A case study of classification in the longitudinal data

Team Six final project

Introduction
Motivation

- Classification problems are not uncommon in longitudinal data analysis (LDA).

  You can see several examples in Google, by searching ‘longitudinal’ and ‘classification’.
- Longitudinal data are often very ‘messy’.

  Different subjects can have different number of observations at different time points.
- Standard classification method requires each observation has the same input space.

Introduction
Purpose of the study

- Find an efficient way to extract the features of the longitudinal data, based on functional data analysis (FDA) point of view.
- Classify the longitudinal data in the feature space by using the Support Vector Machine and tree-based method.

Introduction
FDA vs. LDA

- LDA was only considered as an application of multivariate statistical analysis, in which the ‘atom’ is a vector.
- In the FDA, the ‘atom’ can be a curve, an image or a shape, which can be described as a function.

Introduction
Two approaches in FDA

- Direct approach: each individual curve is represented by a function from a set of basis functions.
- Mixed effects approach: each individual curve is represented by the group effect plus the random effect. Both of them are a linear combination of a set of basis functions.

Data description
yeast cell cycle data

- The data was originally utilized to identify all the genes whose mRNA levels are regulated by the cell cycle.
- Spellman *et al.* (1998) identified 800 genes from 6178 genes as cell cycle regulated genes.
- Three experiments in the data (alpha factor-, cdc- and elu- based experiments)

Data description
yeast cell cycle data (cont'd.)

- We only focused on the alpha factor-based synchronization experiment.
- It started with a collection of yeast cells, whose cycles are synchronized by a chemical process.
- A time-series of cDNA micro-arrays was gathered over 18 equally spaced time points, over about two hours, i.e. two cell cycles.

Data description
yeast cell cycle data (cont'd.)

- Among 6178 genes, 104 of them were known as cell cycle regulated genes by the traditional method (Spellman (1998) missed 9 of them).
- These 104 genes are in five different classes: M/G1, G1, S, S/G2, G2/M.
- This is essentially a semi-supervised problem, i.e. we have many unlabeled data (6074 genes). Hard to deal with. So we only focused on these 103 genes (one has no measurement in the alpha factor experiment).

YEAST CELL CYCLE PHASES

Cdc28: Cyclin-dependent kinase

Cdc36,37,39: START factors

Cdc6,7:Regul. DNA replication

Cdc4:SCF complex

Cdc46,47,48: MCM/P1family

Cdc24,42:GTPase (cell polarity)

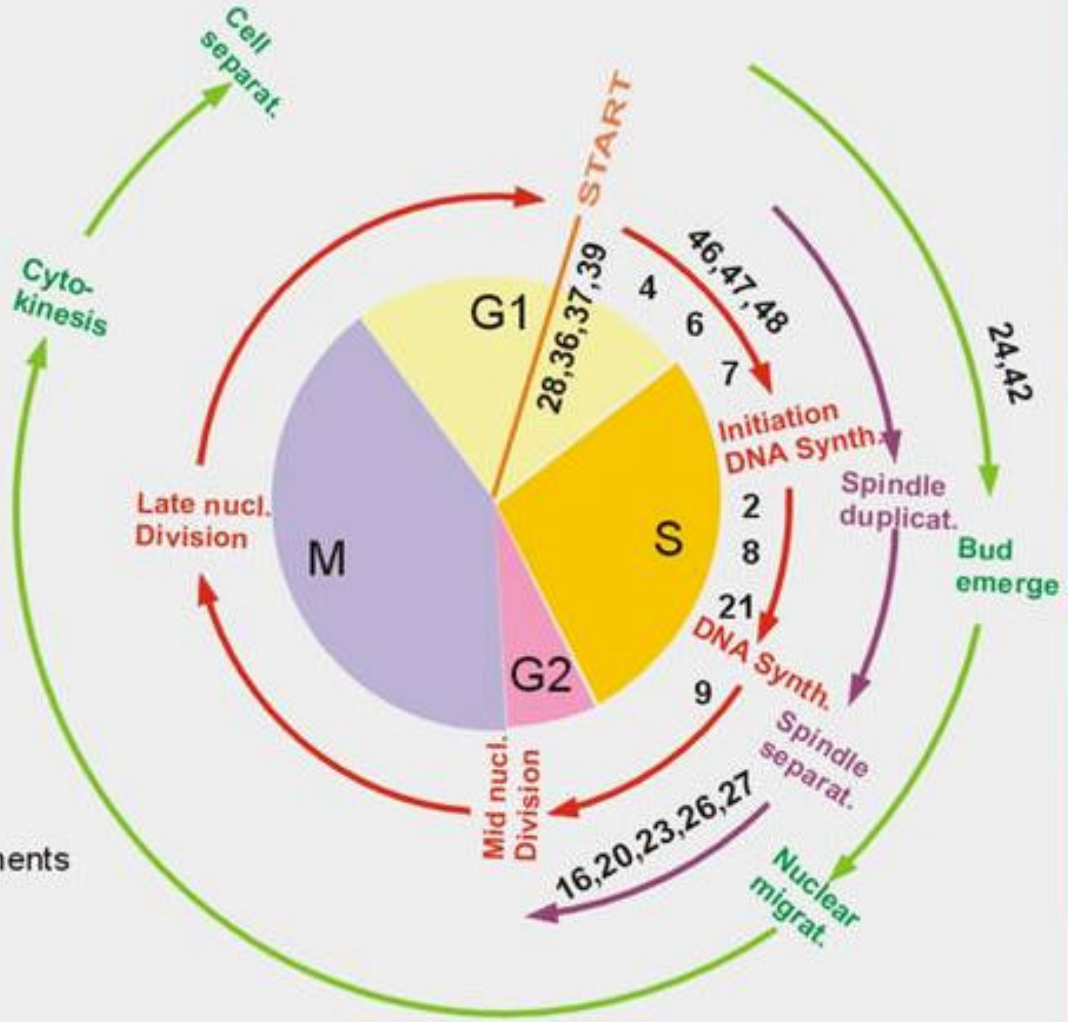
Cdc2:DNA Polymerase

Cdc8: Thymidilate synthase

Cdc21: dTMP synthase

Cdc9: DNA ligase

Cdc16,20,23,26,27: APC components



Data description: yeast cell cycle data (cont'd.)

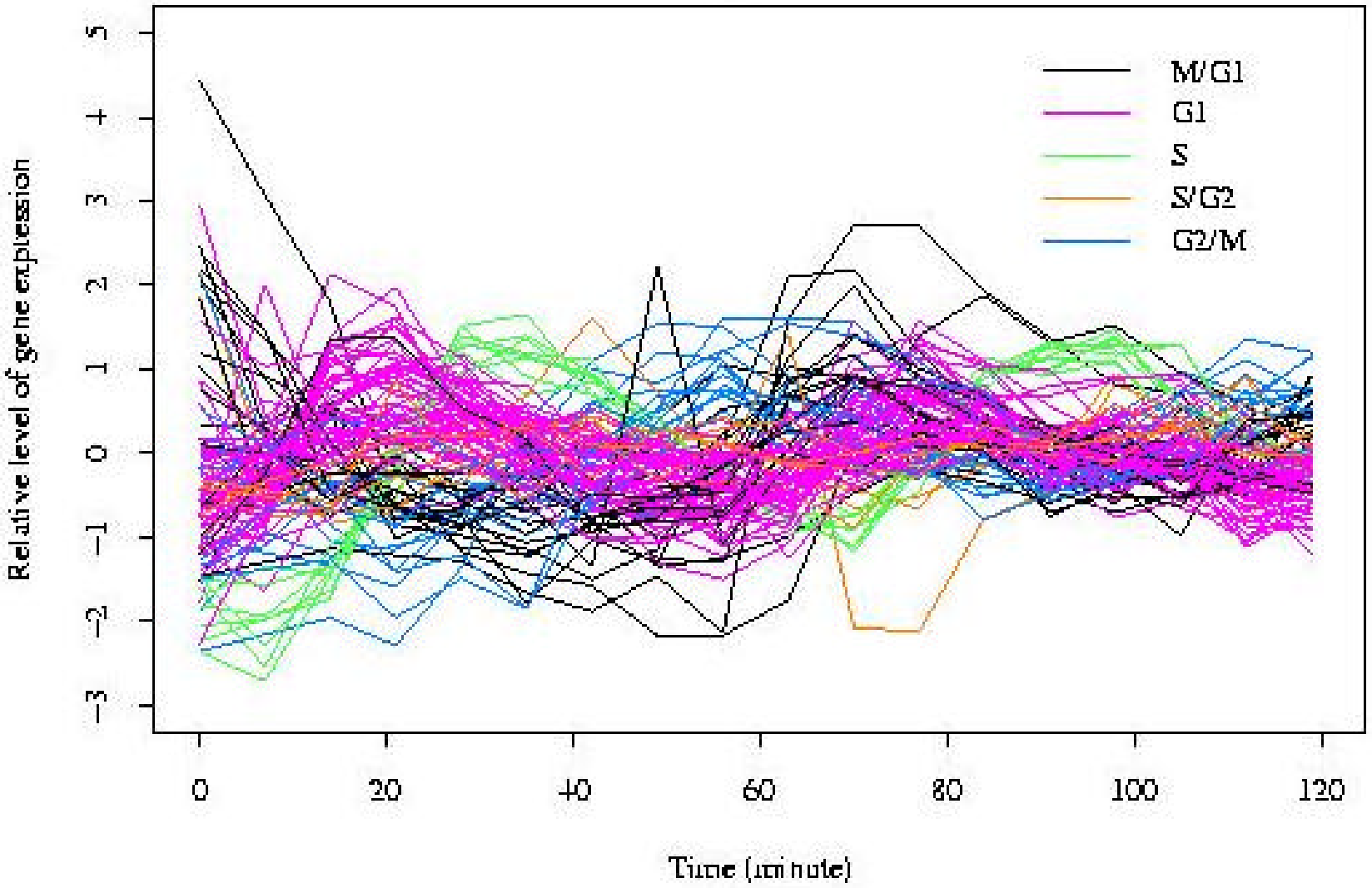
- Sample size: 103
- Number of classes: 5
- Number of genes in each class:

  M/G1: 19; G1: 52;

  S: 8; S/G2: 9;

  G2/M: 15;

Figure 1


Relative level of gene expression
−3
−2
−1
0
1
2
3
4
5
0
20
40
60
80
100
120
Time (minute)
M/G1
G1
S
S/G2
G2/M

Methodology
Model description

- Create a linear functional space, *G*, spanned by the basis functions:

$$\left\{1, \sin\left(\frac{2pt}{60/i}\right), \cos\left(\frac{2pt}{60/i}\right)\right\} \quad (i = 1, 2, \ldots, P)$$

- Fit each individual curve by a function from functional space *G*.
- P is set to 2 in our study.

Methodology
Model description (cont'd)

- When *P* is 2, the basis set is:

$$\left\{1, \sin\left(\frac{2pt}{60}\right), \cos\left(\frac{2pt}{60}\right), \sin\left(\frac{2pt}{30}\right), \cos\left(\frac{2pt}{30}\right)\right\}$$

- By the trigonometry formula, we know that the linear functional space *G* is essentially the linear functional space *F* with the basis set:

$$\left\{1, \sin\left(\frac{2pt}{60} + f_1\right), \cos\left(\frac{2pt}{60} + f_2\right), \sin\left(\frac{2pt}{30} + f_3\right), \cos\left(\frac{2pt}{30} + f_4\right)\right\}$$

Methodology
Model description (cont'd)

- The 'feature space' is the 4 dimensional vector space and each dimension is the coefficients of *sin* and *cos* function in

$$\left\{1, \sin\left(\frac{2pt}{60}\right), \cos\left(\frac{2pt}{60}\right), \sin\left(\frac{2pt}{30}\right), \cos\left(\frac{2pt}{30}\right)\right\}$$

Methodology
Some technical points

- We applied the Support Vector Machine (SVM) and Tree-based method on the created ‘feature space’.
- In the SVM study, Gaussian kernel was used.
- In order to simplify the problem (this is essentially a multicategory classification), we decompose it into five binary classification problems in SVM study.
- The ‘leave-one-out’ criterion was used in the cross-validation procedure.

Methodology
Some technical points (cont'd)

Classes		M/G1	G1	S	S/G2	G2/M
Training set	Positive set	12	35	5	6	10
	Negative set	53	34	59	63	59
Test set	Positive set	7	17	3	3	5
	Negative set	31	17	36	31	29

Results in SVM study

Classes	M/G1	G1	S	S/G2	G2/M
Training error	6/65	9/69	5/64	6/69	5/69
Test error	7/38	23/34	3/39	3/34	1/34

Results in SVM study (cont'd)

- Three classes have test error rates less than 10% (S, S/G2, G2/M).
- M/G1 has test error rate less than 20%.
- Test error rates of all the four classes above are close to their training error rates.
- G1 has training error rate about 13%, and test error around 68% (more than 5 times as much as training error rate)!

Results in SVM study (cont'd)

- Why the classification results is so bad in G1 class?
- We notice that G1 class consists two subclasses (SCB regulated and MCB regulated).
- So we did the classification of these two subclasses.

Results in SVM study (cont'd)

Classes	SCB regulated	MCB regulated
Training error	8/74	10/68
Test error	4/29	6/35

Results in SVM study (cont'd)

- Low test error = High quality classification**?**

  No!

- For example, in our study, we have 8 genes in class 'S' (5/64 in training set, 3/39 in test set). We can attain less than 10% training and test error rates by simply assigning all the samples as negative cases.

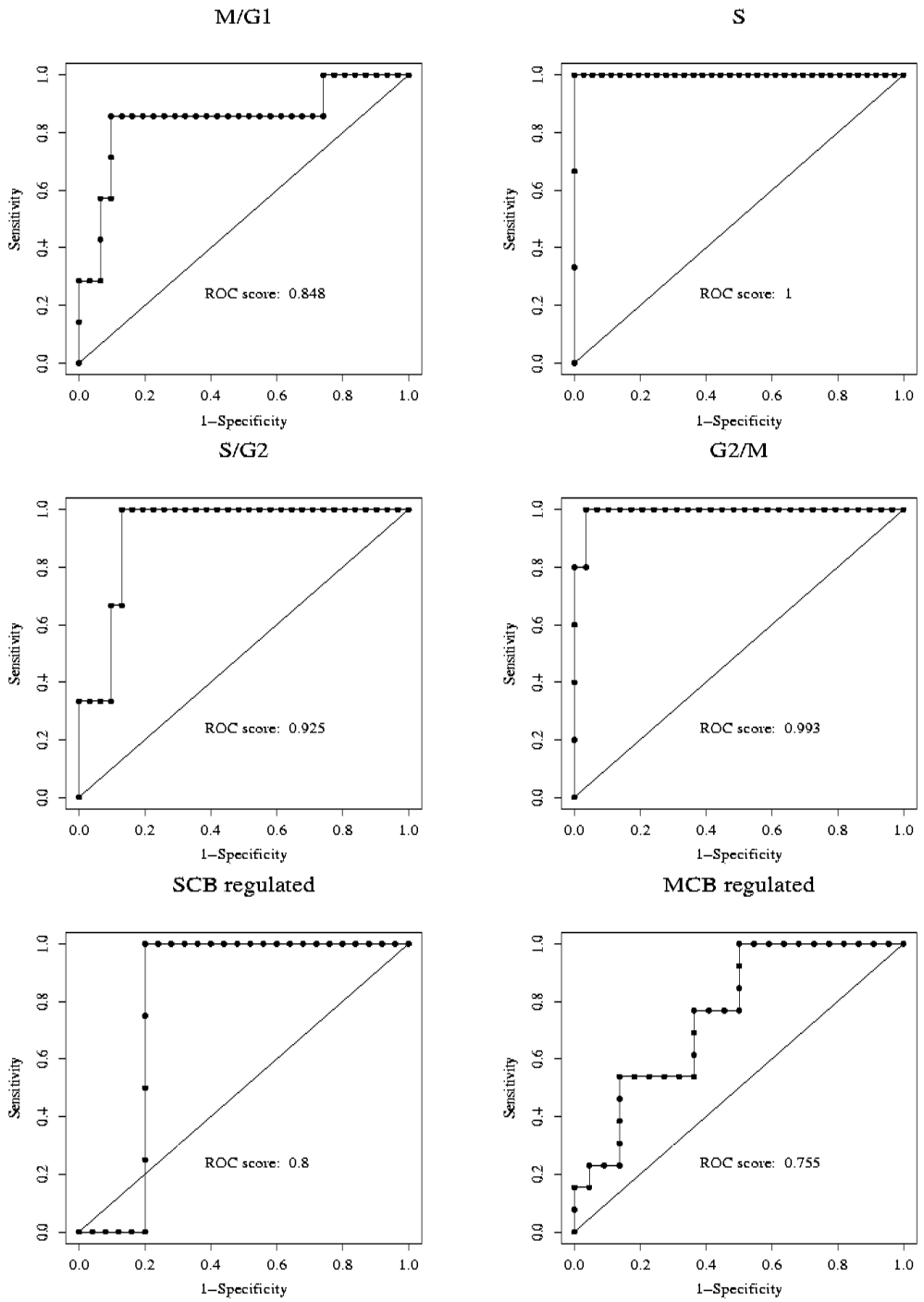
Results in SVM study (cont'd)

- We borrow the receiver operating characteristic (ROC) criterion, which is used extensively in biomedical field, to measure the classification quality.
- ROC score is the area under a curve that plots sensitivity as a function of 1- specificity for varying classification thresholds.
- A random test has ROC score close to 0.5, while a perfect test has ROC score close to 1.

Results in SVM study (cont'd)

- Sensitivity = TP/ (TP+FN)
- Specificity = TN / (TN + FP)

	Disease present	Disease absent
Test positive	True positive (TP)	False positive (FP)
Test negative	False negative (FN)	True negative (TN)


M/G1
Sensitivity
1−Specificity
ROC score: 0.848
S
Sensitivity
1−Specificity
ROC score: 1
S/G2
Sensitivity
1−Specificity
ROC score: 0.925
G2/M
Sensitivity
1−Specificity
ROC score: 0.993
SCB regulated
Sensitivity
1−Specificity
ROC score: 0.8
MCB regulated
Sensitivity
1−Specificity
ROC score: 0.755

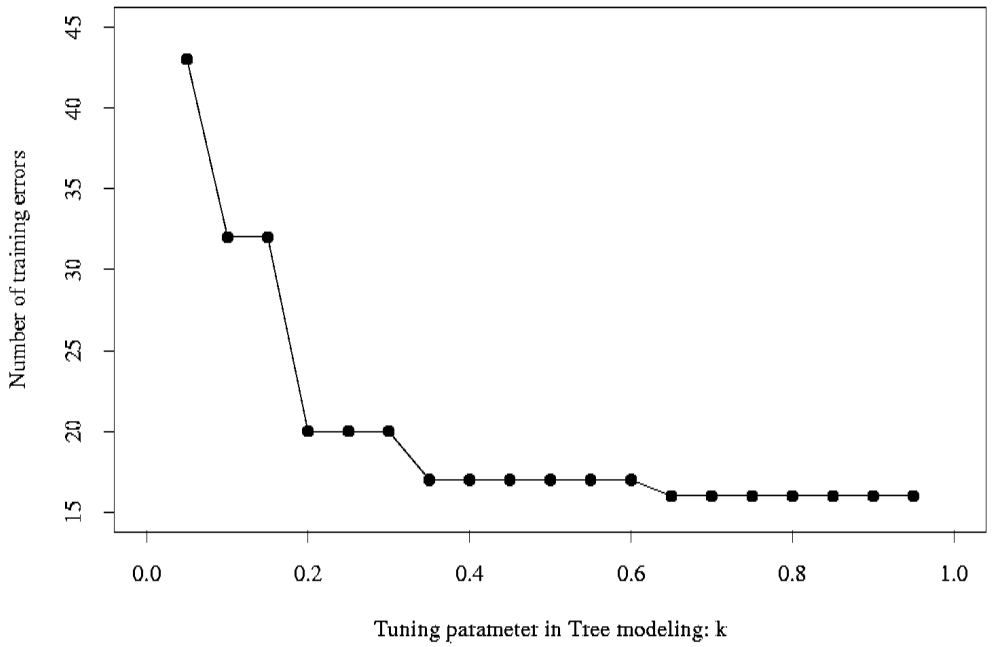
Results in SVM study (cont'd)

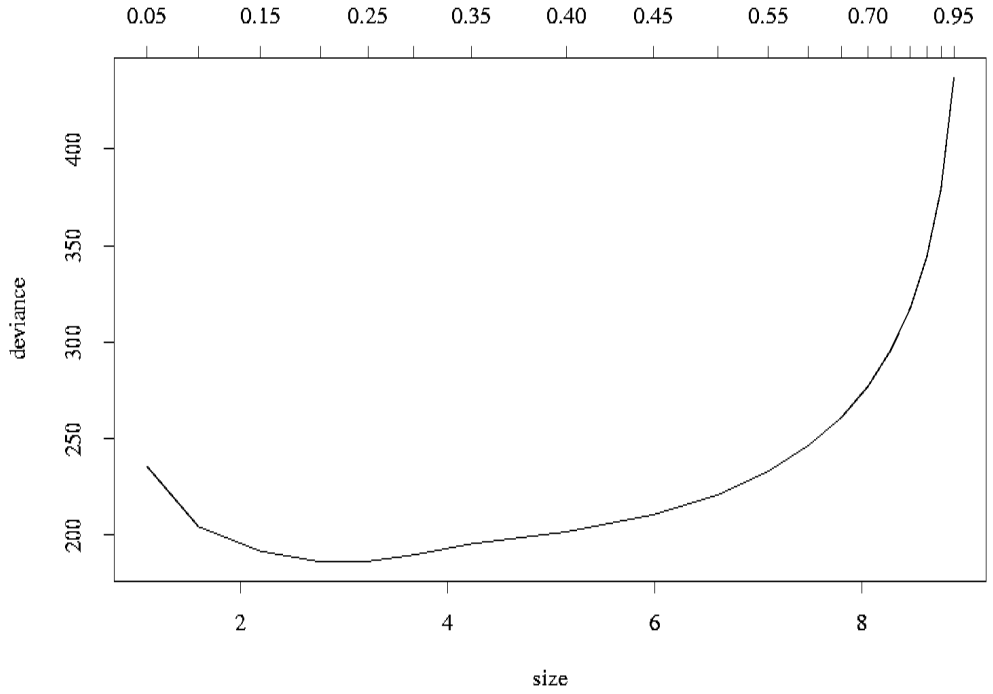
Classes	M/G1	S	S/G2	G2/M	SCB	MCB
ROC scores	0.848	1	0.925	0.993	0.8	0.755

Results in Tree-base method study

- we fitted the classification tree by using ‘shrink.tree’ function in Splus.
- *K* is the tuning parameter in the ‘shrink.tree’ function. So we use the cross-validation (leave-one-out) to determine the value *k*.
- *K* ranges from 0 to 1, the larger the *k*, the more complex the model is.

Figure 4

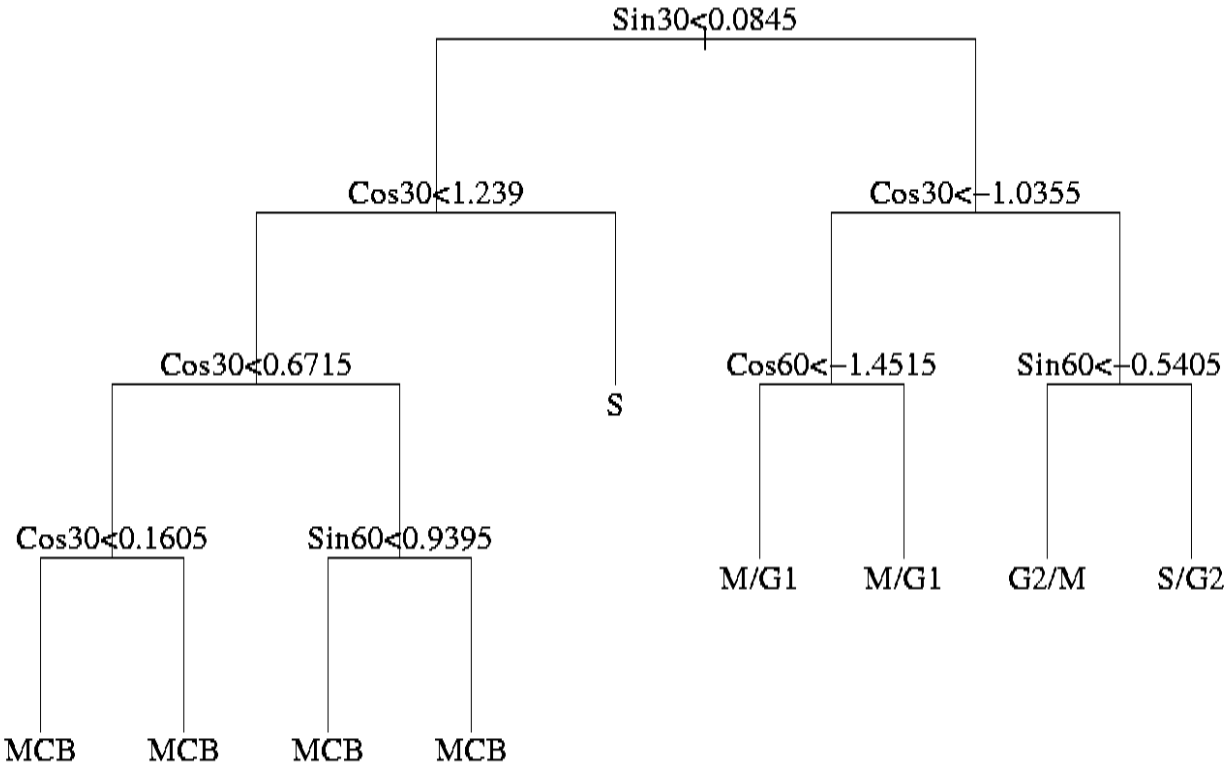

Number of training errors
45
40
35
30
25
20
15
0.0
0.2
0.4
0.6
0.8
1.0
Tuning parametet in Tree modeling: k


0.05
0.15
0.25
0.35
0.40
0.45
0.55
0.70
0.95
deviance
400
350
300
250
200
2
4
6
8
size

Results in Tree-base method study (cont'd)

- Set *k* to 0.35. The test error is 19 out of 34.

Figure 5



Discussion

- Pros and cons.
- Generalization

  The ‘feature space’ can be any vector space or even functional space.
- The key is: find a functional space that can extract the features efficiently.