A Traveling Salesman's Approach to Clustering Gene Expression Data

Sharlee Climer and Weixiong Zhang
Department of Computer Science and Engineering,
Washington University, St. Louis, MO 63130-4899, USA

Abstract

Given a matrix of values, rearrangement clustering involves rearranging the rows of the matrix and identifying cluster boundaries within the linear ordering of the rows. The TSP+k algorithm for rearrangement clustering was presented in [3] and its implementation is described in this note. Using this code, we solve a 2,467-gene expression data clustering problem and identify "good" clusters that contain close to eight times the number of genes that were clustered by Eisen et al. (1998). Furthermore, we identify 106 functional groups that were overlooked in that paper. We make our implementation available to the general public for applications of gene expression data analysis.

Availability:
C++ source code is freely available at http://www.climer.us and http://www.cse.wustl.edu/∼zhang/projects/software.html.

Contact:
sharlee@climer.us, zhang@cse.wustl.edu.

Supplementary information:
Algorithm and implementation details can be found at http://www.climer.us/paper08.pdf.

1 Introduction

Given a matrix of values in which the rows correspond to objects and the columns correspond to features of the objects, *rearrangement clustering* [2] is the problem of rearranging the rows of the matrix and identifying cluster boundaries within the linear ordering of the rows. The rearrangement and cluster boundaries are selected so as to maximize the sum of similarities between adjacent rows within the clusters. Referred to by various names and reinvented several times, this clustering technique has been extensively used in many fields over the last three decades and has recently been applied to the clustering of gene expression data [3, 5].

Rearrangement clustering requires a linear ordering of objects. Visualization of complex data is enhanced by arranging objects in this manner [4, 7]. Furthermore, this property may be advantageous when it is useful to identify elongated, but contiguous, clusters or irregularly shaped clusters as in Figure 1. Although rearrangement clustering requires the objects be linearly ordered within their respective clusters,

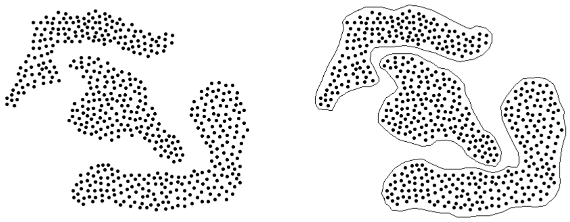

Figure 1: (a) A dataset with Euclidean distance used for the distance measure. (b) Intuitive clustering of the dataset. Many objects are closer to the center of a different cluster than their own and the diameters are not minimized.

it doesn't suffer from the drawbacks that can arise when the objective is based on minimizing diameters or minimizing distances of objects from the centers of their respective clusters.

There are three steps to clustering using TSP+k. First, the expression data is converted to a Traveling Salesman Problem (TSP), based on the desired number of clusters. Second, the TSP is solved. Third, the expression data is rearranged according to the TSP solution and cluster boundaries are indicated. Our code is used for the first and last steps and any TSP solver can be used for the second step.

There has been a vast amount of research devoted to quickly finding high-quality approximate solutions for the TSP, yielding a wealth of available code [6]. Advances in solving TSPs to optimality have been equally impressive. For all of our experiments, we use Concorde [1], an award winning TSP solver that has successfully solved a record 15,112-city TSP instance to optimality. The Concorde code is publicly available at *http://www.tsp.gatech.edu//concorde.html.*

Our code uses the Pearson correlation coefficient for the similarity measure and can be easily revised for an alternate measure.

2 Demonstration

In this section, we summarize the results of using TSP+k for a yeast gene dataset that was clustered using a hierarchical technique in [4]. The dataset consists of 2,467 genes in the budding yeast *Saccharomyces cerevisiae* that were studied during the diauxic shift, mitotic cell division cycle, sporulation, and temperature and reducing shocks, yielding 79 measurements that are used as the features for the genes.

To evaluate the performance of TSP+k, we use Gene Ontology (GO) Term Finder (*http://www.yeastgenome.org/*), a tool for finding functionally related groups of yeast genes in a given cluster. This tool calculates a p-value that indicates the likelihood of observing a group of x genes with a particular functional annotation in a cluster containing y genes, given that M genes have this annotation in the total population of N genes. In these results, we label a functional group as being "good" if it has a p-value with an order of magnitude that is less than or equal to 10^{-7}.

We ran TSP+k with k equal to 50 through 350, using an increment of 50. Table 1 contains the results of these trials. For all the values of k that we tested, TSP+k finds more clusters containing "good" functional

k	(a)	(b)	(c)	(d)	(e)	(f)
50	11	191.3	85.3%	29	52.4%	24
100	13	129.6	68.3%	56	29.5%	39
150	15	80.4	48.9%	85	23.1%	44
200	12	81.1	39.4%	123	15.6%	42
250	15	71.1	43.2%	154	15.6%	45
300	16	63.8	41.4%	191	14.7%	38
350	14	50.0	28.4%	236	12.2%	40
Eisen *et al.*	10	26.3	10.7%	-	-	71

Table 1: Results for 2,467 yeast gene clustering where "good" functional groups are defined as those with p-values with orders of magnitude $\leq 10^{-7}$. (a) Number of clusters found containing one or more "good" functional groups. (b) Average size of clusters containing "good" functional groups. (c) Percentage of genes that are placed into clusters containing "good" functional groups. (d) Number of singleton clusters. (e) Percentage of clusters that contain "good" functional groups when the singletons are ignored. (f) Number of "good" functional groups.

groups than the ten found in [4]. Furthermore, the average cluster size is larger resulting in more genes being clustered.

An interesting result of these tests is the large number of singletons, as listed in Table 1. In all cases, more than half of the clusters contain singletons. Yet there is not a dominance of clusters containing only two or three genes. For instance, when $k = 50$, there are 29 singleton clusters, yet there is only one cluster containing two genes and no clusters containing three genes. Many of the singletons that are found may correspond to outliers in the data, suggesting that TSP+k may be useful for identifying outliers. Note that when the singletons are ignored, the percentage of clusters that are "good" is substantial, as shown in Table 1.

TSP+k missed 8 of the distinct functional groups found in [4]. However, Eisen et al. (1998) missed 106 distinct functional groups found by TSP+k. For example, in all but one trial, TSP+k identified a "good" cluster containing functionally related groups of genes involved in carbohydrate transporter activity and six related functions. All seven of these functional groups were overlooked in [4].

There are only four genes annotated to acid phosphatase activity and only two genes annotated to pyruvate carboxylase activity in the 2,467 gene set. In every trial, the four genes appeared consecutively within a cluster and the two genes were adjacent within another cluster. Again, these functionally related groups were missed in [4].

Tables listing the functional groups for each trial can be found on the web at *http://www.climer.us/TSPk.html* and *http://www.cse.wustl.edu/~zhang/projects/software.html*.

Concorde was able to solve these large problems to optimality, although it had to be restarted a number of times. Each trial required between 20 to 40 minutes running time on an Athlon 1.9 MHz processor with two gigabytes memory.

As a final note, observe that in the work of Eisen et al. (1998), a domain expert was required to identify the clusters. When using TSP+k, the clusters are identified automatically, allowing its use in domains that are not well understood or when a domain expert is unavailable.

Acknowledgement

This research was supported in part by NDSEG and Olin Fellowships and by NSF grants IIS-0196057 and ITR/EIA-0113618.

References

[1] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. TSP cuts which do not conform to the template paradigm. In M. Junger and D. Naddef, editors, *Computational Combinatorial Optimization*, pages 261–304. Springer, 2001.

[2] S. Climer and W. Zhang. Take a walk and cluster genes: A TSP-based approach to optimal rearrangement clustering. In *21st International Conference on Machine Learning (ICML'04)*, pages 169–176, Banff, Canada, July 2004.

[3] S. Climer and W. Zhang. Rearrangement clustering: Pitfalls and remedies. Technical Report WUCSE-2005-2, Washington University, St. Louis, MO, January 2005.

[4] M.B. Eisen, P.T. Spellman, P.O Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. of the Natl. Acad. of Sciences*, 95(25):14863–8, 1998.

[5] Y. Liu, B. J. Ciliax, A. Pivoshenko, J. Civera, V. Dasigi, A. Ram, R. Dingledine, and S. B. Navathe. Evaluation of a new algorithm for keyword-based functional clustering of genes. In *8th International Conf. on Research in Computational Molecular Biology (RECOMB-04)*, San Diego, CA, March 2004. Poster paper.

[6] A. Lodi and A. P. Punnen. TSP software. In G. Gutin and A. Punnen, editors, *The Traveling Salesman Problem and its Variations*. Kluwer Academic, Norwell, MA, 2002.

[7] W. T. McCormick, P. J. Schweitzer, and T. W. White. Problem decomposition and data reorganization by a clustering technique. *Operations Research*, 20:993–1009, 1972.