Supervised Harvesting of Expression Trees

Trevor Hastie
Robert Tibshirani
David Botstein
Patrick Brown

Introduction

Setting: Have many samples of expression data, where each sample consists of

(1) Real-valued expression measurements for thousands of genes

(2) A "response" measurement (e.g., survival time, cancer class)

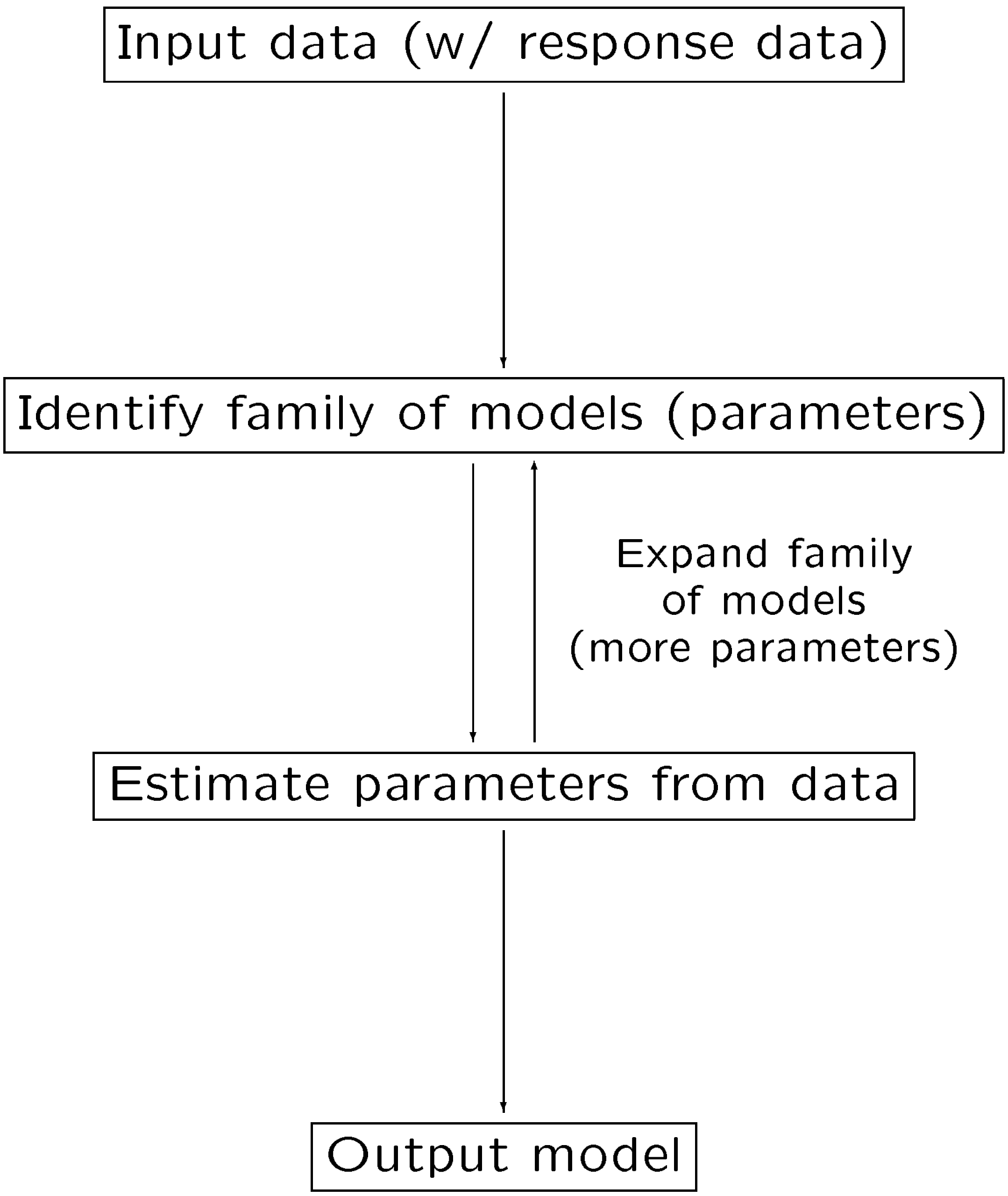
Previously: Methods to cluster genes (e.g. hierarchical, gene shaving)

Today's question: How is response related to genes and gene clusters?

Goal: Develop a "response model" fitting the data, where

- Input is expression levels of genes (and gene clusters) for a sample
- Output is a prediction of the response

Modelling


Input data (w/ response data)
Identify family of models (parameters)
Expand family of models (more parameters)
Estimate parameters from data
Output model

Modelling

- Model family depends on response type
- Parameters estimated by minimizing a "loss function"

Response	**Model**	**Loss function**
Quantitative	Multiple regression	Sum of squares
Categorical	Multiple logistic regression	Multinomial log-likelihood
Censored survival time	Proportional hazard model	Partial log-likelihood

Typical models and loss functions

Tree Harvesting

Original data: For each sample $j = 1, \ldots, n$, a response measure y_j and gene expression data x_{ij} for genes $i = 1, \ldots, p$.

Hierarchical clustering data: Have $2p - 1$ clusters $c_1, c_2, \ldots, c_{2p-1}$. For each sample $j = 1, \ldots, n$, have expression vector

$$X_j = (\overline{x}_{c_1,j}, \ldots, \overline{x}_{c_{2p-1},j})^t$$

where $\overline{x}_{c_k,j}$ is (translated) average expression level of genes in cluster k for sample j.

Model family (quantitatitve response): Given a generic sample $X = (\overline{x}_{c_1}, \ldots, \overline{x}_{c_{2p-1}})^t$,

$$\widehat{y}(X) = \beta_0 + \sum_k \beta_k \overline{x}_{c_k} + \sum_{k,k'} \beta_{k,k'} \overline{x}_{c_k} \overline{x}_{c'_k} + \cdots$$

Estimate parameters: Get β's by minimizing

$$\sum_{j=1}^{n} (\widehat{y}(X_j) - y_j)^2$$

Tree Harvesting

Current model space: Too many parameters (inefficiency and overfitting problems)

$$\hat{y}(X) = \beta_0 + \sum_k \beta_k \overline{x}_{c_k} + \sum_{k,k'} \beta_{k,k'} \overline{x}_{c_k} \overline{x}_{c'_k} + \cdots$$

($p = 5000$, $n = 50$ means 10000 linear terms. Can usually fit data exactly using 50 terms.)

Better: Search a restricted model space,

- Limit number N of terms in model.
- Limit order I of interaction (degree of terms) in model.
- $\boxed{\text{Select terms one-by-one to build up model.}}$

Typical: $p = 5000$, $n = 50$, $N = 5$, $I = 2$

Similar to clustering? Is selecting terms (genes) like clustering genes around response variable?

Building the model

1. Start with model $M = \{1\}$ (corresponds to β_0).

2. Fit the current model M (calculate β's)

3. For each term m in M and each cluster variable $\overline{x}_{c_k}$, consider the new term $m \cdot \overline{x}_{c_k}$. Of these, augment M by term which most improves the model based on a score statistic S (or term with largest cluster whose score is at least $(1-\alpha)S$, e.g. $\alpha = .9$).

4. Repeat (2.) and (3.) until N terms added.

5. Select best model size by backward deletion and cross-validation.

Other features: Bias towards larger clusters, score statistic, cross-validation

Remark: For modelling quantitative response, first iteration identifies cluster c_k giving smallest $\sum_j (y_j - \beta_0 - \beta_k \overline{x}_{c_k,j})^2$ for some β_0, β_k. Like maximizing correlation?

Example: Human Tumor Data

Data from human cancer cell lines: $n = 61$ samples, $p = 6830$ genes

Categorical response: 8 cancer classes y

Cancer y	Samples j	Total
1. breast	1-9	9
2. CNS	10-14	5
3. colon	15-21	7
4. leukemia	22-29	8
5. melanoma	30-37	8
6. nsclc	38-46	9
7. ovarian	47-52	6
8. renal	53-61	9

Example

Model family: Linear MLR (multiple logistic regression) – probabilities of response $\widehat{y}$ given data $X = (\overline{x}_{c_1}, \ldots \overline{x}_{c_{2p-1}})^t$,

$$
\begin{aligned}
P(\widehat{y}(X) = 1|X) &= \frac{\exp(r_1(X))}{1+\sum_{m=1}^{7} \exp(r_m(X))} \\
&\vdots \\
P(\widehat{y}(X) = 7|X) &= \frac{\exp(r_7(X))}{1+\sum_{m=1}^{7} \exp(r_m(X))} \\
P(\widehat{y}(X) = 8|X) &= \frac{1}{1+\sum_{m=1}^{7} \exp(r_m(X))}
\end{aligned}
$$

where

$$
\begin{aligned}
r_1(X) &= \beta_{m,0} + \textstyle\sum_k \beta_{m,k}\overline{x}_{c_k} + \sum_{k,k'} \beta_{m,k,k'}\overline{x}_{c_k}\overline{x}_{c'_k} \\
&\vdots \\
r_7(X) &= \beta_{m,0} + \textstyle\sum_k \beta_{m,k}\overline{x}_{c_k} + \sum_{k,k'} \beta_{m,k,k'}\overline{x}_{c_k}\overline{x}_{c'_k}
\end{aligned}
$$

Adding 1 new term $\overline{x}_{c_k}$ or $\overline{x}_{c_k}\overline{x}_{c'_k}$ introduces 7 new parameters β.

Example

Probability of data D based on model M:

$$P(D|M) = \begin{pmatrix} \prod_{j=1}^{9} P(\widehat{y}(X) = 1|X = X_j)\cdot \\ \prod_{j=10}^{14} P(\widehat{y}(X) = 2|X = X_j)\cdot \\ \vdots \\ \prod_{j=53}^{61} P(\widehat{y}(X) = 8|X = X_j) \end{pmatrix}$$

Loss function: Multinomial log-likelihood

$$\ell = -2\log(P(D|M))$$

Estimate parameters: Minimize ℓ by setting partial derivatives to 0,

$$\frac{\partial \ell}{\partial \beta} = 0, \quad \beta \text{ in } M$$

Solve by Newton's method to get approx β^*.

Score statistic: The model with β_k added is evaluated by

$$S = -\left(\frac{\partial \ell / \partial \beta_k}{\partial^2 \ell / \partial \beta_k^2}\right)(\beta^*, 0)$$

Example

	Node	Parent	Score	-2log likelihood	Size
1	1177	0	6.48	197.53	6
2	3843	0	1.97	132.34	4
3	2008	0	1.78	79.34	3
4	1665	3843	0.85	71.01	3
5	5009	0	0.69	51.91	68
6	5087	2008	0.59	9.32	9
7	820	3843	0.55	0.00	2

7 terms: 56 parameters vs. 61 samples

Cross-validation (CV)

Backwards deletion

- Discard one-by-one the term causing smallest increase in the loss function (set corresponding β's to 0 and evaluate) until model has only constant term.
- Get a model M_i having i terms for each $i = 7, 6, \ldots, 1$.

10-fold CV: Want to select best model M_i

- Split original cancer data into 10 parts.
- Fix a model M_i. For each $k = 1, 2, \ldots, 10$, estimate parameters of M_i from all data except k-th part, & evaluate performance by predicting data of k-th part. Average over $k = 1, \ldots, 10$ to get total performance of M_i.
- Select M_i with best total performance.

Final thoughts

Also in the paper:

- Example: survival of Diffuse Large Cell Lympoma patients ($p = 3624$ genes, $n = 36$ patients)
- Simulation experiment to assess ability to detect additive and interaction structure.
- Nonlinear models

Conclusions and future:

- Potential tool for identifying interesting clusters of genes with respect to a response.
- Larger number of samples probably required to successfully uncover structure.