Classification of Gene Microarrays by Penalized Logistic Regression

Ji Zhu
Trevor Hastie
Department of Statistics
Stanford University
Email:jzhu@stanford.edu

Classification problem

- Training data: matrix $X_{p\times n}$
- p number of genes, n number of samples (arrays).
- x_{ji} be the expression for gene j and sample i. $j = 1, 2 \ldots p$, $i = 1, 2 \ldots n$.
- Classes $1, 2, \ldots K$. c_i assumes value in $\{1, 2, \ldots K\}$ and indicates the class of sample i.
- Wish to find a classification model:

$$\vec{x} \in \mathcal{R}^p \to c \in \{1, 2, \ldots K\}$$

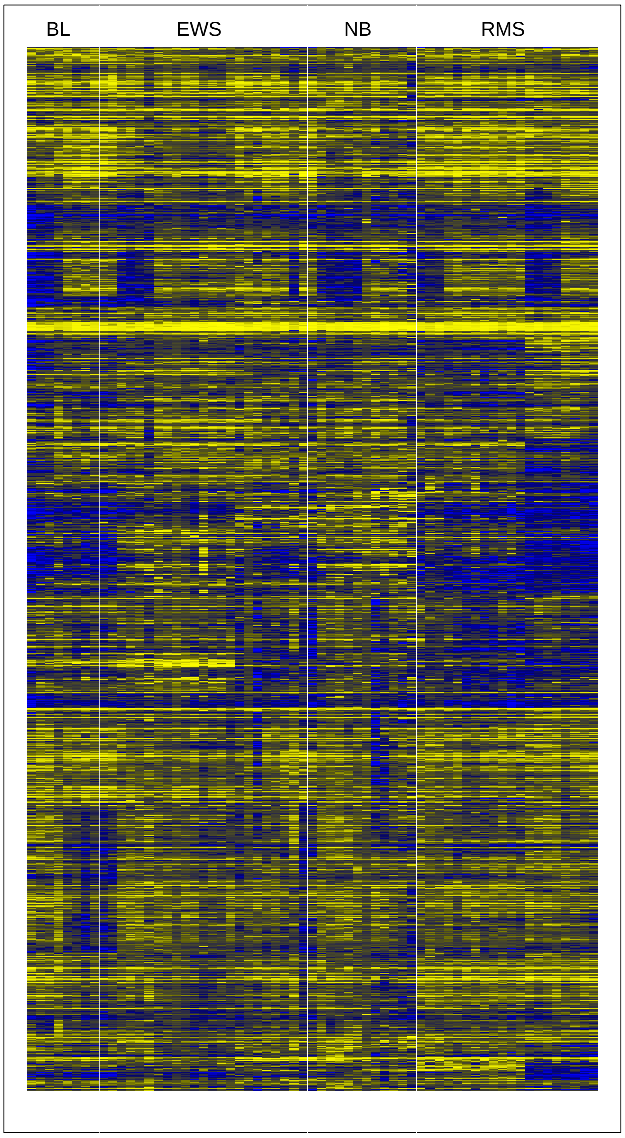
- Construction and evaluation of the model: cross-validation and test data.
- Gene selection.

Example: Khan data

Small round blue cell tumors, Khan et al. Nature Medicine 2001.

- Tumors classified as BL (Burkitt lymphoma), EWS (Ewing), NB (neuroblastima) and RMS (rhabdomyosarcoma). $K = 4$.
- There are $n = 63$ training samples and 25 test samples, although five of the latter were not SRBCTs. $p = 2308$ genes.
- They report zero test errors with 96 genes.
- Complicated: bootstrapped neural network.

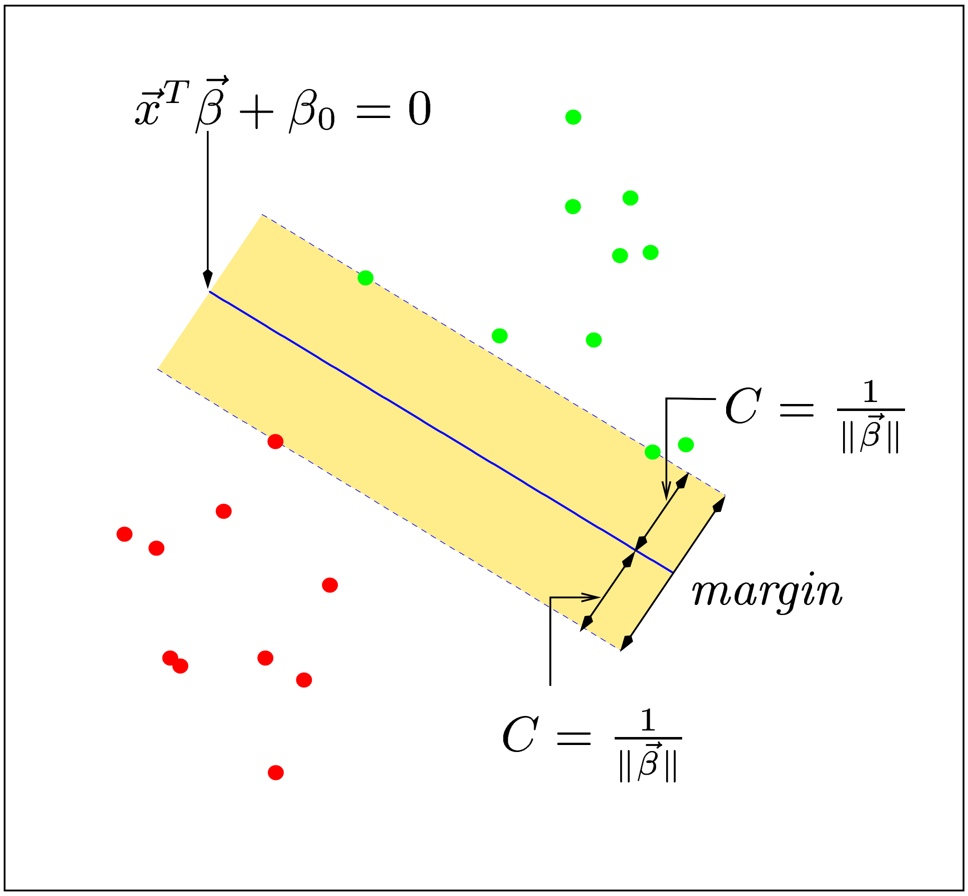
Khan data



Outline

- Linear support vector machine (SVM)
- Penalized logistic regression (PLR)
- PLR vs SVM:
  similar classification performance
  estimate probability
- Gene selection:
  univariate ranking (UR)
  recursive feature elimination (RFE)
- Results on three data sets

Maximum margin classifier: two class

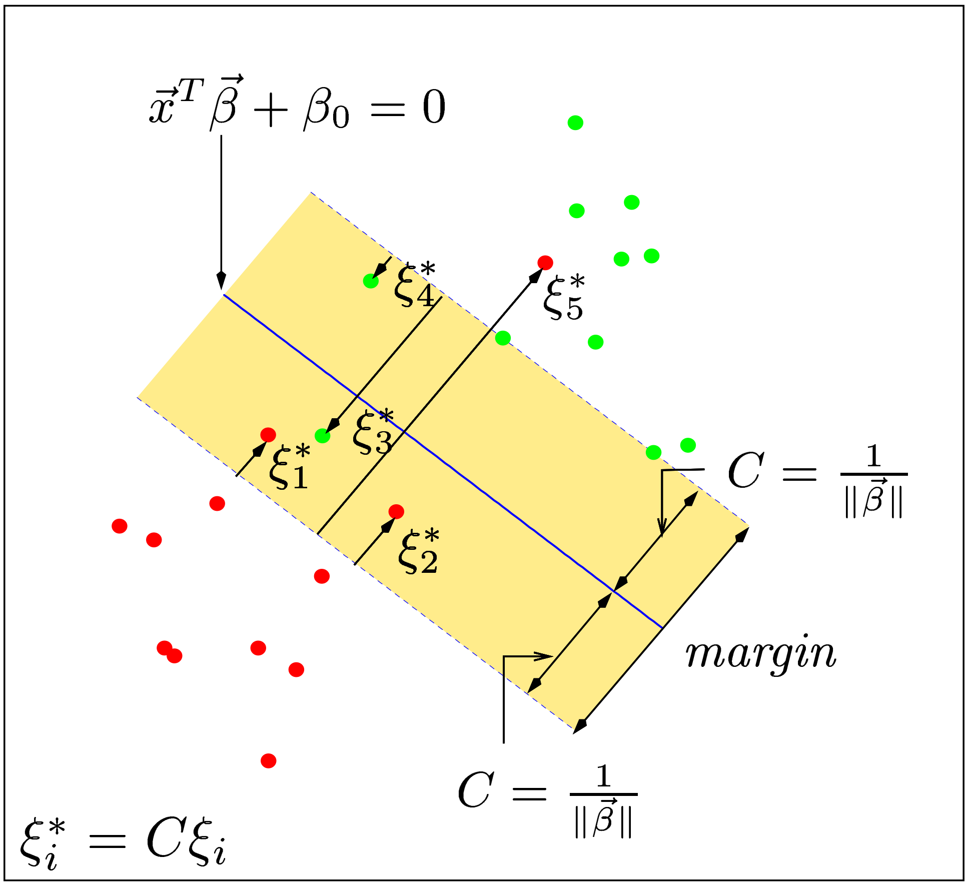


Vapnik(1995) $\quad \vec{x}_i \in \mathcal{R}^p,\ y_i \in \{-1, 1\}$

$$\max_{\vec{\beta},\beta_0,||\vec{\beta}||=1} C$$

subject to $\quad y_i(\vec{x}_i^T\vec{\beta} + \beta_0) \geq C,\ i = 1, \ldots, n.$

Overlapping classes



$$\max_{\vec{\beta},\beta_0,\|\vec{\beta}\|=1} C$$

subject to

$$y_i(\vec{x}_i^T\vec{\beta} + \beta_0) \geq C(1 - \xi_i), \;\; \xi_i \geq 0, \;\; \sum_i \xi_i \leq B$$

Solution

The solution is expressed in terms of the fitted Lagrange multipliers $\hat{\alpha}_i$:

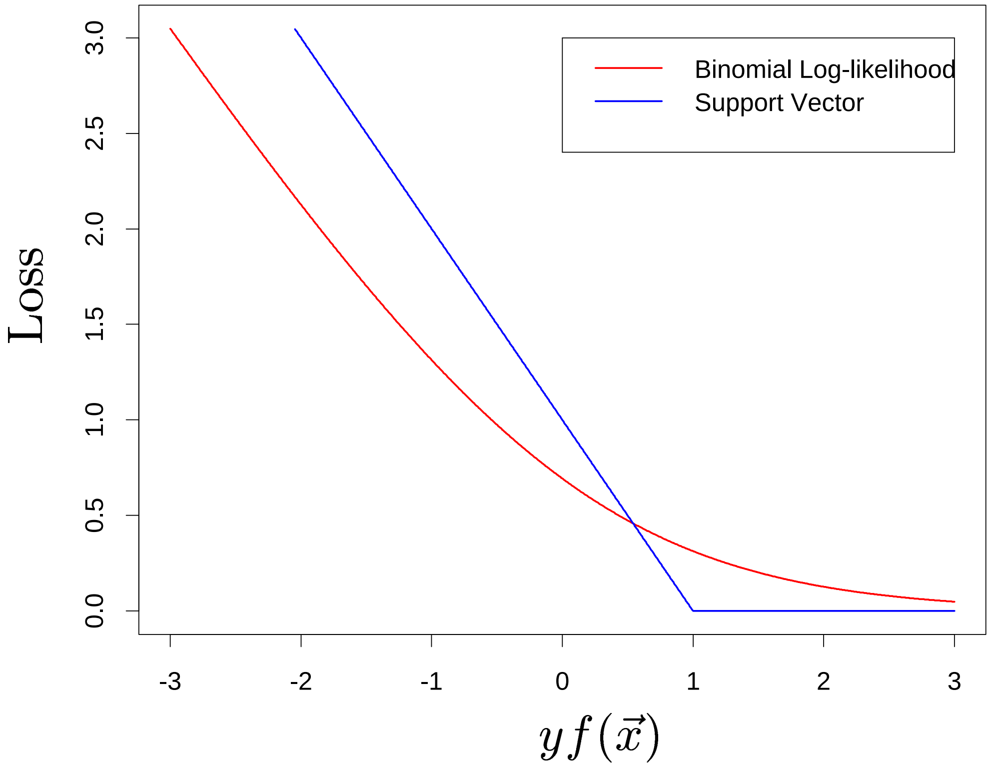
$$\hat{\vec{\beta}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \vec{x}_i$$

Some fraction of $\hat{\alpha}_i$ are exactly zero; the others are called support points S.

$$\begin{aligned} \hat{f}(\vec{x}) &= \vec{x}^T \hat{\vec{\beta}} + \hat{\beta}^0 \\ &= \sum_{i \in \mathcal{S}} \hat{\alpha}_i y_i \vec{x}^T \vec{x}_i + \hat{\beta}^0 \end{aligned}$$

Classification rule: $sign\left(\hat{f}(\vec{x})\right)$.

Linear SVM and function estimation



With $f(\vec{x}) = \vec{x}^T\vec{\beta} + \beta_0$, consider

$$\min_{\beta_0,\,\vec{\beta}} \sum_{i=1}^{n} [1 - y_i f(\vec{x}_i)]_+ + \frac{\lambda}{2}\|\vec{\beta}\|^2$$

Solution identical to SVM solution.

$$\text{In general } \min_{\beta_0,\,\vec{\beta}} \sum_{i=1}^{n} L[y_i, f(\vec{x}_i)] + \frac{\lambda}{2}\|\vec{\beta}\|^2$$

Loss functions

For $y \in \{-1, 1\}$

SVM: $L[y, f(\vec{x})] = (1 - yf(\vec{x}))_+$

- Called "hinge loss"
- Estimates the classifier (threshold)

$$\operatorname{sign}\left(\Pr(y = 1|\vec{x}) - \frac{1}{2}\right)$$

Log-likelihood: $L[y, f(\vec{x})] = \log\left(1 + e^{-yf(\vec{x})}\right)$

- (Negative) binomial log-likelihood or deviance.
- Estimates the logit

$$f(\vec{x}) = \log \frac{\Pr(y = 1|\vec{x})}{\Pr(y = -1|\vec{x})}$$

Comparison: SVM vs PLR

- Classification performance similar.
- Has limiting optimal margin properties.
- Provides estimates of the class probabilities.
- Computationally more expensive $O(n^3)$ vs $O(n^2m)$, where m is the number of support vectors. Fast algorithm for PLR exists.
- Multi-class (next slide).

Multi-class classification

- SVM usually uses the one-vs-rest scheme: fit K separate two-class models and label the sample with the mostly likely class.
- "Masking" effect for one-vs-rest scheme.
- Multi-class PLR is more natural.
- Lee *et al.* 2002 directly estimate $argmax_k \mathrm{Pr}(C = k|\vec{x})$, but still lacks probability estimates.

Multi-class logistic regression

- Wish to model the conditional probabilities of the K classes via linear functions in $\vec{x}$.
- Form:

$$p_k(\vec{x}) = \frac{e^{f_k(\vec{x})}}{\sum_{k'=1}^{K} e^{f_{k'}(\vec{x})}},$$

where

$$f_k(\vec{x}) = b_{k0} + \vec{b}_k^T \vec{x}.$$

- $f_k(\vec{x})$ unidentifiable. Symmetric constraint

$$\sum_{k=1}^{K} b_{k0} = 0$$

$$\sum_{k=1}^{K} \vec{b}_k = 0.$$

Multinomial log-likelihood

$$
\begin{aligned}
& -\sum_{i=1}^{n} \log p_{c_i}(\vec{x}_i) \\
= \quad & \sum_{k=1}^{K} \sum_{c_i=k} [-f_k(\vec{x}_i) + \ln(e^{f_1(\vec{x}_i)} + \cdots + e^{f_K(\vec{x}_i)})].
\end{aligned}
$$

$p \gg n$ implies:

- The classes are usually linearly separable.
- The log-likelihood achieves a maximum of 0.
- The parameters $\vec{b}_k$'s are not uniquely defined, and indeed many will be infinite.

Penalized log-likelihood

$$
\begin{aligned}
H &= \sum_{i=1}^{n} \sum_{c_i=k} [-f_k(\vec{x}_i) + \ln(e^{f_1(\vec{x}_i)} + \cdots e^{f_K(\vec{x}_i)})] \\
&+ \frac{\lambda}{2} \sum_{k=1}^{K} \|\vec{b}_k\|^2,
\end{aligned}
$$

- Constraint $\sum_{k=1}^{K} \vec{b}_k = 0$ not necessary anymore.
- p is very large, minimizing H directly is computationally prohibitive.

Reduce computation

- Inner product

$$\langle \vec{x}, \vec{x}' \rangle = \sum_{j=1}^{p} x_j x_j'.$$

- Solution (Kimeldorf & Wahba 1971)

$$f_k(\vec{x}) = b_{k0} + \sum_{i=1}^{n} a_{ik} \langle \vec{x}, \vec{x}_i \rangle,$$

and

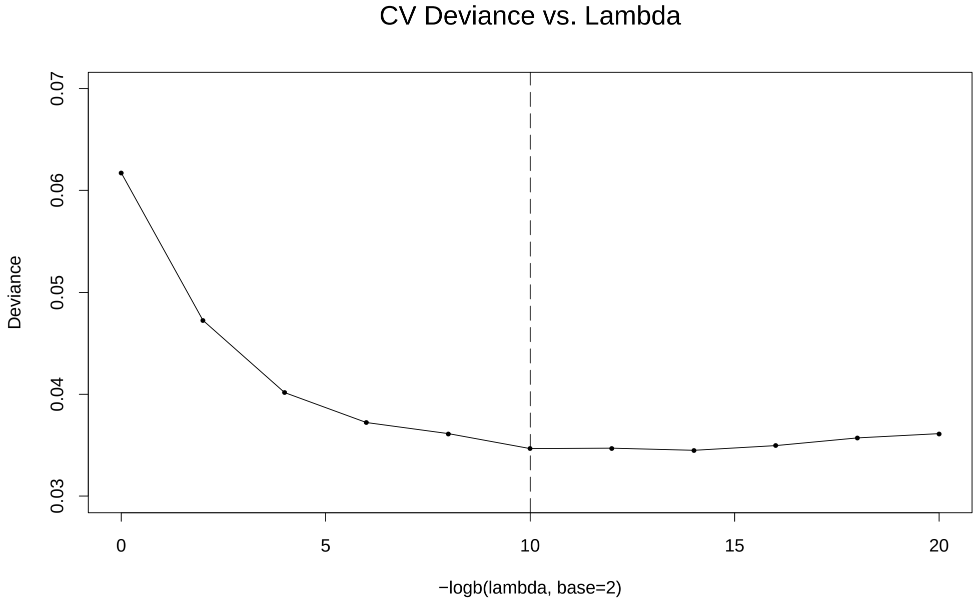
$$\vec{b}_k = \sum_{i=1}^{n} a_{ik} \vec{x}_i.$$

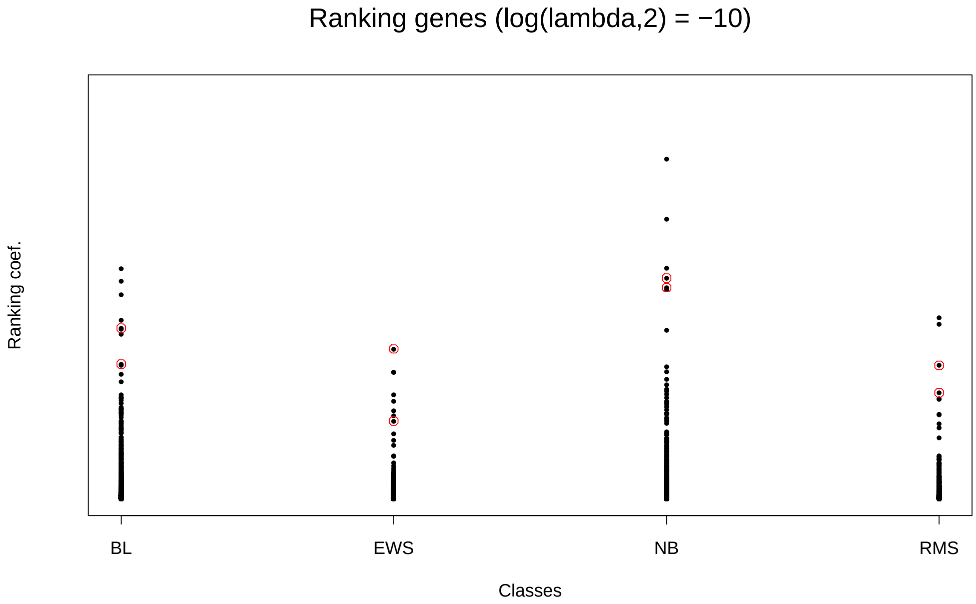
- Penalized negative log-likelihood

$$\begin{aligned} H &= \sum_{i=1}^{n} \sum_{c_i=k} [-f_k(\vec{x}_i) + \ln(\sum_{k'=1}^{K} e^{f_{k'}(\vec{x}_i)})] \\ &+ \frac{\lambda}{2} \sum_{k=1}^{K} \sum_{i,i'} a_{ik} a_{i'k} \langle \vec{x}_i, \vec{x}_{i'} \rangle. \end{aligned}$$

- Number of parameters $(p+1)K \to (n+1)K$.

Choosing λ





Results W/O gene selection

SRBCT data (Kahn *et al.* 2001)

$K = 4, p = 2308$

Method	CV	Test
SVM	0/63	0/20
PLR	1/63	0/20

Two other data sets

Leukemia data (Golub *et al.* 1999)

$K = 2, p = 7129$

Method	CV	Test
SVM	1/38	2/34
PLR	1/38	0/34

Ramas data (Ramaswamy *et al.* 2001)

$K = 14, p = 16063$

Method	CV	Test
SVM	28/144	14/54
PLR	29/144	15/54

Gene selection: UR

- Univariate ranking.
- Two-class (Golub *et al.* 1999)

$$\rho_j = \left| \frac{\bar{x}_j^{(1)} - \bar{x}_j^{(2)}}{\sigma_j^{(1)} + \sigma_j^{(2)}} \right|.$$

- Multi-class (Dudoit *et al.* 2002)

$$\rho_j = \left| \frac{\sum_{k=1}^{K} n_k (\bar{x}_j^{(k)} - \bar{x}_j)^2}{(n-K)\sigma_j^2} \right|.$$

- Shrunken centroids method (Tibshirani *et al.* 2002).
- All implicitly assume orthogonality among the genes, because each ρ_j is computed with information about a single gene and does not take into account mutual information between genes.

Gene selection: RFE

- Recursive feature (gene) elimination (Guyon *et al.* 2002).
- At each step, fit a classifier with the current genes.
- A ranking criterion is then computed for each gene. Eg.

$$\rho_j = \frac{\partial^2 H}{\partial b_j^2} b_j^2 .$$

- Genes with the smallest ρ_j is removed. Iterate.
- SVM $\rho_j = b_j^2$.
  PLR ρ_j more complicated - similar to t^2 statistic.

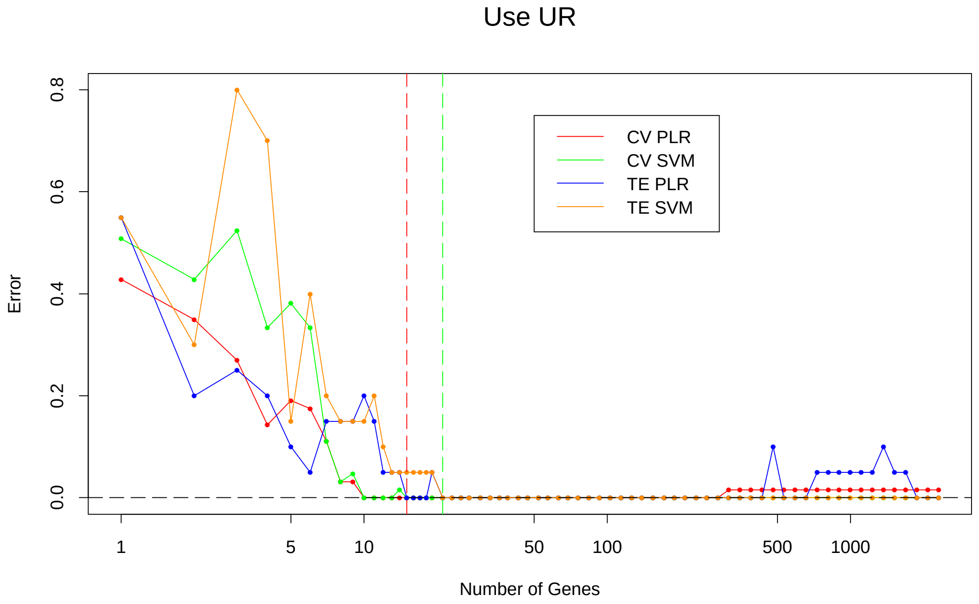
Khan data

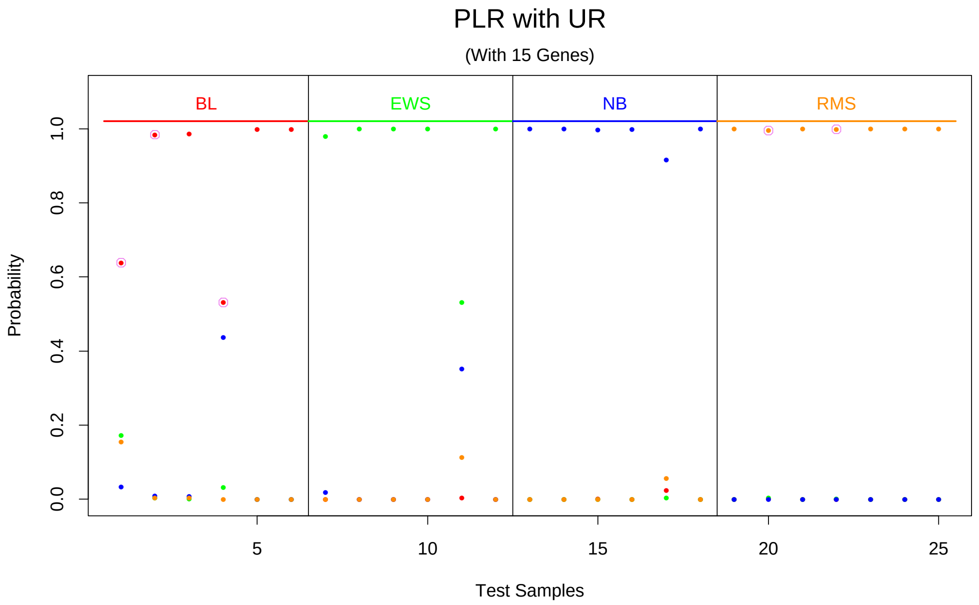
SRBCT 4-class (Khan *et al.* 2001)

Method	CV	Test	Genes
SVM UR	0/63	0/20	21
PLR UR	0/63	0/20	15
SVM RFE	0/63	0/20	32
PLR RFE	0/63	0/20	8

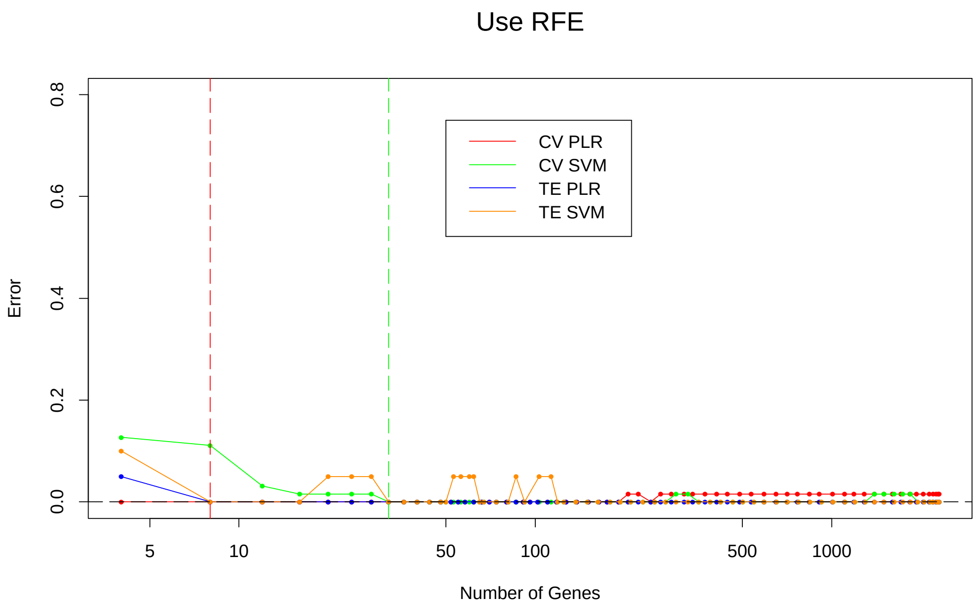
Method	CV	Test	Genes
Khan *et al.* (2001)	0/63	0/20	96
Tibs *et al.* (2002)	0/63	0/20	43
PLR RFE	0/63	0/20	8

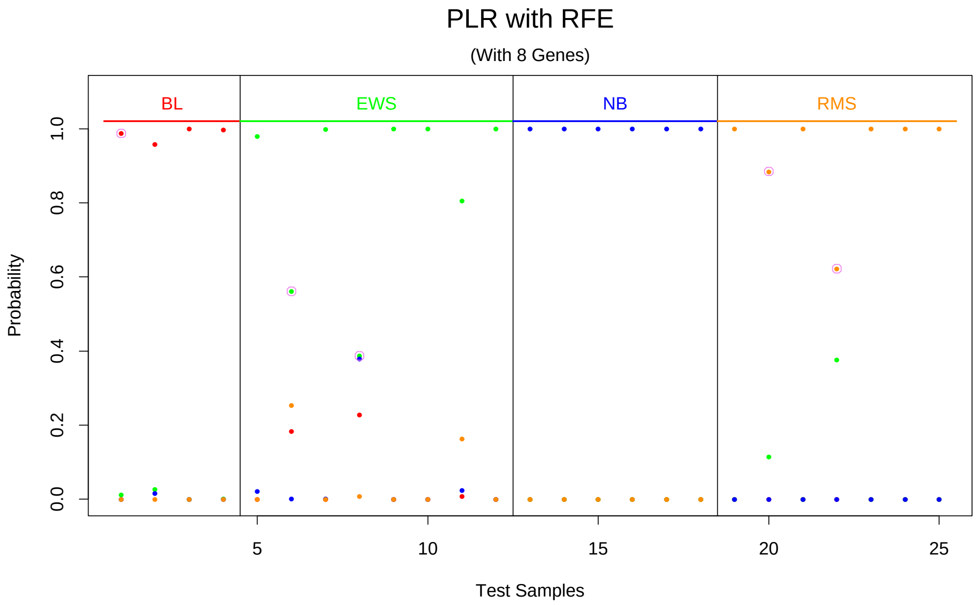
Result of UR


Use UR
Error
0.0
0.2
0.4
0.6
0.8
CV PLR
CV SVM
TE PLR
TE SVM
1
5
10
50
100
500
1000
Number of Genes


PLR with UR
(With 15 Genes)
Probability
0.0
0.2
0.4
0.6
0.8
1.0
BL
EWS
NB
RMS
5
10
15
20
25
Test Samples

Result of RFE





Golub data

Leukemia 2-class (Golub *et al.* 1999)

Method	CV	Test	Genes
SVM UR	2/38	3/34	22
PLR UR	2/38	3/34	16
SVM RFE	2/38	1/34	31
PLR RFE	2/38	1/34	26

Method	CV	Test	Genes
Golub *et al.* (1999)	3/38	4/34	50
Tibs *et al.* (2002)	2/38	2/34	21
PLR RFE	2/38	1/34	26

Ramaswamy data

Cancer 14-class (Ramaswamy *et al.* 2001)

Method	CV	Test	Genes
SVM UR	19/144	12/54	617
PLR UR	20/144	12/54	617
SVM RFE	17/144	9/54	315
PLR RFE	17/144	9/54	294

Method	CV	Test	Genes
Ramas *et al.* (2001)	32/144	12/54	16063
PLR RFE	17/144	9/54	294

Summary

- PLR and SVM are similar in classification.
- PLR provides probability estimates.
- PLR has a natural generalization in multi-class case.
- PLR RFE seems to perform well.
- Fast algorithm for PLR exists (not covered in this talk).
- PLR does not compress data. SVM does. Maybe not an issue for microarray.