Data Mining Methods for Microarray Data Analysis

Lei Yu
Arizona State University, USA

Huan Liu
Arizona State University, USA

INTRODUCTION

The advent of gene expression microarray technology enables the simultaneous measurement of expression levels for thousands or tens of thousands of genes in a single experiment (Schena, et al., 1995). Analysis of gene expression microarray data presents unprecedented opportunities and challenges for data mining in areas such as gene clustering (Eisen, et al., 1998; Tamayo, et al., 1999), sample clustering and class discovery (Alon, et al., 1999; Golub, et al., 1999), sample class prediction (Golub, et al., 1999; Wu, et al., 2003), and gene selection (Xing, Jordan, & Karp, 2001; Yu & Liu, 2004). This article introduces the basic concepts of gene expression microarray data and describes relevant data-mining tasks. It briefly reviews the state-of-the-art methods for each data-mining task and identifies emerging challenges and future research directions in microarray data analysis.

BACKGROUND AND MOTIVATION

The rapid advances of gene expression microarray technology have provided scientists, for the first time, the opportunity of observing complex relationships among various genes in a genome by simultaneously measuring the expression levels of thousands of genes in massive experiments. In order to extract biologically meaningful insights from a plethora of data generated from microarray experiments, advanced data analysis techniques are in demand. Data-mining methods, which discover patterns, statistical or predictive models, and relationships among massive data, are effective tools for microarray data analysis.

Gene expression microarrays are silicon chips that simultaneously measure the expression levels of thousands of genes. The description of technologies for constructing these chips and measuring gene expression levels is beyond the scope of this article (refer to Draghici, 2003, for an introduction). Each expression level of a specific gene among thousands of genes measured in an experiment is eventually recorded as a numerical value. Expression levels of the same set of genes under study are normally accumulated through multiple experiments on different samples (or the same sample under different conditions) and recorded in a data matrix. In data mining, data are often stored in the form of a matrix, of which each column is described by a *feature* or *attribute* and each row consists of feature values and forms an *instance,* also called a *record* or *data point,* in a multidimensional space defined by the features. Figure 1 illustrates two ways of representing microarray data in a matrix form. In Figure 1a, each feature is a sample *(S)* and each instance is a gene (*G*). Each gene's expression levels are measured across all the samples (or conditions), so f_{ij} is the measurement of the expression level of the ith gene for the jth sample, where $i = 1,..., n$ and $j = 1,..., m$. In Figure 1b, the data matrix is the transpose of the one in Figure 1a, in which features are genes, and instances are samples. Sometimes, data in Figure 1b may have class labels c_i for each instance, represented in the last column. The class labels can be different types of diseases or phenotypes of the underlying samples. A typical microarray data set may contain thousands of genes but only a small number of samples (often less than 100). The number of samples is likely to remain small — at least for the near future — due to the expense of collecting microarray samples (Dougherty, 2001).

The two different forms of data shown in Figure 1 have different data-mining tasks. When instances are genes (Figure 1a), *gene clustering* can be performed to find similarly expressed genes across many samples. When instances are samples (Figure 1b), three different tasks can be performed: *sample clustering,* which involves grouping similar samples together to discover classes or subclasses of samples; *sample class prediction,* which involves predicting diseases or phenotypes of novel samples based on patterns learned from training samples with known class labels; and *gene selection,* which involves selecting a small number of genes from thousands of genes to reduce the dimensionality of the data and improve the performance of classification and clustering methods.

Figure 1. Microarray data matrix

$$\begin{array}{ccccccc}
 & S_1 & S_2 & . & . & . & S_m \\
G_1 & f_{11} & f_{12} & . & . & . & f_{1m} \\
G_2 & f_{21} & f_{22} & . & . & . & f_{2m} \\
. & . & . & . & & & . \\
. & . & . & & & & . \\
. & . & . & & . & & . \\
. & . & . & & & & . \\
. & . & . & & . & . & . \\
G_n & f_{n1} & f_{n2} & . & . & . & f_{nm}
\end{array}$$

a

$$\begin{array}{cccccccccc}
 & G_1 & G_2 & . & . & . & . & . & G_n & \\
S_1 & f_{11} & f_{21} & . & . & . & . & . & f_{n1} & c_1 \\
S_2 & f_{12} & f_{22} & . & . & . & . & . & f_{n2} & c_2 \\
. & . & . & . & & & & & . & . \\
. & . & . & & & . & & & . & . \\
. & . & . & & & & & . & . & . \\
S_m & f_{1m} & f_{2m} & . & . & . & . & . & f_{nm} & c_m
\end{array}$$

b

MAJOR LINES OF RESEARCH AND DEVELOPMENT

In this part, we briefly review methods for each of the data-mining tasks identified earlier: gene clustering, sample clustering, sample class prediction, and gene selection. We discuss gene clustering and sample clustering together, for these two tasks are common; however, they are applied on microarray data from different directions.

Clustering

Clustering is a process of grouping similar samples, objects, or instances into clusters. Many clustering methods exist (for a review, see Jain, Murty, & Flynn, 1999; Parson, Ehtesham, & Liu, 2004). They can be applied to microarray data analysis for clustering genes or samples. In this article, we present three groups of frequently used clustering methods.

The first use of *hierarchical clustering* in gene clustering is first described in Eisen et al. (1998). Each instance forms a cluster in the beginning, and the two most similar clusters are merged until all instances are in one single cluster. The clustering results in the form of a tree structure, called *dendrogram,* which can be broken at different levels by using domain knowledge. Tree structures are easy to understand and can reveal close relationships among resulting clusters, but they do not provide a unique partition among all the instances, because different ways to determine a basic level in the dendrogram can result in different clustering results.

Unlike hierarchical clustering methods, *partition-based clustering* methods divide the whole data into a fixed number of clusters. Examples are *K*-means (Herwig, et at., 1999), self-organizing maps (Tamayo, et al., 1999), and graph-based partitioning (Xu, Y., Olman, & Xu, D., 2002). The methods of *K*-means often require specification of the number of clusters, *K,* and the selection of *K* instances as the initial clusters. All instances are then partitioned into the *K* clusters, optimizing some objective function (e.g., inner-cluster similarity) by assigning each instance to the most similar cluster, which is determined by the distance between the instance and the mean of each cluster in the current iteration. Self-organizing maps (SOMs) are variations of *K*-means methods and require specification of the initial topology of *K* nodes to construct the map. In graph-based partitioning methods, a Minimum Spanning Tree (MST) is often constructed, and the clusters are generated by deleting the MST edges with the largest lengths. Graph-based partitioning methods do not heavily depend on the regularity of the geometric shape of cluster boundaries, as *K*-means and SOMs do.

Traditional clustering methods require that each instance belong to a single cluster, even though some instances may be only slightly relevant for the biological significance of their assigned clusters. Fuzzy *C*-means (Dembele & Kastner, 2003) apply a fuzzy partitioning method that assigns cluster membership values to instances; this process is called *fuzzy clustering.* It links each instance to all clusters via a real-value vector of indexes. The value of each index lies between 0 and 1, where a value close to 1 indicates a strong association to the corresponding cluster, while a value close to 0 indicates no association. The vector of indexes thus defines the membership of an instance with respect to the various clusters.

Sample Class Prediction

Apart from clustering methods, which do not require a priori knowledge about the classes of available instances, a classification method requires training instances with labeled classes, leans patterns that discriminate be-