $\Psi_{of}(\mathcal{C}, \mathcal{E})$. The details of the algorithm may be found in [3].

4. EXPERIMENTS

We performed experiments comparing our MOEA with two gradient ascent techniques described in [2], called CB-PCE and FCB-PCE. CB-PCE optimizes with a slow but precise version of $\Psi_{of}(\mathcal{C}, \mathcal{E})$, and FCB-PCE uses a fast approximation. In each case we compared the technique against our MOEA using the same version of $\Psi_{of}(\mathcal{C}, \mathcal{E})$.

The experiments were done on four benchmark datasets from the UCI Machine Learning Repository¹ (Iris, Wine, Glass, and E. Coli) and on a time-series dataset from UCR Time Series Classification Page² (Trace Data). Because of very high time cost involved, we performed only ten independent runs for each combination of technique and dataset. We performed non-parametric two-tailed t-tests to determine statistical significance. P -values are shown in Table 1.

A co-clustering has two equally important measures of accuracy: how well it properly clusters the object data, and how well it properly clusters the features. We used the Θ_o , Θ_f measures from [2], respectively, and also the Θ_{of} measure provided an overall assessment taking into account both co-clustering aspects.

Table 1 summarizes the performance results for CB-PCE. For FCB-PCE (similar), see [3]. Our MOEA model tended

to attain better consensus co-clustering in terms of Θ_f and Θ_{of} on the simpler problems (Iris, Wine, Glass). However on E. Coli and the much more difficult Trace Data, it could not outperform the gradient ascent algorithm.

This was the opposite of what we had expected. We had hypothesized that local optima would trap the gradient ascent algorithm on the more complex problems, but on the simpler problems a straightforward gradient ascent would outperform a global algorithm like the MOEA. This suggests that the underlying spaces may not be what we had expected.

5. ACKNOWLEDGEMENTS

This work was supported in part by NSF grant 0916870.

6. REFERENCES

- [1] F. Gullo, C. Domeniconi, and A. Tagarelli. Projective clustering ensembles. In *Proceedings of the IEEE International Conference on Data Mining*, 2009.
- [2] F. Gullo, C. Domeniconi, and A. Tagarelli. Advancing data clustering via projective clustering ensembles. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2011.
- [3] F. Gullo, A. K. A. Talukder, S. Luke, C. Domeniconi, and A. Tagarelli. Multiobjective optimization of co-clustering ensembles. Technical Report GMU-CS-TR-2012-3, Department of Computer Science, George Mason University, 4400 University Drive MSN 4A5, Fairfax, VA 22030-4444 USA, 2012.
- [4] J. Handl and J. Knowles. Exploiting the tradeoff: The benefits of multiple objectives in data clustering. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, pages 547–560, 2005.
- [5] Y. Jin and B. Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics Part C*, 38(3):397–415, 2008.
- [6] Y. Kim, W. N. Street, and F. Menczer. Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*, 6:531–556, 2002.
- [7] L. S. Oliveira, M. Morita, and R. Sabourin. Feature selection for ensembles using the multi-objective optimization approach. In Y. Jin, editor, *Multi-Objective Machine Learning*, pages 49–74. Springer, 2006.

¹<http://archive.ics.uci.edu/ml/>

²http://www.cs.ucr.edu/~eamonn/time_series_data/