

A Maximum Entropy Approach to Classifying Gene Array Data Sets

Shumei Jiang, Chun Tang, Li Zhang and Aidong Zhang
Department of Computer Science and Engineering
The State University of New York at Buffalo
Buffalo, NY 14260

Murali Ramanathan
Department of Pharmaceutics
The State University of New York at Buffalo
Buffalo, NY 14260

Abstract

New technology such as DNA microarray can be used to determine simultaneously the expression levels of the thousands of genes which determine the function of all cells. Applying this technology to investigate the gene-level responses to different drug treatments could provide deep insight into the nature of many diseases as well as lead in the development of new drugs. In this paper, we present a maximum entropy approach to classifying gene array data sets. The experiments demonstrate the effectiveness of this approach.

1 Introduction

Recently, DNA microarray technology has been developed which permits rapid, large-scale screening for patterns of gene expression, as well as analysis of mutations in key genes associated with cancer [9, 5, 15, 16, 23, 11, 12, 2, 20]. To use the arrays, labelled cDNA is prepared from total messenger RNA (mRNA) of target cells or tissues, and is hybridized to the array; the amount of label bound is an approximate measure of the level of gene expression. Thus gene microarrays can give a simultaneous, semi-quantitative readout on the level of expression of thousands of genes. Just 4-6 such high-density “gene chips” could allow rapid scanning of the entire human library for genes which are induced or repressed under particular conditions. By preparing cDNA from cells or tissues at intervals following some stimulus, and exposing each to replicate microarrays, it is possible to determine the identity of genes responding to that stimulus, the time course of induction, and the degree of change.

Some methods have been developed using both standard cluster analysis and new innovative techniques to extract, analyze and visualize gene expression data generated from DNA microarrays. It has been found using yeast data [13] that by clustering gene expression data into groups, genes of similar function cluster together and redundant representations of genes cluster together. A similar tendency has been found in

humans. Data clustering [1] was used to identify patterns of gene expression in human mammary epithelial cells growing in culture and in primary human breast tumors. Clusters of coexpressed genes identified through manipulations of mammary epithelial cells in vitro also showed consistent patterns of variation in expression among breast tumor samples.

The generated clusters are used to summarize genome-wide expression and to initiate supervised clustering of genes into biologically meaningful groups [10]. In [4], the authors present a strategy for the analysis of large-scale quantitative gene-expression measurement data from time-course experiments. The approach takes advantage of cluster analysis and graphical visualization methods to reveal correlated patterns of gene expression from time series data. The coherence of these patterns suggests an order that conforms to a notion of shared pathways and control processes that can be experimentally verified.

The use of high-density DNA arrays to monitor gene expression at a genome-wide scale constitutes a fundamental advance in biology. In particular, the expression pattern of all genes in *Saccharomyces cerevisiae* can be interrogated using microarray analysis, in which cDNAs are hybridized to an array of each of the approximately 6000 genes in the yeast genome [14]. A key step in the analysis of gene expression data is the detection of groups that manifest similar expression patterns. The corresponding algorithmic problem is to cluster multicondition gene expression patterns. In [?], a novel clustering algorithm is introduced for analysis of gene expression data in which an appropriate stochastic error model on the input has been defined. It has been proven that under certain conditions of the model, the algorithm recovers the cluster structure with high probability.

Multiple sclerosis (MS) is a chronic, relapsing, inflammatory disease. *Interferon- β* (*IFN- β*) has been the most important treatment for the MS disease for last decade [22]. The DNA microarray technology makes it possible to study the expression levels of thousands of genes simultaneously. In this paper, we present a maximum entropy approach to classifying gene array data sets. In particular, we distinguish the healthy control, MS, IFN-treated patients based on the data collected from the DNA Array experiments. The gene expression levels are measured by the intensity levels of the corresponding array spots. The experiments demonstrate the effectiveness of this approach.

This paper is organized as follows. Section 2 introduces the maximum entropy model. Section 3, 4 and 5 describe the details of our approach on how to calculate features, probabilities and classification. Section 6 presents the experimental results. And finally, the conclusion is provided in Section 7.

2 Maximum Entropy Model

Entropy is a measure of uncertainty of random variable [8, 18]. It represents the amount of information required on average to describe the random variable. The entropy $H(X)$ of a discrete random variable X (

the set of which we'll call \mathcal{E}) is defined by

$$H(X) \stackrel{\text{def}}{=} - \sum_{x \in \mathcal{E}} p(x) \log p(x)$$

where $p(x)$ is the probability. The relative entropy $D(p||q)$ (known as Kullback-Leibler divergence) is defined as

$$D(p||q) \stackrel{\text{def}}{=} \sum_{x \in \mathcal{E}} p(x) \log \frac{p(x)}{q(x)}$$

it is a measure of the distance between two probability distributions. $D(p||q) \geq 0$ and the equality occurs if and only if $p(x) = q(x)$. The proof is simple.

In data classification, the goal is to classify the data from all known information. The *Principle of Maximum Entropy* [6] can be stated [18] as (1) Reformulate the different information sources as constraints to be satisfied by the target estimate. (2) Among all probability distributions that satisfy these constraints, choose the one that has the *highest* entropy. One way to represent the known information is to encode it as features and impose some constraints on the value of those feature expectations [17]. Here a feature is a binary-valued functions on events (or data) $f_j : \mathcal{E} \rightarrow 0, 1$. Given k features, the desired expectations can be formalized as

$$E_p f_j \stackrel{\text{def}}{=} \sum_{x \in \mathcal{E}} p(x) f_j(x) \quad j = 1, 2, \dots, k$$

They must satisfy the observed expectation i.e. constraints.

$$E_p f_j = E_{\tilde{p}} f_j \tag{1}$$

where \tilde{p} is the observed probability distribution in the training sample.

$$E_{\tilde{p}} f_j \stackrel{\text{def}}{=} \sum_{x \in \mathcal{E}} \tilde{p}(x) f_j(x) \quad j = 1, 2, \dots, k$$

The Principle of Maximum Entropy [6, 7, 17] recommends that we use $p^* = \arg \max_{p \in P} H(p)$ where $P = \{p \mid E_p f_j = E_{\tilde{p}} f_j, j = 1, 2, \dots, k\}$. It can be shown [17] that $D(p||p^*) = 0$ exists under the expectation constraints and p^* must have the form of

$$p^*(x) = A \prod_{j=1}^k \alpha_j^{f_j(x)}, 0 < \alpha_j < \infty \tag{2}$$

where A is a constant and the α_j 's are the model parameters. Each parameter α_j corresponds to exactly one feature f_j and can be viewed as a weight for that feature.

To find these weights, an iterative algorithm *Generalized Iterative Scaling* (GIS) is used, which is guaranteed to converge to the solution [3, 17]. [3] shows that $D(\tilde{p}||p^{(n+1)}) \leq D(\tilde{p}||p^{(n)})$ and $\lim_{n \rightarrow \infty} p^{(n)} = p^*$.

Here is the sketch of the procedure. In our approach, *Improved Iterative Scaling* (IIS) algorithm [19] is used.

$$\alpha_j^{(0)} = 1$$

$$\alpha_j^{(n+1)} = \alpha_j^{(n)} \left(\frac{E_{\tilde{p}} f_j}{E_{p^{(n)}} f_j} \right)^{\frac{1}{C}}$$

where

$$E_{p^{(n)}} f_j = \sum_{x \in \mathcal{E}} p^{(n)}(x) f_j(x)$$

$$p^{(n)}(x) = C \prod_{j=1}^{k+1} \left(\alpha_j^{(n)} \right)^{f_j(x)}$$

The maximum entropy model is simple and yet extremely general. It only imposes the constituent constraints without assuming anything else. The feature functions can represent the detailed information accurately. Using the maximum entropy, we can model very subtle dependencies among variables. This is important and useful, especially in high dimensions since all high dimensional data are detailed information. By defining feature functions, we make reasonable, unspurious assumptions of the data.

In our task of distinguishing healthy control people from MS patients, and MS patients from IFN treated patients, the information we have are the intensity values of about 4,000 genes for each identity. It is hard for human beings to look at the data and figure out the hidden pattern of each class. It is important to develop a reliable algorithm to perform the task. In the next two sections, we first define feature functions then apply IIS to find the weights for p^* . Finally, a classifier is built based on these weights.

3 Feature Definitions

The feature functions are very important in applying the maximum entropy theory. Bad features have no positive effects but causing noise and decreasing the classification precision. In the problem of classifying healthy control and MS patients, how to transfer the gene intensity values to feature functions requires biology knowledge. Although the absolute intensity changing values are important in classifying patients, we believe the relative change levels are more intrinsic. Also, different genes have different intensity value changing levels. The intensity change level alone by itself has no meaning. It varies with the the gene intensity changing level (denoted CL) for each patient and each gene. Our general formula is

$$CL_{gi} = (x_{gi} - \mu_g * t) / (\mu_g * t) \quad \text{where } \mu_g = \sum_{i \in S} CL_{gi} / |S| \quad (3)$$

where x_{gi} represents the intensity value for gene g of person i , t is a parameter, and μ_g is the mean of the intensity values for gene g for all patients S . Since the CL 's values are real numbers, we also bucket them into 21 predefined buckets CL^* by

$$CL_{gi}^* = \begin{cases} \lfloor CL_{gi} * 10 \rfloor & -1 < CL_{gi} < 1 \\ -10 & CL_{gi} \leq -1 \\ 10 & CL_{gi} \geq 1 \end{cases}$$

Different bucketing strategies affect the performance. One more definition is needed before we define the feature functions. We divide all patients into three data classes C : healthy control (HC), MS patients (MS) or IFN treated patients (IFN). Now for each gene g , each changing level bucket cl^* and each class c , we define a feature functions $f_{g,cl^*,c} : S \rightarrow 0, 1$ to be

$$f_{g,cl^*,c}(S) = \begin{cases} 1 & \text{if } \exists i \in S, \text{ such that } i \in c \text{ and for} \\ & \text{gene } g' \text{ of } i, g' = g, CL_{g'i}^* = cl^* \\ 0 & \text{otherwise} \end{cases}$$

Here we have multiple genes, multiple gene intensity value changing levels, and multiple groups, each combination of them makes up one feature function.

4 Probability

The ultimate goal is to classify all kinds of people into different classes. We can treat the intensity value of each gene as the context to decide the patient class. Here, class (C) has three values: HC , MS and IFN . *Context* is defined as $m = (g, cl^*)$ i.e. gene and its intensity value changing level bucket. Which class the patient belongs to depends on all the context information it has. In our situation, we adopt a probability model to describe it. If we can find the conditional probability $p(class|contexts)$ for each class, we can claim that the patient belongs to the class with the highest probability based on the context information.

$$p(class|contexts) = \frac{p(contexts, class)}{p(contexts)}$$

where

$$p(contexts, class) = \prod_j \alpha_j \quad j = (m, c) = (g, cl^*, c)$$

Weights α_j 's are governed by

$$p(m, c) = \frac{\prod_j \alpha_j^{f_j(S)}}{B} \quad m = (g, cl^*) \quad c \in C \quad (4)$$

and

$$B = \sum_{(m,e)} \prod_j \alpha_j^{f_j(S)}$$

Thus, B is a normalization constant, α_j 's are the model parameters. Compare the format of equation (2) and (4), f_j 's here are the feature functions we defined in the previous section with $j = (g, cl^*, c)$. According to maximum entropy model, we can apply IIS (Improved Iterative Scaling [19]) to calculate α_j 's. α_j 's are viewed as weights for f_j . We call this process the *training stage*.

There are two steps, the feature function induction and weight evaluation [19]. In the feature function induction step, when a single candidate feature function is introduced, we calculate the reduction of the Kullback-Leiber divergence by adjusting the weight of the candidate feature function while all the other parameters are kept constant. After one feature function is selected, all the weights of the selected feature functions are recalculated. IIS (Improved Iterative Scaling) algorithm is adopted to calculate the model parameters. The loop stops when the log-likely gain is less than the predefined threshold. The whole structure is shown in Figure 1.

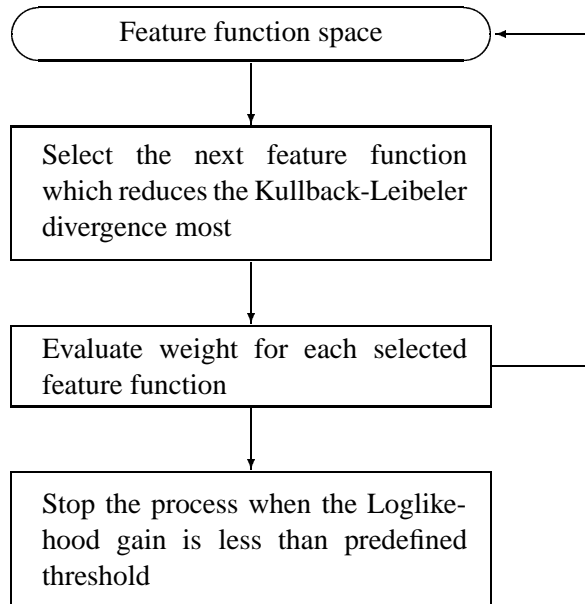


Figure 1: Training Structure.

5 Classification

In practice, among all the 4132 genes for each person, not all genes have the same contribution in distinguishing the classes. Actually, most of them have little contribution. We need to select some genes which are more important than others in solving the problem. To find those important genes, first, all genes are sorted by their degree of correlation, then the “neighborhood analysis” method is applied to extract the genes which are more correlated with the class distinction than other genes [21]. For all *HC*, *MS* and *IFN* data, we choose 88 genes for each identity.

After *training stage*, classification can be preformed easily. Given a patient *s*, we first calculate the gene intensity changing levels for all his genes, then construct the feature functions. From the training stage, we have weight α_j for all f_j of each class *c*.

We calculate $p(class|contexts)$ for all classes. Actually, only $\prod_j \alpha_j$ is necessary since all the denominators are the same. Higher α for a class indicates higher probability of the sample belong to that class. Finally we set the sample data to the class c^* such that $p(class|contexts)$ is the highest i.e.

$$c^* = \arg \max_{c \in C} \prod_j \alpha_j^{f_j(s)} \quad j = (g, cl^*, c)$$

The structure is shown in Figure 2.

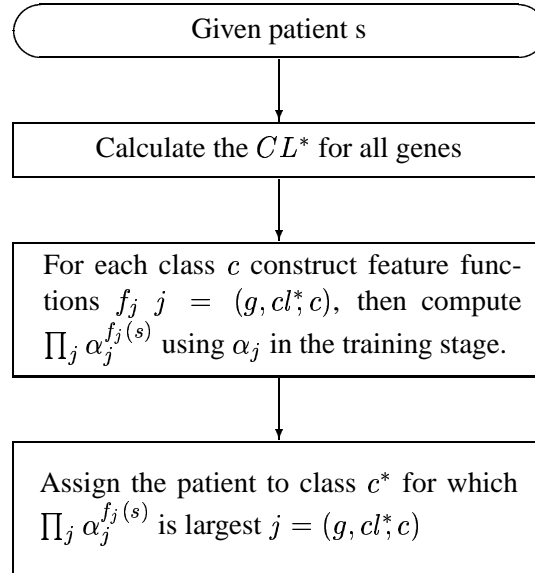


Figure 2: Classification Structure.

6 Experimental Results

The experiments are based on two different mix of the data sets: the MS_IFN group and the CONTROL_MS group. The MS_IFN group contains 14 MS samples and 14 IFN samples while the CONTROL_MS group contains 15 control samples and 15 MS samples. We perform the classification separately on each group.

For the MS_IFN group, in each experiment, we conduct 14 tests. In each test, we choose one different sample from the 14 MS samples and one different sample from the 14 IFN samples to make the test set, and use the other 26 samples as the training set. Thus each sample appears just once in the test set and the total number of samples we test is 28 which is the cardinality of the dataset.

Similarly, for the CONTROL_MS group, in each experiment, we conduct 15 tests. In each test, we choose one different sample from the 15 control samples and one different sample 15 MS samples correspondingly as the test set, and use the other 28 samples as the training set. The total number of samples we test is 30.

For each data set, we perform several experiments by adjusting the parameter t to calculate changing level CL in the formula Equation (3). In Table 1, we use the error classification number to evaluate the performance of our approach. We choose five different t values varying from 0.5 to 3 to perform five experiments on each data sets. As it can be observed from Table 1, different calculations of the changing level will affect the testing result.

Experiment#	1	2	3	4	5
Parameter t	0.5	1	1.5	2	3
Error# of MS_IFN(out of 28)	5	2	2	1	3
Error# of CONTROL_MS(out of 30)	7	8	6	12	12

Table 1: Experiment results.

7 Conclusion

In this paper, we have given a maximum entropy approach to classifying gene array data sets. In particular, we used the above approach to distinguish the healthy control, MS, IFN-treated patients based on the data collected from DNA Array experiments. To the best of our knowledge, the maximum entropy has not been used before to classify gene data. From our experiments, we demonstrated that the maximum entropy approach is a promising approach to be used for classifying gene array data sets.

8 References

- [1] Charles M. Perou, Stefanie S. Jeffrey, Matt Van De Rijn, Christia A. Rees, Michael B. Eisen, Douglas T. Ross, Alexander Pergamenschikov, Cheryl F. Williams, Shirley X. Zhu, Jeffrey C. F. Lee, Deval Lashkari, Dari Shalon, Patrick O. Brown, and David Bostein. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl. Acad. Sci. USA*, Vol. 96(16):9212–9217, August 1999.
- [2] D. Shalon, S.J. Smith, P.O. Brown. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research*, 6:639–645, 1996.
- [3] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals for Mathematical Statistics*, 43(5):1470–1480, 1972.
- [4] G.S. Michaels, D.B. Carr, M. Askenazi, S. Fuhrman, X. Wen and R. Somogyi. Cluster Analysis and data visualization of large-scale expression data. In *Pac Symposium of Biocomputing*, volume 3, pages 42–53, 1998.
- [5] J. DeRisi, L. Penland, P.O. Brown, M.L. Bittner, P.S. Meltzer, M. Ray, Y. Chen, Y.A. Su, J.M. Trent. Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nature Genetics*, 14:457–460, 1996.
- [6] E. T. Jaynes. Information theory and statistical mechanics. *Physics Reviews*, 106:620–630, 1957.
- [7] E. T. Jaynes. *Papers on Probability, Statistics, and Statistical Physics*. R. Rosenkrantz, ed., D. Reidel Publishing Co., Dordrecht-Holland, 1983.
- [8] F. Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, 1997.
- [9] J.J. Chen, R. Wu, P.C. Yang, J.Y. Huang, Y.P. Sher, M.H. Han, W.C. Kao, P.J. Lee, T.F. Chiu, F. Chang, Y.W. Chu, C.W. Wu, K. Peck. Profiling expression patterns and isolating differentially expressed genes by cDNA microarray system with colorimetry detection. *Genomics*, 51:313–324, 1998.
- [10] L.J. Heyer, S. Kruglyak and S. Yooseph. Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Res*, 1999.
- [11] M. Schena, D. Shalon, R.W. Davis, P.O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470, 1995.
- [12] Mark Schena, Dari Shalon, Renu Heller, Andrew Chai, Patrick O. Brown, and Ronald W. Davis. Parallel human genome analysis: Microarray-based expression monitoring of 1000 genes. *Proc. Natl. Acad. Sci. USA*, Vol. 93(20):10614–10619, October 1996.
- [13] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, Vol. 95:14863–14868, 1998.
- [14] M.Q. Zhang. Large-scale gene expression data analysis: a new challenge to computational biologists. *Genome Res*, 1999.
- [15] O. Ermolaeva, M. Rastogi, K.D. Pruitt, G.D. Schuler, M.L. Bittner, Y. Chen, R. Simon, P. Meltzer, J.M. Trent, M.S. Boguski. Data management and analysis for gene expression arrays. *Nature Genetics*, 20:19–23, 1998.
- [16] R.A. Heller, M. Schena, A. Chai, D. Shalon, T. Bedilion, J. Gilmore, D.E. Woolley, R.W. Davis. Discovery and analysis of inflammatory disease-related genes using cDNA microarrays. *Proc. Natl. Acad. Sci. USA*, 94:2150–2155, 1997.
- [17] A. Ratnaparkhi. A simple introduction to maximum entropy models for natural language processing, 1997.
- [18] R. Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, Carnegie Mellon University, 1994.
- [19] S. Pietra, V. Pietra, and J. Lafferty. Inducing Features of Random Fields. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 19(4):1–13, 1997.

- [20] S.M. Welford, J. Gregg, E. Chen, D. Garrison, P.H. Sorensen, C.T. Denny, S.F. Nelson. Detection of differentially expressed genes in primary tumor tissues using representational differences analysis coupled to microarray hybridization. *Nucleic Acids Research*, 26:3059–3065, 1998.
- [21] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gassenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, D.D. Bloomfield and E.S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, Vol. 286(15):531–537, October 1999.
- [22] V. Yong, S. Chabot, Q. Stuve and G. Williams. Interferon beta in the treatment of multiple sclerosis: mechanisms of action. *Neurology*, 51:682–689, 1998.
- [23] V.R. Iyer, M.B. Eisen, D.T. Ross, G. Schuler, T. Moore, J.C.F. Lee, J.M. Trent, L.M. Staudt, Jr. J. Hudson, M.S. Boguski, D. Lashkari, D. Shalon, D. Botstein, P.O. Brown. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, 1999.