

Feature selection and feature extraction with Gene Expression Data

H.C. Lam

November 7, 2006

Abstract

Gene expression data are driving computational resources to a limit that practitioners in the field of bioinformatics are looking for ways to reduce the size of the data to a meaningful group of features. These significant features are then used in the subsequent analysis steps to find biological significant genes. Reviews of different methodologies using various mathematical models are presented.

1 Introduction

Gene expression data analysis inherits a well known problem of "curse of dimensionality." This is due to the massive data collected during biological experiment with very few treatment variables against shockingly large number

of gene variables. We are essentially looking for a needle in a haystack when hunting for biologically significant genes in this given environment. Methods have been developed to circumvent this curse of dimensionality problem by applying unsupervised learning algorithm such as Principle Component Analysis or adopting statistical methods to filter genes down to a manageable size. In order to appreciate the difficulty this problem posts to gene expression data analysis, let's assume that we would like to analyze data points in a D dimensional space and would like to find out how much of each dimension is needed if we would only like to sample a proportion P of the total data volume in this D dimensional space. Let x be the amount needed from each dimension, then, $P = x^D$ and $x = P^{1/D}$. Hence, an exponential growth of each dimension is needed to sample P proportion of total data volume. In order to perform accurate data analysis without giving up vital information in the data, a certain amount of sample proportion is necessary. In other words, we need an exponential grow in data samples with respect to an increase in the dimensionality of the data volume to discover any valuable information hidden in the data. With gene data matrix, this can be translated as requiring the least number of genes from the data matrix for "good" analysis result but as the number of gene variables grow, the least number of required gene needed for "good" analysis result grows exponentially.

2 Background

There are two major methods in reducing the dimensionality of a gene data set, one is feature selection and the other is feature extraction. Most often classification of gene data is the eventual goal in using dimensional reduction technique to preprocess data before any classification algorithm is used. There exist supervised and unsupervised classifiers with linear or nonlinear scheme for dimensionality reduction. Linear schemes are usually for simpler data structure but when the data structure is complex, nonlinear schemes give more precise separating hypersurfaces. In Bioinformatics community, emphasis has been placed in applying machine learning methods for classification tasks rather than developing statistical tools for result verification. These areas of research are certainly important but are at the other end of the processing pipeline. The chain of gene data processing usually starts with a preprocessing procedure and unfortunately it is often done without paying close attention to its cause and effect. There can be different outcome if different assumption is made in advance.

In feature selection, the interest is to select a subset k of the d dimensions that provide the most valuable information and discard the $(d-k)$ dimensions of variables. In other words, the goal of feature selection is to find the best subset of the set of features. The best subset should contain the least number of features that account for the most information. In gene expression analysis, this method is usually carried out by selecting a subset or subsets

of genes that most represent the entire genes collection from the experiment. The selected subset are then used for subsequent analysis and any biological significant gene is found in the subset.

In feature extraction, the goal is to construct a new set of k dimensions from the original d dimensions with $k < d$. This new set of features is usually a linear or nonlinear combination of the original d dimensions. From a gene expression data analysis perspective, feature extraction translates to finding a new set of meta features that best explains the phenomena one observes in the experiment. This new set of meta features are often found in a subspace after projecting the original gene data to this lower dimensional space.

3 Feature Selection methods

Feature selection can be treated as finding individual feature and filtering each of the collected feature according to pre-defined criteria or it can be cast in a more subtle manner in which the aim is to find a subset or subsets of features that can be used to build good predictors. The main different is that the former methodology often result in collecting sub-optimal features. The subset selection method, however, often excludes redundant genes but retains relevant features within the set. This set can then be used as good training set for building class predictor.

3.1 Ranking of genes

Gene ranking methods have been used since the inception of microarray technology to solve the high dimension problem and to select genes with strong predictive power when classification of samples is needed. This baseline method is simple, scalable, and have shown empirical success. Besides numerous statistical methods available (which is not covered in the present paper), ranking gene can also be done by either select a fixed number of top rank features based on pre-defined parameters or by setting a threshold on the ranking criterion. Gene selection based on ranking is very common and it's a baseline method in gene data analysis. However, it makes a crucial assumption that each gene is independent of each other. This assumption can be devastating because we now know that genes are regulated together. The advantage of using a ranking scheme is that one can build a class predictor using selected genes. A class predictor is a decision function that has the forms:

$$(1) \quad f(x) > 0; x \Rightarrow class(+)$$

$$(2) \quad f(x) < 0; x \Rightarrow class(-)$$

and in general, a linear form of the decision function is:

$$(3) \quad f(x) = w^T \cdot x + b$$

with weight w and bias b and x being the new data to be classified.

3.1.1 Gene ranking based on correlation coefficients

In [13], the correlation coefficients used as ranking criteria is defined as:

$$(4) \quad g_i = [\alpha_i(+)-\alpha_i(-)]/[\sigma_i(+)+\sigma_i(-)]$$

where α_i and σ_i are the mean and standard deviation of the gene expression values of gene i for all the samples of class (+) and class (-) for $i = 1, \dots, n$ where n is number of genes in the data set. A large positive g_i value indicates a strong correlation with class (+) whereas a large negative g_i value indicates a strong correlation with class (-). The g_i selected can then be used as weight in equation (3) to perform classification task.

3.1.2 Gene ranking using recursive scheme

Recursively eliminate feature using a classifier to optimize the weights corresponding to the genes selected have been proposed by [14]. This method of trimming the data set recursively relies on the weight learned by the classifier. Gene is eliminated at each training step based on its weight produced by the classifier in terms of a cost function. The classifier is usually a linear support vector machine. There are a numbers of variations[12][38][24] of this type of schemes used in gene data preprocessing. The main different comes from the criterion to evaluate and select the most important genes through

the classifier. The ranking is created by adding the first removed gene to the bottom of the gene list at each step. The drawback of this approach is that one needs to use all the original features even though some features may be only background noise or noisy genes.

3.2 Gene pair selection

Gene pair ranking has been proposed by [5] whereas unlike the methods mentioned so far, a pair of genes is considered instead of a single gene is evaluated. A similar characteristic shared by this method with others methods is that it uses classifiers as the helper. A specific classifier called Diagonal Linear Discriminant (DLD) [26] is used for the prediction method. A gene pair is evaluated by using the two sample t-statistic as the pair score. The pair score is the projected points on the DLD axis. The prediction is determined by:

$$(5) \quad a^T[x - 1/2(\alpha_1 + \alpha_2)] > 0$$

where a is the DLD axis, α_1 and α_2 are the mean expression profile of class 1 and class 2 respectively. The DLD axis a is computed from the common variance estimate of the genes. The attractive property of this approach is that one is not only considering an individual gene like the above mentioned methods. The DLD axis is easy to compute and a visualization approach can be used to display the DLD axis with the genes selected.

3.3 Wrapper based gene selection

In wrapper approach, gene subset selection is performed in two steps. First step is to use a classifier to select genes based on the prediction. The classifier is usually a binary classifier, thus can only distinguish genes between two classes of samples. The second step is to apply an evaluation function to the selected genes. In [17][9][25][29][28], the accuracy (classification error rate) of the classifier is used as the evaluation function for the informativeness of genes. This method can be used regardless of the learning algorithm chosen and most of the off-the-shelf packages are used instead of implementing a specific algorithm to fit the problem. In essence, once a classifier is selected, one needs to search through the gene space for features to feed into the classifier. An exhaustive search can be performed, but this approach suffers from being computationally expensive because each set of evaluation requires the training of a classifier. Its advantage is that one can view the classifier as a black box and this black box can be obtained readily from any classification packages. One example is the Bioconductor [3] package which contains different classifiers ready to use.

4 Feature Extraction methods

In this section, meta feature construction methods are discussed. The popular and common feature extraction method called Principal Component Analysis (PCA) is first detailed. Multilayer perceptron for dimensionality

reduction is then presented. A kernel based PCA is also detailed and discussed.

4.1 Principal Component Analysis

Apart from visualization purpose, Principle Component Analysis (PCA)[18][11] has been widely used as a dimensionality reduction technique in gene data processing. PCA achieves its dimensionality reduction by projecting the original data to a reduced dimensional space where the variance of the original data is kept as much as possible. It is a linear transformation of the original features space. The transformation first rotates the original data space and then ranks the transformed features and selects the few features that best approximates the given dimensionality. A popular implementation of PCA is through the use of Singular Value Decomposition (SVD) method[19]. Let A denote an $m \times n$ matrix of real valued data that contains gene expression values. The row of A is the n dimensional vector that specifies the expression profile of a gene g . The column of A is the m dimensional vector that specifies the treatment condition profile. The Singular Value Decomposition of A has the form:

$$(6) \quad A = U\Sigma V^T$$

where U is $m \times m$ orthogonal matrix, Σ is $m \times n$ diagonal matrix and V^T is a $n \times n$ orthogonal matrix. The diagonal entries σ are called the singular

values of A and are usually ordered from high to low. These singular values are proportional to the variances of principle components. The columns u of U and v of V are the corresponding left and right singular vectors. SVD allows one to obtain a better estimate of the dimensionality of the data, which is the rank r of matrix A . The reduced dimensions can be represented by the matrix U when $r < n$ and the matrix V^T generally yields a representation of expression profiles with reduced number of variables. The principle components are chosen according to the singular values of Σ and discarding small σ , thus leading to a reduced representation of the original matrix A . Due to the noisy nature of gene expression data, the small σ values and its corresponding singular vectors are often identified as noise and ignored. PCA has been applied in [2][7][10][20] with promising results. A main excitement in using PCA/SVD for gene expression analysis is that one can plot the principle components in two dimensional space for visualization. The graphic plot can be used for spotting outliers or any clustering of genes. A gene ranking can also be done by only projecting raw genes data to the first principal component and then using a ranking criteria to perform gene ranking or selection. However, there are two drawbacks in using PCA for feature extraction. One is the interpretability of the principal components found and the other is the number of principal component to select. These two problems can sometimes be very difficult to solve and the solutions so far have been "ad hoc" at best.

4.2 Multilayer Perceptrons

Multilayer perceptrons (MLP) belongs to a class of Artificial Neural Networks (ANN) that has the ability to classify objects by training using the Backpropagation algorithm[31]. MLP with hidden layers can be used to perform dimensionality reduction. A simple MLP has one hidden layer, there are X_i input units, H_i hidden units, and Y_i output units. A successive hidden layers of processing units can be added to the network architecture with connections running from every unit in one layer to every unit in the next layer. In standard MLP, the output of the i th hidden unit is determined by forming a weighted sum of the input unit values and then applying a transfer function to the result.

$$(7) \quad a_i = \sum_{i=1}^d W^{(1)} x_i + b_1$$

where $W^{(1)}$ is the weight and b_1 is the bias. The transfer function can be a step function or a continuous sigmoidal function $f = 1/(1 + e^{-x})$ to give the output h_i .

$$(8) \quad h_i = f(a_i).$$

The network outputs are obtained by taking the output from the hidden layer units and construct a linear combination of h_i to form

$$(9) \quad a_k = \sum_{i=1}^N W^{(2)} h_i + b_2$$

where $W^{(2)}$ is the weight and b_2 is the bias. The summation is from 1 to N

where N is the number of hidden units. Then applying the transfer function to give

$$(10) \quad y_i = f(a_k).$$

Note that the transfer function does not have to be the same transfer function used in the preceding layer. The dimensionality reduction is achieved through the reduced number of units in the hidden layer. For non-linear dimensionality reduction purpose, usually three hidden layers are needed [4] [22].

Although not very common to train a MLP for dimensionality reduction purpose only, In [8], a non-linear encoder and decoder neural network based on autoassociation network[6] was used to perform low dimensional encoding of biological data. Others[22] [35] [36]had shown that a nonlinear representation of original data can be found in lower dimensional space with multiple hidden layers. In [23], a perceptron for feature selection is used. The selection is based on the weights produced by the network. There are three major factors one needs to consider before applying this approach for feature extraction. First, it is hard to rationalize what is actually being extracted from the hidden layer of the MLP. Second, it is a very expensive proposition due to the number of genes are in the tens of thousands. For each unit representing one gene and the network size is very fast to become a major problem for even a powerful computer. Third, the size of the hidden layers

needs to be specified in advance and this is hard to determine. Besides the drawback, a MLP is capable in finding nonlinear relationship among gene data and can find high predictive power genes[23].

4.3 Kernel based method

A kernel based[32] principal component analysis has been proposed for finding principal components in high dimensional feature space. This method is based on using a kernel function that computes the dot product of two input vectors in feature space without explicitly mapping the input vectors to the feature space. The kernel function chosen determines the characteristic of principal components extracted in feature space. Kernel-based PCA is based on the observation that the covariance matrix C of the given data can be diagonalized with nonnegative eigenvalues because C is positive definite.

$$(11) \quad C = 1/l \sum_{j=1}^l x_j x_j^T$$

where the covariance matrix of the centered data $x_j, j = 1, \dots, l, x_j \in R^N$ and the eigenvalue problem is

$$(12) \quad \lambda v = C v$$

where v is the eigenvector and λ is the corresponding eigenvalue and solving (12) for positive λ and $v \in R^N$. In order to use the kernel function to solve

the above problem we need to see that (12) is equivalent to (13) :

$$(13) \quad \lambda(x_j \cdot v) = (x_j \cdot v)$$

for all $j = 1, \dots, l$ and the corresponding computation of the dot product in feature space F via a mapping function Φ and rewrite equation (11) in terms of Φ .

$$(14) \quad C' = 1/l \sum_{j=1}^l \Phi(x_j) \Phi(x_j)^T$$

If the mapping function Φ is properly chosen, a kernel function k can be used to compute the inner product in F without performing explicit mapping where

$$(15) \quad k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$$

and the projections of data on the eigenvectors α can be found by

$$(16) \quad p = \sum_{i=1}^l \alpha_i k(x_i, x)$$

A benchmarking study[30] has been done using 7 gene data sets: colon cancer data[1], acute leukemia data[13], breast cancer data[16], hepatocellular carcinoma data[21], high grade glioma data[27], prostate cancer data[34], and breast cancer data[37] to assess the performance of kernel based PCA with RBF kernel function and linear kernel function. The author concluded that: (1) a linear kernel function is often better than a nonlinear one for feature selection. This is not surprising because Kernel PCA is "standard" PCA in

high dimensional space if a linear kernel is used. (2) The problem of overfitting (with RBF kernel) tends to occur when large number of component vectors are used for downstream analysis. This is also not surprising due to the fact that kernel PCA with nonlinear kernel is essentially solving the PCA problem with the gram matrix, which is far more bigger in size than the covariance matrix in which standard PCA is solving. This is also why kernel PCA can extract more components than standard PCA. Besides some of the drawbacks mentioned in section 4.1, kernel based PCA has another subtle but important drawback of finding the pre-image problem. This problem exists if one wish to use the feature selected to reconstruct the original gene data. Standard PCA mentioned in section 4.1 can reconstruct the original data with all its principal component vectors but this is not the case in kernel based PCA. Even though there are some drawbacks in kernel based PCA, its performance is almost always better than other feature extraction methods in terms of using the extracted features for classification tasks in the downstream processing steps[30].

5 Discussions and Conclusions

Most often researchers are interested in performing either feature selection or feature extraction as a data preprocessing step. The advantages of using such methods on gene data can be outlined as follows: (1) unnecessary input can be discarded thus saving memory space and speed up computation, (2)

data can be conveniently plotted and visually analyzed for structure and outliers if represented by only a few dimensions, (3) knowledge extraction can be done faster when data are represented with less features, (4) with smaller dataset, one can model the features more easily with lesser variation during analysis.

For gene selection, a study done by [33] tried to address the questions of: (1) Can the feature selection algorithm find a feature gene set that is close to the optimal feature gene set in terms of error produced from a classifier. (2) If the feature selection algorithm is not able to find a good feature set according to the error indication in (1), then should one expect that a good feature set does not exist. The outcome from [33] is that one cannot expect to find a feature set that is close to the optimal in terms of error alone. There needs to be prior biological information available to help the searching. In addition, the authors concluded that even though finding a good feature set is difficult, the existence of it cannot be ruled out.

Gene selection certainly helps in reducing computational burden while providing better gene predictor quality for downstream analysis. Feature extraction, on the other hand, has been seeing as a major contributing force in building a new frontier of metagene research due to the fact that this approach is still in its infancy and active research is still in progress. Nonetheless, using metagene one can eventually construct a gene network template for organisms to infer gene to gene interactions and activities. With the help

of gene expression databases around the world, one should be able to "dig" deeper into the universal truth about gene regulation by finding metagenes that can provides evident across different data sets. In this review, different mathematical models for performing gene selection and extraction are presented and their benefits along with the drawbacks are outlined. The dilemma in gene data preprocessing is that linear models are much simpler and straight forward in gene expression analysis but real life gene data set are very noisy and often has "nonlinear" relationship between variables in nature. This is not to say linear models suffer or lack strength, in fact, a good linear model often outperforms a nonlinear model in some specific problem setting. One should combine the strength of feature selection models with feature extraction models to build a model to generate metagenes from properly selected genes.

References

- [1] Alon, A., et al., Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon tissues Probed by Oligonucleotide Arrays, Proc. Natl. Acad. Sci. USA 1999, 96 6745-6750.
- [2] Alter O., et. al., Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci USA 2000, 97: 10101-06.
- [3] BioConductor open source software for bioinformatics, www.bioconductor.org
- [4] Bishop C. M., Neural Networks for Pattern Recognition. Oxford Press, 2000.

- [5] Bo T. H., and Jonassen I., New feature subset selection procedures for classification of expression profiles. *Genome Biology*. 2002.
- [6] Bourlard H., and Kamp Y., Auto-Association by Multilayer Perceptrons and Singular Value Decomposition. *Bio. Cybern.* 59, pp.291-294, 1988.
- [7] Cho R. J., et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell* 1998; 2:65-73.
- [8] DeMers D. and Cottrell G., Non-Linear Dimensionality Reduction. *Advances in Neural Information Processing System*. Vol. 5, pp.580-587, 1993
- [9] Deutsch J. M., et. al., Evolutionary Algorithms for Finding Optimal Gene Sets in Microarray Prediction. *Bioinformatics*, vol. 19, pp. 45-52, 2003
- [10] Eisen M. B., et al., Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA* 1998; 95:14863-68.
- [11] Fukunaga K., and Koontz W., Application of Karhunen-Loeve Expansion to Feature Selection and Ordering. *IEEE Trans. Comput.*, C-19, 311 (1970).
- [12] Furlanello C., et al. Entropy-based gene ranking without selection bias for the predictive classification of microarray data. *BMC Bioinformatics*. 4:54 2003
- [13] Golub, R.T., et al. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, Science, Vol 286. Oct 15. 1999.
- [14] Guyon et al., Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389-422, 2002.
- [15] Hastie T., Tibshirani R., Sherlock G., Eisen M., Brown P., and Botstein D., Imputing Missing Data for Gene Expression Arrays, Stanford University Statistics Department Technical report. 1999.
- [16] Hedenfalk I., et al., Gene Expression Profiles in Hereditary Breast Cancer, *The New England Journal of Medicine*, 344, 2001, pp 539-548.

- [17] Inza I., et al., Gene selection by Sequential Search Wrapper Approaches in Microarray Cancer Class Prediction. *Journal of Intelligent and Fuzzy Systems*, vol 12, pp. 25-34, 2002.
- [18] Jolliffe I. T., *Principal Component Analysis*. New York: Springer, 1986.
- [19] Heath M. T., *Scientific Computing. An introductory survey*. McGraw Hill, 2002.
- [20] Holter N. S., et al., Fundamental patterns underlying gene expression profiles: simplicity from complexity. *Proc Natl Acad Sci USA* 2000; 97:8409-14.
- [21] Iizuka N., et al., Oligonucleotide microarray for prediction of early intrahepatic recurrence of hepatocellular carcinoma after curative resection. *The Lancet*, 361, pp 923-929. 2003.
- [22] Kramer M., Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. *AIChE Journal*. Vol.37, No. 2, 1991.
- [23] Karzynski M., et al., Using a Genetic Algorithm and a Perceptron for feature selection and supervised class learning in DNA microarray Data. *Artificial Intelligence Review*. Vol 20, pp 39-51, 2003.
- [24] Li F, and Yang Y. Analysis of recursive gene selection approaches from microarray data. *Bioinformatics*, Vol 21, 19. pp. 3741-3747, 2005.
- [25] Li L, et al., Gene Selection for Sample Classification based on Gene Expression Data: Study of Sensitivity to Choice of Parameters of the GA/KNN Method. *Bioinformatics*, vol. 17, pp. 1131-1142, 2001
- [26] Mardia K.V., Kent J.T., and Bibby J.M., *Multivariate Analysis*. London: Academic Press, 1979.
- [27] Nutt C. L., et al., Gene expression based classification of malignant gliomas correlates better with survival than histological classification, *Cancer Research*, 63(7), 1602-1607, 2003.
- [28] Ooi C. H. and Tan P., Genetic Algorithms applied to Multi-class Prediction for the Analysis of Gene Expression Data. *Bioinformatics*, vol. 19, pp. 37-44, 2003

- [29] Peng S., et. al., Molecular Classification of Cancer Types from Microarray Data using the Combination of Genetic Algorithms and Support Vector Machines. *FSBS Letters*, vol. 555, pp. 358-362, 2003
- [30] Pochet N., et al., Systematic benchmarking of microarray data classification: assessing the role of nonlinearity and dimensionality reduction. *Bioinformatics*, July 1, 2004.
- [31] Rumelhart D. E., Hinton G. E., Williams R. J., Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and PDP Research Group (Eds), *Parallel Distributed Processing: Explorations in Microstructure of Cognition, Vol. 1: Foundations*, pp. 318-362. Cambridge, MA: MIT Press.
- [32] Scholkopf B., et al., Kernel Principal Component Analysis. Tech Report, No.44, Max-Planck Institut Fur biologische Kybernetik, 1996.
- [33] Sima C. and Dougherty E.R., What should be expected from feature selection in small-sample settings. *Bioinformatics*, July 26, 2006
- [34] Singh D., et al., Gene expression correlates of clinical prostate cancer behavior, *Cancer Cell*, 1(2), 203-209, 2002.
- [35] Oja E., "Data Compression, Feature Extraction, and Autoassociation in Feedforward Neural Networks" in Kohonen T., Simula O., and Kangas J., eds, *Artificial Neural Networks*, pp.737-745.
- [36] Usui S., Nakauchi S., and Nakano M., "Internal Color Representation Acquired by a Five-Layer Neural Network", in Kohonen T., Simula O., and Kangas J., eds, *Artificial Neural Networks*, pp.867-872
- [37] Van Veer L. J., et al., Gene expression profiling predicts clinical outcome of breast cancer, *Nature*, 415, 530-536, 2002.
- [38] Zhang X, et al. Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics*. Vol 7, 2006.