Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Fuzzy clustering with biological knowledge for gene selection

Sampreeti Ghosh[a,*], Sushmita Mitra[a], Rana Dattagupta[b]

[a] Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India
[b] Department of Computer Science & Engineering, Jadavpur University, Kolkata, India

ARTICLE INFO

Article history:
Received 3 January 2013
Received in revised form 18 April 2013
Accepted 13 November 2013
Available online 11 December 2013

Keywords:
Fuzzy CLARANS
Feature selection
Gene ontology
Fold-change

ABSTRACT

This paper presents an application of Fuzzy Clustering of Large Applications based on Randomized Search (FCLARANS) for attribute clustering and dimensionality reduction in gene expression data. Domain knowledge based on gene ontology and differential gene expressions are employed in the process. The use of domain knowledge helps in the automated selection of biologically meaningful partitions. Gene ontology (GO) study helps in detecting biologically enriched and statistically significant clusters. Fold-change is measured to select the differentially expressed genes as the representatives of these clusters. Tools like Eisen plot and cluster profiles of these clusters help establish their coherence. Important representative features (or genes) are extracted from each enriched gene partition to form the reduced gene space. While the reduced gene set forms a biologically meaningful attribute space, it simultaneously leads to a decrease in computational burden. External validation of the reduced subspace, using various well-known classifiers, establishes the effectiveness of the proposed methodology on four sets of publicly available microarray gene expression data.

1. Introduction

A primary goal of bioinformatics is to increase the understanding of biological processes, possibly by the application and development of data mining techniques [1]. Analyzing biological data sets requires making sense of the data by inferring structure or generalization from it. With the advent of microarray technology, scientists started developing informatics tools for the analysis and information extraction from gene expression data. Due to the inherent characteristic of microarray data, involving high levels of noise, high cardinality of genes, and small samples size, feature selection becomes [2–4] an important and frequently used technique in gene expression analysis. Since the small number of samples narrows down the acquirable knowledge, it reduces the probability of correct decision making. An increase in the number of features may also lead to increased processing time and memory requirements. Therefore a selection of the proper features is an important preprocessing stage for classification; often directly associated with tissue category, disease state or clinical outcome. It reduces computation cost and increases classification accuracy.

Selection of features can be supervised [5,6] as well as unsupervised [8,7]. Simultaneous feature selection and clustering has also been reported in literature [9,10]. However, a subset selected by a supervised feature selection method may not always be good for unsupervised learning, and vice versa. It becomes desirable to have a feature selection method which works well in both unsupervised and supervised frameworks.

A central limitation of most commonly used algorithms is that they are unable to identify genes whose expression is similar to multiple, distinct gene groups. Moreover, gene expression data are redundant and noisy, leading to ambiguity and vagueness. Computational intelligence, particularly fuzzy sets, provides a natural framework to handle the associated uncertainty and ambiguity [11].

Clustering [12,13] is a useful exploratory technique for gene expression data as it groups similar objects together and allows the biologist to identify potentially meaningful relationships between genes. Those genes belonging to the same cluster are typically involved in related functions and are frequently co-regulated. Thus, grouping similar genes can provide a way to understand some of their (as yet) unknown functions. Often some genes are likely to be similarly expressed with different groups in response to different subsets of the experiments. Fuzzy clustering [14,15] facilitates the identification of such overlapping groups of genes by allowing one sample to simultaneously belong to multiple clusters. This becomes appropriate in many cases of uncertainty and noise. It is also suitable in modeling a single gene with multiple functionality.

The use of biological domain knowledge often serves to aid the process of data analysis [18], and can make the results more acceptable to biologists. The use of gene ontology [19] and fold-change [20] has been reported in literature. Biological knowledge

[*] Corresponding author.
E-mail addresses: sampreeti_t@isical.ac.in (S. Ghosh), sushmita@isical.ac.in (S. Mitra), rdattagupta@cse.jdvu.ac.in (R. Dattagupta).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.11.007

about co-expressed genes has also been incorporated in clustering [16,17], for determining quality-based partitions. Gene ontology (GO) annotations have been used to extend the *k*-medoids algorithm, such that genes with known function get clustered together [37]. The incorporation of biological knowledge provides a direction towards the extraction of meaningful groups of genes [38,39]. Computation of pairwise distances between gene annotation similarities has been used [40] to develop a fast software.

Gene ontology (GO) provides a common language to describe aspects of a gene product's biology, and is represented in a taxonomic form. The use of a consistent vocabulary allows genes from different species to be compared based on their GO annotations. Fold-change is used to measure the differential expressions of genes in a microarray experiment, thereby providing some indication about their importance. Differentially expressed genes are reported to be important due to their involvement in important biological processes [25]. Microarray technology enables the simultaneous measurement of the expression levels of genes throughout the genome. Its use in discovering genes, which are differentially expressed between two or more groups of patients, has many biomedical applications; including the identification of disease biomarkers that can potentially be used to understand and diagnose diseases in a better way. There exist several methods for the identification of such differentially expressed genes, and the choice of a method can profoundly affect the resultant set. Despite a wealth of available methods, biologists show a fondness for one of the earliest approach, viz. fold change [20,29], presumably because of its computational simplicity and interpretability.

In this article we propose a novel way of feature selection through fuzzy gene clustering, while incorporating biological knowledge. Since the number of genes is very large we use Fuzzy Clustering Large Applications based on Randomized search (FCLARANS) [21] for initial fuzzy partitioning based on similarity. Domain knowledge, involving gene ontology, is used for selecting the best clusters. These clusters are qualitatively evaluated in terms of Eisen plot and the constituent gene expression profiles. The differential expression of genes is computed, using fold-change, to select the most-representative feature (or gene) from each cluster to collectively form the reduced subset of genes. The classification accuracy is tested for external validation. We used WEKA [22] for implementing *k*-nearest neighbors (*k*-NN), decision tree C4.5, random forest, multilayer perceptron (MLP) and naive Bayes' (NB) classifiers. It has been observed that incorporation of biological knowledge, along with fuzzy clustering, lead to better performance over a smaller subset of genes.

The rest of the paper is structured as follows: Section 2 introduces the fuzzy clustering algorithm (Fuzzy CLARANS). The proposed feature selection method is described in Section 3. Case studies on *Colon*, *Medulloblastoma*, *Gastric* and *Leukemia* data are presented in Section 4. Finally, Section 5 provides the conclusion.

2. Fuzzy CLARANS

Here we describe the fuzzy clustering algorithm, Fuzzy CLARANS (FCLARANS) [21], which performs efficiently on large data. The inherent fuzziness in FCLARANS allows the handling of uncertainty, while reducing the computational time for searching neighbours and eliminating user-defined parameters. It incorporates the concept of fuzzy membership onto the framework of CLARANS [23] for manoeuvering uncertainty in the context of data mining. FCLARANS is employed here as a tool for clustering the large gene space. Any other clustering algorithm could be used as well.

Let us first briefly introduce CLARANS, before we move onto its fuzzy version FCLARANS. Typically CLARANS searches through a graph $G_{N,c}$, where node ν^q is represented by a set of c medoids (or centroids) $\{m_1^q, \ldots, m_c^q\}$ of the clusters. Two nodes are termed as neighbors if they differ by only one medoid, and are connected by an edge. More formally, two nodes $\nu^1 = \{m_1^1, \ldots, m_c^1\}$ and $\nu^2 = \{m_1^2, \ldots, m_c^2\}$ are termed neighbors if and only if the cardinality of the intersection of ν^1 and ν^2 is given as $card(\nu^1 \bigcap \nu^2) = c - 1$. Hence each node in the graph has $c * (N - c)$ neighbors. For each node ν^q we assign a cost function

$$J_c^q = \sum_{x_j \varepsilon U_i} \sum_{i=1}^{c} d_{ji}^q, \qquad (1)$$

where d_{ji}^q denotes the dissimilarity measure of the *j*th object x_j from the *i*th cluster medoid m_i^q in the *q*th node. The aim is to determine that set of *c*-medoids $\{m_1^0, \ldots, m_c^0\}$ at node ν^0, for which the corresponding cost is the minimum as compared to all other nodes in the graph.

The algorithm considers two parameters *numlocal*, representing the number of iterations (or runs) for the algorithm, and *maxneighbor*, the number of adjacent nodes (set of medoids) in the graph *G* that need to be searched upto convergence. These parameters are provided as input at the beginning. Random search is used to generate neighbors, by starting from an arbitrary node and randomly checking *maxneighbor* neighbors. If a neighbor represents a better partition, the process continues with this new node. Otherwise a local minimum is found, and the algorithm restarts until *numlocal* local minima are obtained. The variable *maxneighbor* is computed as

$$maxneighbor = p\% \text{ of } \{c * (N - c)\}, \qquad (2)$$

with *p* being provided as input by the user. Typically, $1.25 \le p \le 1.5$ [23]. The main steps of the algorithm are outlined as follows:

```
1. Set iteration counter $i \leftarrow 1$, and set a parameter mincost [measuring
   the minimum cost attained by Eq. (1)] to an arbitrarily large value.
   A pointer bestnode refers to the solution set.
2. Start randomly from any node $\nu^{current}$ in graph $G_{N,c}$, consisting of c
   medoids. Compute cost $J_c^{current}$ by Eq. (1).
3. Set node counter $j \leftarrow 1$.
4. Select randomly a neighbor $\nu^j$ of node $\nu^{current}$. Compute the cost $J_c^j$
   by Eq. (1).
5. If the criterion function improves as $J_c^j < J_c^{current}$
     Then set the current node to be this neighbor node by
   $current \leftarrow j$, and go to Step 3 to search among the neighbors of the
   new $\nu^{current}$
     Else increment j by one.
6. If $j \le maxneighbor$
     Then go to Step 4 to search among the remaining allowed
   neighbors (among its maxneighbor neighbours) of $\nu^{current}$
     Else calculate the average distance of patterns from medoids for
   this node; this requires one scan of the database.
7. If $J_c^{current} < mincost$
     Then set $mincost \leftarrow J_c^{current}$ and choose as a solution this set of
   medoids given by $bestnode \leftarrow current$.
8. Increment the number of iterations i by 1.
   If $i > numlocal$
     Then output bestnode as the solution set of medo-ids and halt
     Else go to Step 2 for the next iteration.
```

Fuzzification of CLARANS is done by incorporating fuzzy membership to the cost function of Eq. (1). It allows a pattern x_i to have finite, non-zero membership $\mu_{ij} \in [0, 1]$ to two or more partitions. FCLARANS now minimizes the cost function [21]

$$J_{fc}^q = \sum_{j=1}^{N} \sum_{i=1}^{c} (\mu_{ij}^q)^{m'} d_{ji}^q, \qquad (3)$$

where m' was chosen as 2 after several experiments. Here the distance component is weighted by the corresponding membership value. This is used in Steps 2, 4, 5, 7, of the algorithm CLARANS. Fuzzy partitioning is carried out through an iterative optimization