An Evolutionary Subspace Clustering Algorithm for High-Dimensional Data

S. N. Nourashrafeddin
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia
Canada B3H 4R2
nourashr@cs.dal.ca

Dirk V. Arnold
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia
Canada B3H 4R2
dirk@cs.dal.ca

Evangelos Milios
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia
Canada B3H 4R2
eem@cs.dal.ca

ABSTRACT

We present an algorithm for generating subspace clusterings of large data sets with many attributes. An evolutionary algorithm is used to form groups of relevant attributes. Those groups are replaced by their centroids, making it possible to cluster the objects in a much lower dimensional space. Preliminary experiments with scalable synthetic data sets suggest that the algorithm generates competitive clusterings while scaling quite well.

Categories and Subject Descriptors

I.5.3 [**Clustering**]: Algorithms

General Terms

Algorithms

Keywords

Subspace clustering, evolutionary algorithm

1. INTRODUCTION

An instance of a clustering problem is described by a two-dimensional matrix. The rows of the matrix correspond to the objects to be clustered. The columns correspond to attributes. The entries of the matrix are numerical values that indicate the (often normalised) degree of presence of an attribute in the respective object. For a matrix with m rows and n columns, each object corresponds to a point in n-dimensional attribute space. The goal of clustering is to partition the objects into groups such that objects within a group are similar in that they have short pairwise distances from each other, while objects in different groups are separated by large distances.

If the number of attributes n is large, then basic clustering techniques, such as k-means, may fail to generate useful results. Meaningful groups of objects may be identified by the presence or absence of a few relevant attributes. The objects are close in the subspace corresponding to those attributes. However, that proximity fails to register if distances are computed in the full-dimensional space as it is hidden in the noise from the remaining attributes. Algorithms identifying groups of attributes which characterise a cluster of objects are referred to as subspace clustering algorithms [1]. The runtime of most subspace clustering algorithms in use today scales poorly with the number of attributes, making them not applicable to problems where n is large.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

2. ALGORITHM

Our subspace clustering algorithm, *EsubClus*, proceeds in two phases. In the first phase, an evolutionary algorithm (EA) is used to identify a small number of groups of relevant attributes that can be used to cluster the data objects. In the second phase, the objects are clustered in the low-dimensional space spanned by the centroids of those groups. The first phase of the algorithm thus removes irrelevant attributes while at the same time exploiting redundancies among relevant ones.

Each individual in the EA encodes a limited, small number of attributes. Corresponding groups are formed by averaging the attributes encoded in the individuals with their nearest neighbours, where similarity of attributes is defined in terms of cosine similarity of their corresponding columns in the data matrix. Mutation adds, removes, or replaces attributes encoded in individuals. To determine the fitness of an individual, the entries of the column vectors of the data matrix within the attribute groups are averaged, resulting in an m-dimensional vector for each group of attributes. X-means with the maximum number of clusters set to two is applied to each one of those vectors individually, potentially identifying two one-dimensional clusters one of which consists of indices corresponding to data objects partially identified by the attribute group. The fitness function favours tightly clustered groups of attributes and objects with large degrees of separation from other groups. The EA is terminated if no improvement has been made in a number of generations.

The second phase of *Esubclus* has the purpose of identifying clusters of objects, which are characterised by combinations of relevant attribute groups. Due to the small number of attribute groups, this goal can be accomplished by interactive thresholding.

3. EXPERIMENTS

In order to evaluate the performance of *EsubClus*, we have generated scalable synthetic data sets with varying numbers of objects m, numbers of attributes n, and numbers of clusters k. Each data set has been generated by assigning a number of characteristic attributes to each cluster of objects. Characteristic attributes of different clusters may

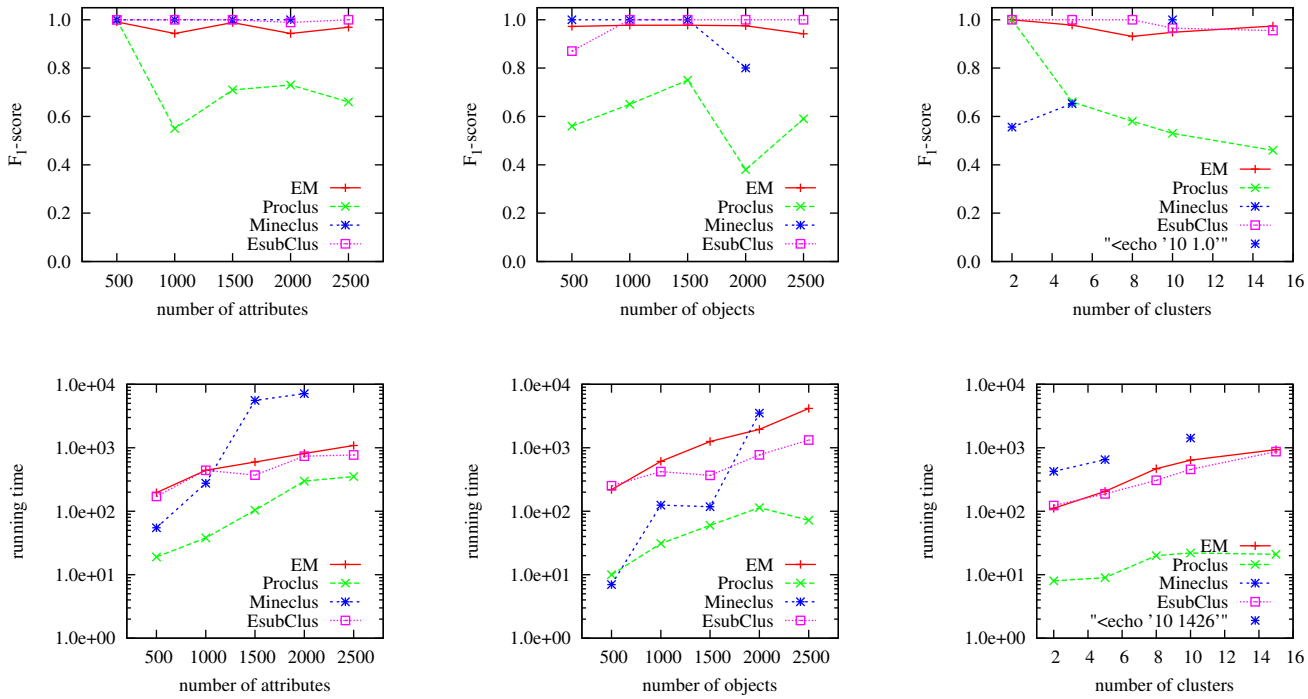

Figure 1: Comparison of F_1-score measures and running times (in seconds) of Expectation Maximisation, Proclus, Mineclus, and EsubClus.

overlap, and further attributes that are not characteristic of any cluster are added. Entries to the data matrix are sampled from normal distributions with means in $[2, 4]$ and standard deviation in $[2, 3]$ for characteristic attributes, and from a uniform distribution in $[0, 3]$ for the remaining attributes.

We use the F_1-score (i.e., the harmonic mean of precision and recall) as a performance metric. Figure 1 compares the performance of *EsubClus* with those of the expectation maximisation algorithm (EM) and two popular subspace clustering algorithms, *Proclus* and *Mineclus*. Implementations of all three of those are available in WEKA with the OpenSubspace framework [2]. From left to right, the graphs vary the number of attributes (with $n = 500$ and $k = 5$ fixed), the number of data objects (with $m = 500$ and $k = 5$ fixed) and the number of clusters (with $m = n = 500$ fixed). The EM algorithm generates good clusterings throughout. *Proclus* is the fastest of the algorithms, but the quality of clusterings it generates is comparatively poor. *Mineclus* fails to generate clusterings within the preset maximum time of 24 hours for large values of n and k. The quality of the results generated by *EsubClus* is at least on par with the best of the other algorithms. The computational cost of the algorithm is similar to or somewhat below that of EM and grows much slower than that of *Mineclus*.

4. FUTURE WORK

A more extensive experimental comparison that involves further subspace clustering algorithms, including evolutionary ones [3, 4], is necessary in order to judge the promise of our approach. It is also desirable to evaluate *EsubClus* using real-world data sets. Our interest in subspace clustering stems from the need to cluster text, which is often represented as a high-dimensional, sparse data set in the bag of words model, and where the number of attributes is in the thousands. Finally, we believe that the approach of identifying a small number of relevant attribute groups opens up possibilities for visualising the second stage clustering problem, which allows applying interactive data clustering techniques.

Acknowledgements

This research was supported by the NSERC (Natural Sciences and Engineering Research Council of Canada) Business Intelligence Network.

5. REFERENCES

[1] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1–58, 2009.

[2] E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment*, 2(1):1270–1281, 2009.

[3] I. A. Sarafis, P. W. Trinder, and A. M. S. Zalzala. Towards effective subspace clustering with an evolutionary algorithm. In *IEEE Congress on Evolutionary Computation*, pages 797–806, 2003.

[4] A. Vahdat, M. Heywood, and N. Zincir-Heywood. Bottom-up evolutionary subspace clustering. In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2010.