Supervised Learning from Micro-Array Data: Datamining with Care

Trevor Hastie  
Stanford University

joint work with Robert Tibshirani, Balasubramanian Narasimhan, Gil Chu, Pat Brown and David Botstein
Grade 9 view of cell biology

mRNA

mRNA Transcription

Mature mRNA

Nuclear membrane

Transport to cytoplasm for protein synthesis (translation)

Cell Membrane

DNA
DNA microarrays

- Exciting new technology for measuring gene expression of tens of thousands of genes SIMULTANEOUSLY in a single sample of cells
- first multivariate, quantitative way of measuring gene expression
- a key idea: to find genes, follow around messenger RNA
- also known as “gene chips” — there are a number of different technologies: Affymetrix, Incyte, Brown Lab,...
- techniques for analysis of microarray data are also applicable to SNP data, protein arrays, etc.
DNA microarray process

A Comparative Hybridization Experiment

1

2

3

4

5

6

Copyright © 1998-9 by Jeremy Buhler
The entire Yeast genome on a chip
Statistical challenges

- Typically have $\sim 5,000$–$40,000$ genes measured over $\sim 50$–$100$ samples.

- Goal: understand patterns in data, and look for genes that explain known features in the samples.

- Biologists don’t want to miss anything (low type II error). Statisticians have to help them to appreciate Type I error, and find ways to get a handle on it.
Types of problems

- Preprocessing: Li, Wong (Dchip), Speed

- The analysis of expression arrays can be separated into:
  - unsupervised — “class discovery” and
  - supervised — “class prediction”

- In unsupervised problems, only the expression data is available. Clustering techniques are popular. Hierarchical (Eisen’s TreeView - next slide), K-means, SOMs, block-clustering, gene-shaving (H&T), plaid models (Owen & Lazzeroni). SVD also useful.

- In supervised problems, a response measurement is available for each sample. For example, a survival time or cancer class.
Two-way hierarchical clustering
Some editorial comments

- Most statistical work in this area is being done by non-statisticians.
- Journals are filled with papers of the form “Application of <machine-learning method> to Microarray Data”
- Many are a waste of time. $P \gg N$ i.e. many more variables (genes) than samples. Data-mining research has produced exotic enhancements of standard statistical models for the $N \gg P$ situation (neural networks, boosting,...). Here we need to restrict the standard models; cannot even do linear regression.
- Simple is better: a complicated method is only worthwhile if it works significantly better than the simple one.
• Give scientists **good statistical software**, with methods they can understand. They know their science better than you. With your help, they can do a better job analyzing their data than you can do alone.

• Software should be easy to install (e.g. **R**) and easy to use (e.g. **SAM** and **PAM** are excel add-ins)
How to do 7000 t-tests all at once!

Significance Analysis of Microarrays (Tusher, Tibshirani and Chu, 2001)

- At what threshold should we call a gene significant?
- how many false positives can we expect?
SAM plot

$\text{expected } d(i)$

$\text{observed } d(i)$

-5 0 5

-5 0 5
Δ is the half-width of the band around the 45° line.

<table>
<thead>
<tr>
<th>Δ</th>
<th>Ave # falsely significant</th>
<th># Called significant</th>
<th>False discovery rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>75.1</td>
<td>294</td>
<td>.255</td>
</tr>
<tr>
<td>0.4</td>
<td>33.6</td>
<td>196</td>
<td>.171</td>
</tr>
<tr>
<td>0.5</td>
<td>19.8</td>
<td>160</td>
<td>.123</td>
</tr>
<tr>
<td>0.7</td>
<td>10.1</td>
<td>94</td>
<td>.107</td>
</tr>
<tr>
<td>1.0</td>
<td>4.0</td>
<td>46</td>
<td>.086</td>
</tr>
</tbody>
</table>
• SAM is popular in genetics and biochemistry labs at Stanford and worldwide already

• SAM is freely available for academic and non-profit use. The SAM site is:
  www-stat.stanford.edu/~tibs/SAM

• For commercial use, software is available for licensing from Stanford: kirsten.leute@stanford.edu
Nearest Prototype Classification

An extremely simple classifier that

- performs well on test data, and
- produces subsets of informative genes.

Classification of Samples

Example: small round blue cell tumors; Khan et al, Nature Medicine, 2001

- Tumors classified as BL (Burkitt lymphoma), EWS (Ewing), NB (neuroblastima) and RMS (rhabdomyosarcoma).

- There are 63 training samples and 25 test samples, although five of the latter were not SRBCTs. 2308 genes.

- Khan et al report zero training and test errors, using a complex neural network model. Decided that 96 genes were “important”.

- Upon close examination, network is linear. It’s essentially extracting linear principal components, and classifying in their subspace.
Khan data
Khan’s neural network
Centroids are shrunk towards the overall centroid using soft-thresholding. Classification is to the nearest shrunken centroid.
Nearest Shrunken Centroids

- Simple, includes nearest centroid classifier as a special case.
- Thresholding denoises large effects, and sets small ones to zero, thereby selecting genes.
- With more than two classes, method can select different genes, and different numbers of genes for each class.
- Still very simple. In statistical parlance, this is a restricted version of a naive Bayes classifier (also called idiot’s Bayes!)
Results on Khan data

At optimal point chosen by 10-fold CV, there are 27 active genes, 0 training and 0 test errors.
Predictions and Probabilities

Training Data

Test Data
The genes that matter

<table>
<thead>
<tr>
<th></th>
<th>BL</th>
<th>EWS</th>
<th>NB</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>24142</td>
<td>18337</td>
<td>212542</td>
<td>303163</td>
<td>21652</td>
</tr>
<tr>
<td>204545</td>
<td>504791</td>
<td>563673</td>
<td>207274</td>
<td>296446</td>
</tr>
<tr>
<td>784224</td>
<td>298082</td>
<td>796258</td>
<td>244618</td>
<td>44663</td>
</tr>
<tr>
<td>812105</td>
<td>325182</td>
<td>398231</td>
<td>41581</td>
<td>365536</td>
</tr>
<tr>
<td>629896</td>
<td>41581</td>
<td>365536</td>
<td>510057</td>
<td>377461</td>
</tr>
<tr>
<td>772094</td>
<td>814260</td>
<td>885702</td>
<td>295985</td>
<td></td>
</tr>
</tbody>
</table>
Heatmap of selected genes

PAM — “Prediction Analysis for Microarrays”

R package, available at

http://www-stat.stanford.edu/~tibs/PAM
Leukemia classification

Golub et al 1999, Science. They use a “voting” procedure for each gene, where votes are based on a t-like statistic

<table>
<thead>
<tr>
<th>Method</th>
<th>CV err</th>
<th>Test err</th>
</tr>
</thead>
<tbody>
<tr>
<td>Golub</td>
<td>3/38</td>
<td>4/34</td>
</tr>
<tr>
<td>PAM</td>
<td>1/38</td>
<td>2/34</td>
</tr>
</tbody>
</table>

Breast Cancer classification

Hedenfalk et al 2001, NEJM. They use a “compound predictor” $\sum_j w_j x_j$, where the weights are t-statistics.

<table>
<thead>
<tr>
<th>Method</th>
<th>BRCA1+</th>
<th>BRCA1-</th>
<th>BRCA2+</th>
<th>BRCA2-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heden et. al.</td>
<td>3/7</td>
<td>2/15</td>
<td>3/8</td>
<td>1/14</td>
</tr>
<tr>
<td>PAM</td>
<td>2/7</td>
<td>1/15</td>
<td>2/8</td>
<td>0/14</td>
</tr>
</tbody>
</table>
Other Promising Approaches

- Quadratically penalized linear regression models. For example, ridge regression
  \[
  \min_{\beta} \sum_{i=1}^{N} (y_i - x_i^T \beta) + \lambda \beta^T \beta
  \]
- Quadratically penalized logistic regression (binomial and multinomial), Cox model, etc
- Regularized linear and quadratic discriminant analysis
- Optimal separating hyperplanes and Support Vector Machines
- Lasso and \( L_1 \) penalized regression methods
  \[
  \min_{\beta} \sum_{i=1}^{N} L(y_i, x_i^T \beta) + \lambda |\beta|
  \]
Computational tricks

- SVMs use the kernel trick. Even though the model has $p \gg N$ parameters, computations done in $N$ dimensional dual space.
- In fact ANY of the quadratically regularized linear models can benefit from a similar trick:
  - Let $X = R \cdot Q$, with $R_{N \times N}$, $Q_{N \times p}$, and rows of $Q$ orthogonal.
  - Fit model with $R$ rather than $X$ ($N$ vs $p$ variables), same quadratic penalty; let solution be $\hat{\beta}^*$.
  - Then solution to original problem is $\hat{\beta} = QT \hat{\beta}^*$.
- This applies to regularized logistic regression, multinomial, any GLM, Cox model, and LDA. It also applies to Euclidean methods like K-nearest neighbor classification.
Example: Lasso and Leukemia Classification

![Graph showing LASSO with standardized coefficients and \(|\beta/\max|\beta|\) on the x-axis.](image)
10-fold Crossvalidation for Lasso

10-fold cross-validation for Leukemia Expression Data (Lasso)

See http://www-stat.stanford.edu/~hastie/Papers/#LARS
Summary

- With large numbers of genes as variables \((P \gg N)\), we have to learn how to tame even our simplest supervised learning techniques. Even linear models are way too aggressive.
- We need to talk with the biologists to learn their priors; i.e. genes work in groups.
- We need to provide them with easy-to-use software to try out our ideas, and involve them in the design. They are much better at understanding their data than we are.