

帕金森氏症言語重建

Rehabilitative speech treatment in
Parkinson' s disease

國立台北大學 統計學系 陳庭安

目錄

3 Data: 帕金森氏症言語重建 Rehabilitative speech treatment in Parkinson' s disease.....	3
3.1 緒論.....	3
研究背景.....	3
研究動機.....	3
研究目的.....	3
3.2 資料概述.....	4
資料來源.....	4
資料說明.....	4
變數說明.....	4
3.3 資料分析.....	9
3.3.1 探索性資料分析.....	9
3.3.2 分類—LDA 與 SVM.....	17
3.3.2.1 判別分析.....	17
3.3.2.2 SVM.....	24
3.3.3 治療後有(無)改善的聲音特徵.....	32
3.3.3.1 判別分析下的 ISOMAP 投影值.....	38
3.3.3.2 SVM 下的 ISOMAP 投影值.....	41
3.3.4 PD 患者發聲治療後未改善的症狀.....	43
3.3.5 分類—SVM.....	46
3.3.5.1 重複試驗 100 次的 10-Fold SVM.....	46
3.3.5.2 Leave-one-out 10-Fold SVM.....	53
3.3.5.3 治療後有(無)改善的聲音特徵.....	59

3 Data: 帕金森氏症言語重建 Rehabilitative speech treatment in Parkinson' s disease

3.1 緒論

- 研究背景

帕金森氏症(Parkinson' s disease, 簡稱 PD)是一種慢性中樞神經系統退化疾病，症狀通常隨時間緩慢出現，主要影響運動神經系統，早期的症狀為顫抖、肢體僵硬、運動功能減退和步態異常，也可能有認知和行為問題，平均每 10 萬人口就約有 100 人罹患 PD。

研究指出多數 PD 患者都有言語障礙症狀，其中約 30% 的患者認為言語障礙是相關症狀中最令他們感到不便的，通常也是 PD 出現的第一個症狀。典型的言語障礙症狀包括音量偏弱、單一音調、嘶啞、高音量的呼吸聲、不精確的發音和聲音震顫。

目前較常被施行，且被認為最有效果的療程叫做 Lee Silverman Voice Treatment (LSVT)，它又常被稱作 LSVT LOUD。這是因為這個治療法著重於矯正音量。藉由練習，訓練音量加大、音調變化、甚至誇張的發音，使患者習慣身體發不同聲音時，肌肉應該出多少力來把一個音發得清楚、音量適中。

- 研究動機

在做了 LSVT 療程一段時間後，醫師會追蹤患者發聲能力的治療狀況。專家會根據患者發聲的頻率、振幅、穩定性等，判斷患者治療後的發聲狀況是否可被接受，即是否與一般人發聲相似。若療程後發聲可被接受，就可以持續同樣的療程加以改善；若沒有通過判斷，療程就會稍作調整。

每位患者每隔一段時間就會被追蹤治療改善情形，然而人力的判斷耗時耗力，故此研究目的在於—將聲音特性經由演算法量化後，藉由訓練出的判斷器，自動預測發聲是否可被接受，療程是否需要稍作調整。

- 研究目的

治療結果預測錯誤將可能對患者的治療有很嚴重的影響。當帕金森氏症患者接受 LSVT 療程治療後發聲障礙的狀況有所改善，卻被預測為沒有改善，此時治療方式的調整可能會使得患者症狀改善速度停滯，甚至惡化。當患者治療後沒有改善，但被誤判為有改善，持續接受同樣的療程，可能對

患者成效仍然不大，症狀還是沒有獲得改善。因而，此治療結果自動判斷器的分類錯誤率越小越好。

有一定的預測準確度後，進一步欲探討治療後仍沒有改善的聲音主要特徵，即有發聲障礙的帕金森氏症患者的聲音的特徵有哪些。接著，若已預測某個聲音為未改善的聲音，此自動判斷器能更進一步經由演算法得到的數值，針對該名患者的發聲，判斷該聲音是可能是因為何種特徵所以被判斷為不合格，提供給 LSVT 療程專家作為未來治療調整的方向，

3.2 資料概述

- 資料來源

LSVT Voice Rehabilitation Data Set, UCI Machine Learning Repository

- 資料說明

此資料蒐集自 14 位具有言語障礙症狀且均透過 LSVT Loud 療程治療的帕金森氏症患者其聲音資訊。錄音地點統一在美國愛荷華大學國家語音中心 National Center for Voice and Speech-Denver (NCVS) 的隔音空間，使用相同錄音設備與系統(Audacity software package)。每位受測患者均錄下 9 種聲音，為高、中、低音調以及高、中、低振幅(音量)的組合情形，9 種聲音各錄 3 次，再由專家判斷 3 次錄音中錄得最成功(音調、振幅符合該次需要且無咳嗽等雜音)的一次記入資料檔，故共有 126(14*9)筆資料。透過 309 種臨床語音信號處理演算法從言語信號中提取訊息，檢驗發音的特徵並量化為數值資料。

在進行分析以前，資料由 2 位內部專家評估每筆聲音為可接受(正常範圍內、治療有效果)或不可接受(效果不彰)，再由 5 位外部專家(不知受測患者個人資料)獨立評估。最後整合 7 位專家的評估結果記錄在資料檔內。

- 變數說明

A. 工作表一

310 個變數，包含資料長度以及 309 種臨床語音信號處理演算法，依照衡量聲音的不同特性演算法分成四類，類型及其中含括的演算法與說明如下：

- 聲帶振動週期的偏離性**——一般人發聲聲帶振動具週期性，PD 患者則會部分偏離，甚至完全不具週期性。

Jitter 量化週期間基礎頻率的擾動（瞬時差異）

1. Jitter->F0_abs_dif

每期與下期頻率的平均絕對差，衡量整體頻率的擾動

2. Jitter->F0_dif_percent

平均絕對差/平均頻率，衡量整體頻率擾動的百分比

3-5. Jitter->F0_ PQ5_classical_

Schoentgen, classical_Baken,
generalised_Schoentgen

以自我迴歸模型的自我迴歸係數作為離均差的權數，再做平均絕對差/平均頻率(每五期計算)，衡量瞬時頻率擾動的百分比

6. Jitter->F0_abs0th_perturb

7. Jitter->F0_CV

8-13. Jitter -> F0_TKE0_

mean, std, prc5, prc25, prc75, prc95 TKE0

為振幅、頻率的調整，與信號的瞬時振幅和瞬時頻率成正比，衡量頻率調整後的平均、標準差與不受離群值影響的第 5、25、75、95 百分位數

14. Jitter->F0_FM

頻率調整 $(\text{Max}(\text{F0}) - \text{min}) / (\text{Max} + \text{min})$

15. Jitter->F0range_5_95_perc

頻率第 5、95 百分位的差

16-30. Jitter->pitch_

abs, percent, PQ5_classical_Schoentgen,
PQ5_classical_Baken, pitch_range_5_95_perc
(同 1-15. Jitter->F0_)

Shimmer 類比 Jitter，衡量的是振幅差異

31. Shimmer->Ampl_abs_dif

每期與下期振幅的平均絕對差，衡量整體振幅的

擾動

32. Shimmer->Ampl_dif_percent

平均絕對差/平均振幅，衡量整體振幅擾動的百分比

33-35. Shimmer->Ampl_PQ3_

classical_Schoentgen、

Shimmer->Ampl_PQ3_classical_Baken、

Shimmer->Ampl_PQ3_generalised_Schoentgen

以自我迴歸係數作為離均差的權數，再做平均絕對差/平均振幅(每三期計算)，衡量瞬時振幅擾動的百分比

36-38. Shimmer->Ampl_PQ5_

classical_Schoentgen、

Shimmer->Ampl_PQ5_classical_Baken、

Shimmer->Ampl_PQ5_generalised_Schoentgen

同上，但以每五期計算

39-41. Shimmer->Ampl_PQ11_

classical_Schoentgen、

Shimmer->Ampl_PQ11_classical_Baken、

Shimmer->Ampl_PQ11_generalised_Schoentgen

同上，但以每十一期計算

42. Shimmer->Ampl_abs0th_perturb

43. Shimmer->Ampl_dB

44. Shimmer->Ampl_CV

45-50. Shimmer->Ampl_TKE0

mean, std, prc5, prc25, prc75, prc95 TKE0
為振幅、頻率的調整，與信號的瞬時振幅和瞬時頻率成正比，在此衡量振幅調整後的平均、標準差與不受離群值影響的第 5、25、75、95 百分位數

51. Shimmer->Ampl_AM

振幅調整($\text{Max(AMPL)} - \text{min}$) / ($\text{Max} + \text{min}$)

0Q(Glottal Quotient) 發聲持續性(考慮聲門開闔)

56. 0Q→prc5_95

聲門打開時第 5、95 百分位的持續時間差與全距的比值

57. 0Q→std_cycle_open

聲門打開的時間標準差

58. 0Q→std_cycle_closed

聲門關閉的時間標準差

59. RPDE (Recurrence Period Density Entropy)

聲帶發聲週期持續性(發聲的穩定性)，
Jitter_pitch 的變形

61. PPE (Pitch Period Entropy)

控制頻率的障礙，Jitter_F0 的變形

- b. 信號噪音比(信噪比)——一般人發出聲音時聲帶會關閉，部分 PD 患者發出聲音時聲帶無法及時關閉，或是閉合不完全，此時空氣仍會在聲帶流動，導致不必要的噪音。

HNR (Harmonics to Noise Ratio) 噪音比例

52-53. HNR→HNR_dB_Praat_

mean、std

利用分貝序列的自我相關係數計算而成 NHR，
與 HNR 反向

54-55. NHR→NHR_Praat_

mean、std

同 HNR

60. DFA (Detrended Fluctuation Analysis)

能辨識由聲道中的氣流引起的噪音、處理非定態的序列

GNE (Glottal to Noise Excitation) 量化噪音在信號中的程度，但無法辨識出氣流引起的共鳴

62-63. GNE→mean、std

64、66. GNE→SNR_TKEO、NSR_TKEO

調整後的信噪程度

65、67. GNE→SNR_SEO、NSR_SEO

VFER (Vocal Fold Excitation Ratio) 目標是量化非線性，相互作用的生理現象，病理性聲帶振動模式（不完整的聲帶閉合導致產生漩渦和湍流噪聲）和咬合的位置（影響能量分佈）

68-76. VFER→

mean、std、entropy、SNR_TKEO1、
SNR_TKEO、SNR_SEO、NSR_TKEO1、NSR_TKEO、
NSR_SEO

IMF (Intrinsic Mode Functions) 在振幅和頻率兩方面貢獻的觀測信號

77-82. IMF→

SNR_SEO、SNR_TKEO、SNR_entropy、
NSR_SEO、NSR_TKEO、NSR_entropy

c. 小波衡量—受測者被要求持續性的發聲，並將聲波分成數段，衡量不同波段的波型。

125-127. F0-series 基礎頻率分布、差異等

129. Ea

130-139. Ed coef 1-10 段

140-310. entropy_shannon, entropy_log,
det_TKEO_mean, det_TKEO_std 1-10, 2_1-
4_10

- d. 語音訊號能量—評估不同波段訊號的能量大小，由於 PD 患者無法有效控制發聲，故音量大小並不穩定。

83. Log energy

84-96. MFCC_0-12th coef

97. delta log energy

98-110. 0-12th delta

111. delta delta log energy

112-124. 0-12th delta delta

- e. 其他

128. Data_length

B. 工作表二

1 個變數，專家判斷結果(1=acceptable,
2=unacceptable)

C. 工作表三

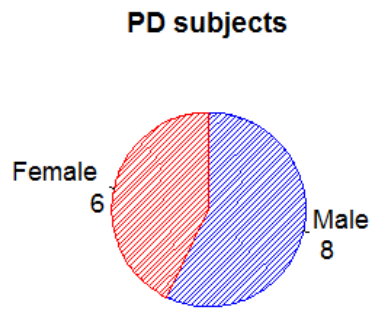
3 個變數，受測者編號(1~17, 不含 9、10、17，原受測者
為 17 位，其中 3 位放棄受測)、年齡、性別(0=Male,
1=Female)

3.3 資料分析

3.3.1 探索性資料分析

一、受測者性別比

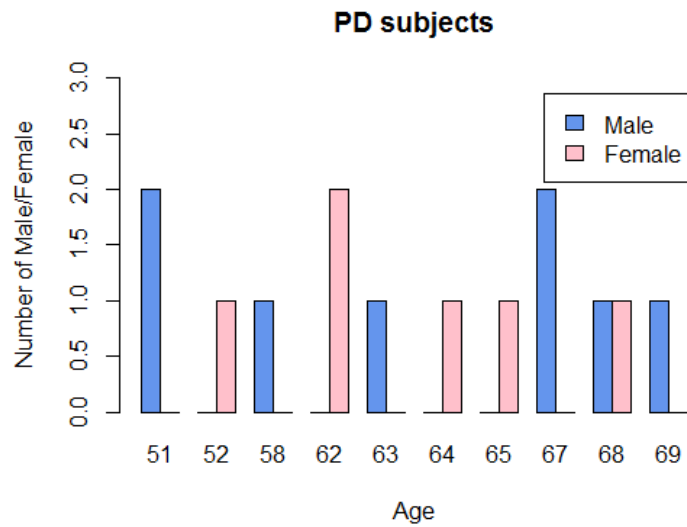
```
> library(readxl)
>
> data <-
read_excel("data\\LSVT_voice_rehabilitation.xlsx",
+           sheet = 1, col_names = T)
> d.class <-
read_excel("data\\LSVT_voice_rehabilitation.xlsx",
+           sheet = 2, col_names = T)
> d.subject <-
read_excel("data\\LSVT_voice_rehabilitation.xlsx",
+           sheet = 3, col_names = T)
> colnames(d.class) <- "class"
> colnames(d.subject) <- c("id", "age", "sex")
> class <- d.class$class
> dim(data)
[1] 126 310
> dim(d.class)
[1] 126 1
> dim(d.subject)
[1] 126 3
> # EDA
> ## Subject
> # gender
> pie.lab <- paste(c("Male", "Female"), "\n",
table(d.subject$sex) / 9)
> pie(table(d.subject$sex) / 9, labels = pie.lab,
+      col = c("blue", "red"), clockwise = T,
+      main = "PD subjects", cex = 1.1, density = 30)
>
```



受測者共 14 名，其中 6 名為女性，8 名為男性。

二、受測者性別與年齡分布

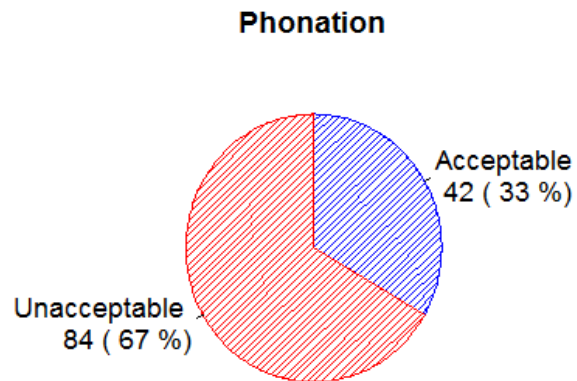
```
> # age, gender
> d.age.sex <- data.frame(table(d.subject[, -1]) / 9)
> barplot(t(cbind(d.age.sex[1:10, 3], d.age.sex[11:20,
3])),
+       beside = T, names.arg = d.age.sex[1:10, 1],
+       main = "PD subjects", ylim = c(0, 3),
+       xlab = "Age", ylab = "Number of Male/Female",
+       col = c("cornflowerblue", "pink"),
+       legend.text = c("Male", "Female"),
+       args.legend = "topright")
>
```



14 位受測者年齡介於 51 至 69 歲，分布如上。

三、受測者發聲治療結果

```
> ## class(whether the treatment effect is
good(acceptable))
> prop.phon.judge <- round(prop.table(table(d.class)) *
100)
>
> pie.lab2 <- paste(c("Acceptable", "Unacceptable"), "\n",
+                   table(d.class), "(", prop.phon.judge,
"%)")
> pie(table(d.class), labels = pie.lab2,
+      col = c("blue", "red"), clockwise = T,
+      main = "Phonation", cex = 1.1, density = 30)
>
```



總共 126 筆接受 LSVT 療程的 PD 患者的聲音資料，經 7 位專家判斷後，有 42 筆(33%)可被接受，即療程治療有效；有 84 筆(67%)治療沒有明顯幫助。

四、 不同性別與不同年齡的治療結果差異

```
> # Gender/Age v.s. class
> prop.table(table(d.subject$sex, class), 1)
      class
      1      2
0 0.3333333 0.6666667
1 0.3333333 0.6666667
> prop.table(table(d.subject$age, class), 1)
      class
      1      2
51 0.3333333 0.6666667
52 0.3333333 0.6666667
58 0.3333333 0.6666667
62 0.3333333 0.6666667
63 0.3333333 0.6666667
64 0.3333333 0.6666667
65 0.3333333 0.6666667
```

```

67 0.3333333 0.6666667
68 0.3333333 0.6666667
69 0.3333333 0.6666667
> table(class, d.subject$age, d.subject$sex)
, , = 0

class 51 52 58 62 63 64 65 67 68 69
  1  6  0  3  0  3  0  0  6  3  3
  2 12  0  6  0  6  0  0 12  6  6

, , = 1

class 51 52 58 62 63 64 65 67 68 69
  1  0  3  0  6  0  3  3  0  3  0
  2  0  6  0 12  0  6  6  0  6  0

> # No difference in accept./unaccept. ratio between
> # F/M., ages.
>

```

<div>治療結果</div> <div>性別</div>	有改善	無改善
男性	0.33	0.67
女性	0.33	0.67

<div>治療結果</div> <div>年齡</div>	有改善	無改善
51, 52, 58, 62~69	0.33	0.67

<div>治療結果</div> <div>性別(年齡)</div>	有改善	無改善
-----------------------------------	-----	-----

男性(51, 58, 63, 67~69)	0.33	0.67
女性(52, 62, 64, 65, 68)	0.33	0.67

五、演算法與治療結果的關聯性

```

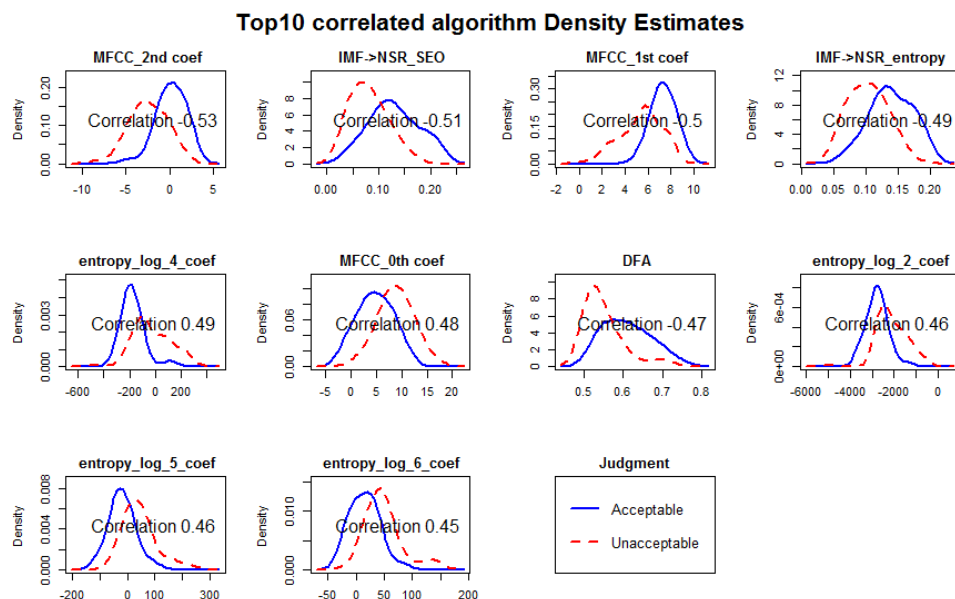
> ## Relationship between algorithm & class(treatment
effect)
> # correlations
> corr.algo.cl <- apply(data, 2, function(x) cor(x, class))
> top10.corr.var <- head(names(sort(abs(corr.algo.cl),
+                               decreasing = T)), 10)
> top10.high.corr <- corr.algo.cl[top10.corr.var]
> last10.corr.var <- head(names(sort(abs(corr.algo.cl))),
10)
> last10.corr <- corr.algo.cl[last10.corr.var]
>
> ## Density plot
> library(sm)
> # accept./ unaccept.
> class.f <- factor(class, levels = c(1, 2),
+                   labels = c("Acceptable", "Unacceptable"))
> denplot.fun <- function(algor, cor) {
+   var <- as.numeric(as.matrix(data[, algor]))
+   sm.density.compare(var, class.f, xlab = "",
+                       col = c(4, 2), lwd = 2)
+   title(paste("\n\n", algor))
+   legend("center", bty = "n", cex = 1.5,
+          paste("Correlation", round(cor[algor], 2)))
+ }
> par(mfrow = c(3, 4))
> par(mai = c(0.4, 0.6, 0.6, 0.2))
> # Top10
> top10.denplot <- apply(as.matrix(top10.corr.var), 1,
+                         denplot.fun, top10.high.corr)
> plot(0, xaxt = "n", yaxt = "n", type = "n",
+       xlab = "", ylab = "", main = "\n\nJudgment")
> legend("center", levels(class.f), bty = "n", cex = 1.2,
+       col = c(4, 2), lty = 1:2, lwd = 2)

```

```

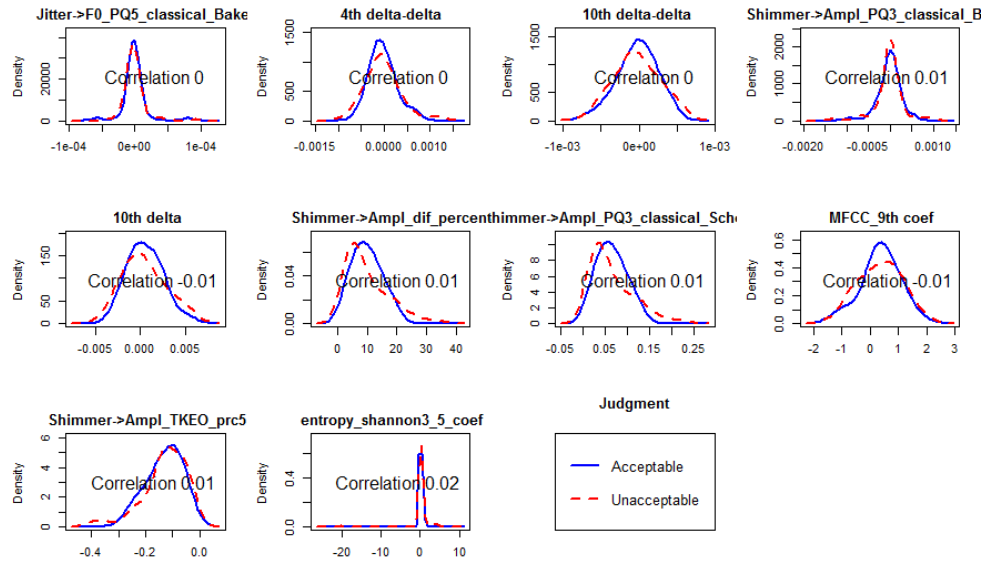
> title("\nTop10 correlated algorithm Density Estimates",
+       outer = T, cex.main = 2)
> # Last10
> par(mfrow = c(3, 4))
> par(mai = c(0.4, 0.6, 0.6, 0.2))
> last10.denplot <- apply(as.matrix(last10.corr.var), 1,
+                          denplot.fun, last10.corr)
> plot(0, xaxt = "n", yaxt = "n", type = "n",
+       xlab = "", ylab = "", main = "Judgment")
> legend("center", levels(class.f), bty = "n", cex = 1.2,
+       col = c(4, 2), lty = 1:2, lwd = 2)
> title("\n10 Least correlated algorithm Density
Estimates",
+       outer = T, cex.main = 2)
>

```



上圖為與發聲改善結果最相關的十個訊號演算法在兩種判斷結果的分布，圖中數字表示改善結果與演算法的相關係數。由圖可知有無改善的發聲的演算數值有所不同，有改善的發聲在部分演算法數值較高，在部分演算法則較低。

10 Least correlated algorithm Density Estimates



與發聲改善結果最不相關的十個訊號演算法，有無改善的聲音的演算觀測值相近。

3.3.2 分類—LDA 與 SVM

治療結果的錯誤預測將可能對患者的治療有很嚴重的影響。當帕金森氏症患者接受 LSVT 療程治療後發聲障礙的狀況有所改善，卻被預測為沒有改善，此時治療方式的調整可能會使得患者症狀改善速度停滯，甚至惡化。當患者治療後沒有改善，但被誤判為有改善，持續接受同樣的療程，可能對患者成效仍然不大，症狀還是沒有獲得改善。因而，分類錯誤率最小化是首要目標。

由前一節知並非所有的演算法均與帕金森氏症患者聲音治療結果有高相關，故期望經由變數的篩選以及維度縮減使得分類錯誤率下降。

3.3.2.1 判別分析

一、分類錯誤率最小化下的變數選取以及維度縮減方法

```

> ## Feature & Dimension Reduction method Selection
> # (criterion: Max. classification(LDA) accuracy)
>
> # 1. Data updated by
> #   Oredered significant variables (algorithm)
> bw.ratio <- function(x, y){
+   tg <- table(y)
+   gm <- tapply(x, y, mean)
+   bw <- (sum((gm-mean(x))^2))/(sum((x - rep(gm, tg))^2))
+ }
> bw.values <- apply(data, 2, bw.ratio, class)
> order.bw <- order(bw.values, decreasing = T)
> d.algor.ord <- as.matrix(data)[ , order.bw]
>
> # 2. LDA, leave-one-out, function
> library(MASS)
> lda accur.fun <- function(dim, data) {
+   lda.data <- lda(x = data[ , 1:dim],
+                   grouping = class.f, CV = T)
+   # In case of NaN
+   keep <- unique(unlist(apply(lda.data$posterior, 2,
function(x)
+     which(is.nan(x) != 1))))
+   confusion.mat <- table(class[keep],
+                           lda.data$class[keep])
+   accuracy <- sum(diag(prop.table(confusion.mat)))
+   accuracy
+ }
>
> # 2. Dimension Reduction method Selection (matplot.)
> algor.DR.method.select.fun <- function(algor.num) {
+   LSVT <- d.algor.ord[ , 1:algor.num]
+   # a). PC scores
+   library(FactoMineR)
+   library(ggplot2)
+   library(factoextra)
+   lsvt.pca <- PCA(LSVT, scale.unit = T, graph = F,
+                   ncp = dim(LSVT)[1] - 1)
+   eig <- get_eigenvalue(lsvt.pca)
+   var <- get_pca_var(lsvt.pca)

```

```

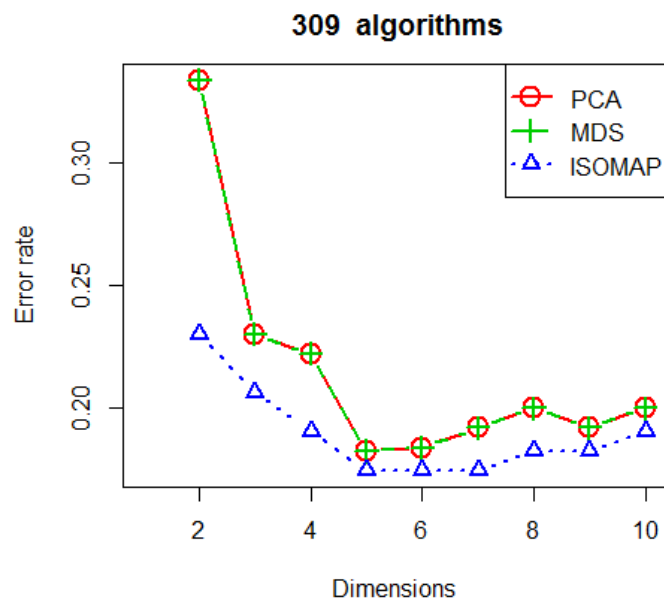
+ ind <- get_pca_ind(lsvt.pca)
+ lsvt.pc.score <- ind$coord[ , 1:10]
+ lda.pc.score accur <- apply(as.matrix(2:10), 1,
+                             lda accur.fun, lsvt.pc.score)
+ # b). MDS projected value
+ mds.subj <- cmdscale(dist(scale(LSVT)), k = 10)
+ lda.mda. accur <- apply(as.matrix(2:10), 1,
+                         lda accur.fun, mds.subj)
+ # c). ISOMAP projected value
+ library(vegan)
+ iso.score.k <- apply(as.matrix(2:20), 1,
function(neighbor) {
+   lsvt.isomap <- isomap(dist(scale(LSVT)),
+                         ndim = 10, k = neighbor)$points
+   dim. accur <- apply(as.matrix(2:10), 1,
+                      lda accur.fun, lsvt.isomap)
+   dim. accur
+ })
+ lda.iso. accur <- apply(iso.score.k, 1, max)
+
+ lda. accur <- rbind(rep(NA, 3),
+                    data.frame(PCA = lda.pc.score. accur,
+                               MDS = lda.mda. accur,
+                               ISOMAP = lda.iso. accur))
+ matplot(1 - lda. accur, pch = c(1, 3, 2), type = "b",
+         col = 2:4, lwd = 2, cex = c(2, 1.5, 1.2),
+         main = paste("\n", algor.num, " algorithms"),
+         xlab = "Dimensions", ylab = "Error rate")
+ }
> # Before feature selection
> algor.DR.method.select.fun(dim(d.algor.ord)[2])
> legend("topright", c("PCA", "MDS", "ISOMAP"),
+       pch = c(1, 3, 2), lwd = 2, col = 2:4,
+       lty = 1:3, pt.cex = c(2, 1.5, 1.2))
> # After feature selection (ISOMAP is always the best)
> par(mfrow = c(3, 3))
> par(mai = c(0.6, 0.6, 0.5, 0.2))
> algor.DR.method.select <- apply(as.matrix(c(20, 40, 60,
+       80, 100, 150,
+       200, 250, 309))),

```

```

+                                     1, algor.DR.method.select.fun)
> legend("top", c("PCA", "MDS", "ISOMAP"),
+       pch = c(1, 3, 2), lwd = 2, col = 2:4, horiz = T,
+       lty = 1:3, pt.cex = c(1.5, 1.2, 1.2), cex = 0.8,
+       bg = cm.colors(1, alpha = 0), bty = "n")
> title("\nError Rate by LDA upon Variables Selection",
+       outer = T)
>

```

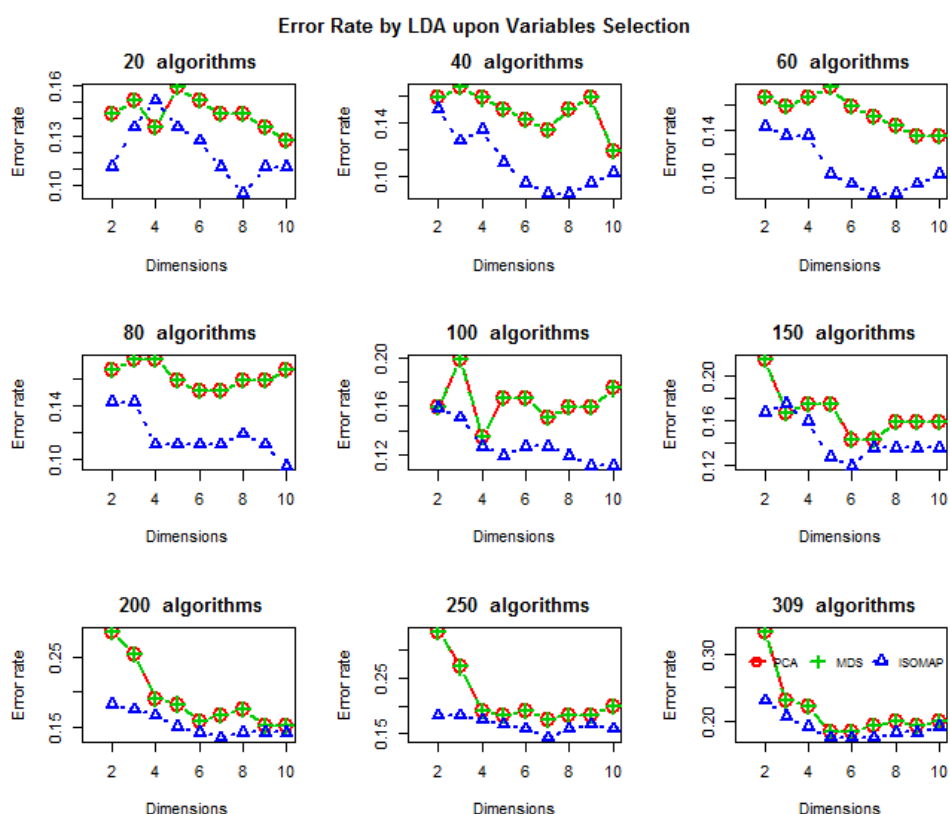


將總共 309 個訊號演算法以主成分分析、多元尺度法分析以及 ISOMAP 維度縮減後，分別取前 2 至 10 個維度作判別分析，預測 129 個發聲經治療後是否有改善，進一步比對判別分析的預測與專家判斷的結果，探究預測的錯誤率大小。

無論所取的維度大小為何，ISOMAP 的分類錯誤率均比 PCA 與 MDS 小，其中又以取 ISOMAP 前 5 個維度的錯誤率最小。

然而，此 309 個信號演算法並非所有都對治療是否改善的結果具顯著的區辨力，因此需做變數的篩選，篩選出—在不同改善結果，演

算值差異較大；在相同的結果下，差異較小。故將所有變數依照群間、群內差異比值由大至小排序，選擇群間、群內變異比值較大的變數。以下為不同變數篩選個數下，經過維度縮減後，判別分析的分類錯誤率：



先分別篩選前 20、40、60、80、100、200、250 及 309 個最具區辨力的變數，即群間、群內變異比值最大的變數，再分別利用 PCA、MDS 與 ISOMAP 維度縮減後的維度去做分類，分類錯誤率如上圖。由圖可知，無論選擇變數的多寡、維度的多寡，ISOMAP 的分類錯誤率幾乎是最小的。因此 ISOMAP 為當中最適合的維度縮減方法。

以下接著將上圖中不同篩選個數—ISOMAP 所取各個維度的錯誤率繪製在一起，討論較適合的變數篩選個數。

```

> # 3. Feature selection
> # Compare isomap error rate in diff. number of var.
selected
> algor.isomap.select.fun <- function(algor.num) {
+   LSVT <- d.algor.ord[ , 1:algor.num]
+   # c). ISOMAP projected value
+   iso.score.k <- apply(as.matrix(2:20), 1,
function(neighbor) {
+     lsvt.isomap <- isomap(dist(scale(LSVT)),
+                           ndim = 10, k = neighbor)$points
+     dim accur <- apply(as.matrix(2:10), 1,
+                         lda accur.fun, lsvt.isomap)
+     dim accur
+   })
+   lda.iso accur <- apply(iso.score.k, 1, max)
+   lda.iso accur
+ }
> algor.isomap.select <- rbind(rep(NA, 9),
+                               apply(as.matrix(c(20, 40, 60,
+                                                  80, 100, 150,
+                                                  200, 250, 309)),
+                               1, algor.isomap.select.fun))
> library(fields)
> matplot(1-algor.isomap.select, pch = 16,
+         type = "l", lwd = 4:3, lty = 2:1,
+         xlim = c(0.5, 10.5), ylim = c(0.06, 0.28),
+         col = tim.colors(9), xlab = "Dimensions",
+         ylab = "Error rate", main = "ISOMAP Classification
Error Rate",
+         xaxt = "n")
> axis(1, at = 2:10, 2:10)
> legend("topright", legend = rev(c(20, 40, 60, 80, 100,
150,
+                                   200, 250, 309)),
+        lwd = 3, col = rev(tim.colors(9)), ncol = 3,
+        title = "Feature selection", cex = 0.8)
> points(rep(7, 2),
+         rep(1-algor.isomap.select[8, 2], 2),
+         pch = c(16, 1), col = 2, cex = c(1, 3)) #
(Continue...)

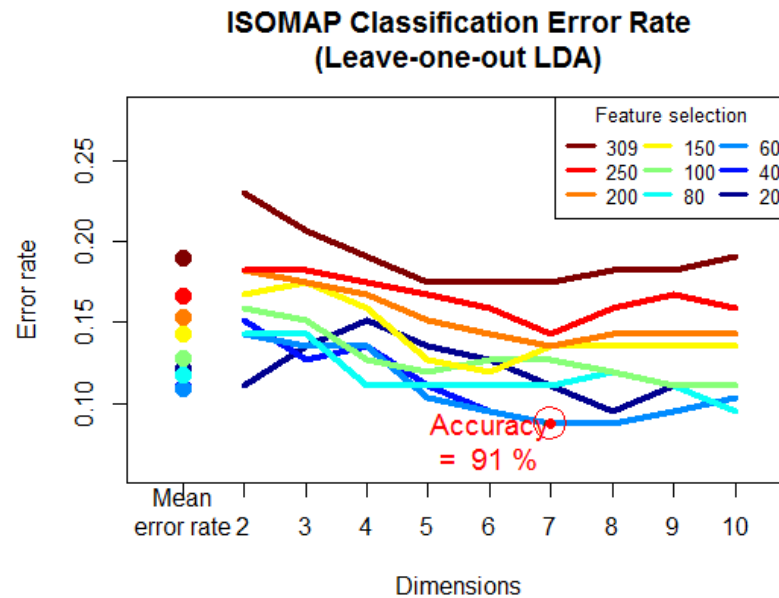
```

```

> # Thus, the first 40 sig. var.(features) are selected
> # to classify trt. consequences (LDA)
> # after Dimension Reduction (ISOMAP dim.7).
>
> # 4. Data that contains only the 40 most sig. var.
> data.sig.var <- d.algor.ord[ , 1:40]
>
> # 5. ISOMAP dim.7, LDA accuracy
> ## Isomap dim.1-7
> # a). Find the most proper k(# of neighbors)
> accur.given.k <- apply(as.matrix(2:20), 1, function(x) {
+   iso.pts <- isomap(dist(scale(data.sig.var)),
+                     ndim = 7, k = x)$points
+   accuracy <- lda accur.fun(dim = 7, data = iso.pts)
+   accuracy
+ })
> best.k <- (2:20)[which.max(accur.given.k)]
>
> # b). Isomap
> lsvt.isomap <- isomap(dist(scale(data.sig.var)),
+                       ndim = 7, k = best.k)
> # c). classification by LDA
> lda.isomap <- lda(x = lsvt.isomap$points,
+                  grouping = class.f, CV = T)
> iso.best accur <- sum(diag(prop.table(
+   table(class, lda.isomap$class)))) # 91.27%
>
> # (continue...accuracy written in)
> text(6, 1-algor.isomap.select[8, 2]-0.01,
+      paste("Accuracy\n= ", round(100*iso.best accur),
+            "%"),
+      col = 2, cex = 1.2)
> # Mean error rate in Different numbers of var. selected
> mean.err.rate <- 1-apply(algor.isomap.select[-1, ], 2,
mean)
> par(new = T)
> plot(rep(1, 9), mean.err.rate,
+      col = tim.colors(9), pch = 16, cex = 1.5,
+      xlab = "", ylab = "", xaxt = "n", yaxt = "n",
+      bty = "n", xlim = c(0.5, 10.5), ylim = c(0.06, 0.28))

```

```
> axis(1, at = 1, "Mean\nerror rate", tick = "n")
>
```



隨著最具區辨力的變數個數從 309、250、……篩選至 60、40 個，做維度縮減後分類，錯誤率會下降，但篩選至 20 個時錯誤率又上升，再次說明有部分的變數—信號演算法不具足夠的區辨力，以至於考量那些演算法的值時會使得整體分類錯誤率增加。

由圖所示，變數篩選前 40 個最具區辨力的演算法，再做 ISOMAP 降維，並取前 7 個維度，此時預測發聲是否改善的錯誤率最低，僅 9%，即有 91% 的預測準確率。

3.3.2.2 SVM

在利用判別分析方法找到最適(分類錯誤率最低)的變數篩選以及維度縮減方法後，此節另使用 SVM 做分類，與判別分析法相互比較。

一、分類錯誤率最小化下的變數選取以及維度縮減方法

```
> ## Feature & Dimension Reduction method Selection
> # (criterion: Max. classification(SVM) accuracy)
>
> # 1. Data updated by
> #   Ordered significant variables (algorithm)
>
> # d.algor.ord <- as.matrix(data)[ , order.bw]
>
> # 2. SVM, 10-fold CV, function
> library(caret)
> set.seed(12345)
> test.id.set <- createFolds(y = 1:dim(LSVT)[1], k = 10)
>
> library(e1071)
> svm accur.fun <- function(test.id, data, dim) {
+   x.train <- data[-test.id, 1:dim]
+   x.test <- data[test.id, 1:dim]
+   y.train <- class.f[-test.id]
+   y.test <- class.f[test.id]
+   train.set <- data.frame(x.train, y.train)
+
+   svm.model <- svm(y.train ~ ., data = train.set)
+
+   pred <- predict(svm.model, x.test)
+   confus.mat <- table(pred, y.test)
+   accuracy <- sum(diag(prop.table(confus.mat)))
+   accuracy
+ }
>
> # 2. Dimension Reduction method Selection (matplot.)
> library(dplyr)
> algor.DR.method.select.svm <- function(algor.num) {
+   LSVT <- d.algor.ord[ , 1:algor.num]
+   # a). PC scores
+   # library(FactoMineR)
+   # library(ggplot2)
+   # library(factoextra)
+   lsvt.pca <- PCA(LSVT, scale.unit = T, graph = F,
+                   ncp = dim(LSVT)[1] - 1)
```

```

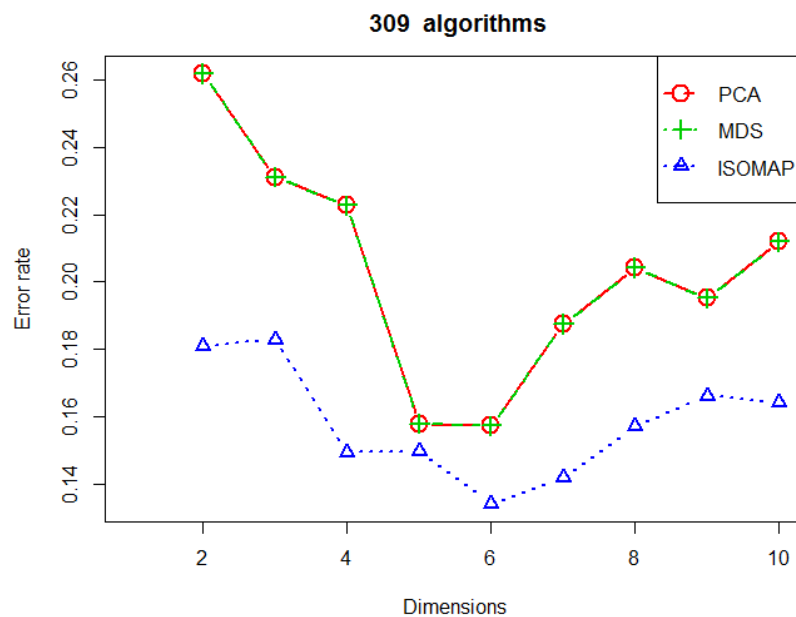
+ eig <- get_eigenvalue(lsvt.pca)
+ var <- get_pca_var(lsvt.pca)
+ ind <- get_pca_ind(lsvt.pca)
+ lsvt.pc.score <- ind$coord[ , 1:10]
+ svm.pc accur <- apply(as.matrix(2:10), 1, function(x)
+   lapply(test.id.set, svm accur.fun,
+     lsvt.pc.score, x) %>% unlist %>% mean)
+
+ # b). MDS projected value
+ mds.subj <- cmdscale(dist(scale(LSVT)), k = 10)
+ svm.mda accur <- apply(as.matrix(2:10), 1, function(x)
+   lapply(test.id.set, svm accur.fun,
+     mds.subj, x) %>% unlist %>% mean)
+
+ # c). ISOMAP projected value
+ # library(vegan)
+ iso.score.k <- apply(as.matrix(2:20), 1,
function(neighbor) {
+   lsvt.isomap <- isomap(dist(scale(LSVT)),
+     ndim = 10, k = neighbor)$points
+   dim accur <- apply(as.matrix(2:10), 1, function(x)
+     lapply(test.id.set, svm accur.fun,
+       lsvt.isomap, x) %>% unlist %>% mean)
+   dim accur
+ })
+ svm.iso accur <- apply(iso.score.k, 1, max)
+
+ svm accur <- rbind(rep(NA, 3),
+   data.frame(PCA = svm.pc accur,
+     MDS = svm.mda accur,
+     ISOMAP = svm.iso accur))
+ matplot(1 - svm accur, pch = c(1, 3, 2), type = "b",
+   col = 2:4, lwd = 2, cex = c(2, 1.5, 1.2),
+   main = paste("\n", algor.num, " algorithms"),
+   xlab = "Dimensions", ylab = "Error rate")
+ }
> # Before feature selection
> algor.DR.method.select.svm(dim(d.algor.ord)[2])
> legend("topright", c("PCA", "MDS", "ISOMAP"),
+   pch = c(1, 3, 2), lwd = 2, col = 2:4,

```

```

+       lty = 2:4, pt.cex = c(2, 1.5, 1.2))
> # After feature selection (ISOMAP is always the best)
> par(mfrow = c(3, 3))
> par(mai = c(0.6, 0.6, 0.5, 0.2))
> svm.algor.DR.method.select <-
+   apply(as.matrix(c(20, 40, 60, 80, 100, 150, 200, 250,
309)),
+       1, algor.DR.method.select.svm)
> legend("top", c("PCA", "MDS", "ISOMAP"),
+       pch = c(1, 3, 2), lwd = 2, col = 2:4, horiz = T,
+       lty = 1:3, pt.cex = c(1.5, 1.2, 1.2), cex = 0.8,
+       bg = cm.colors(1, alpha = 0), bty = "n")
> title("\nError Rate by SVM upon Variables Selection",
+       outer = T)
>

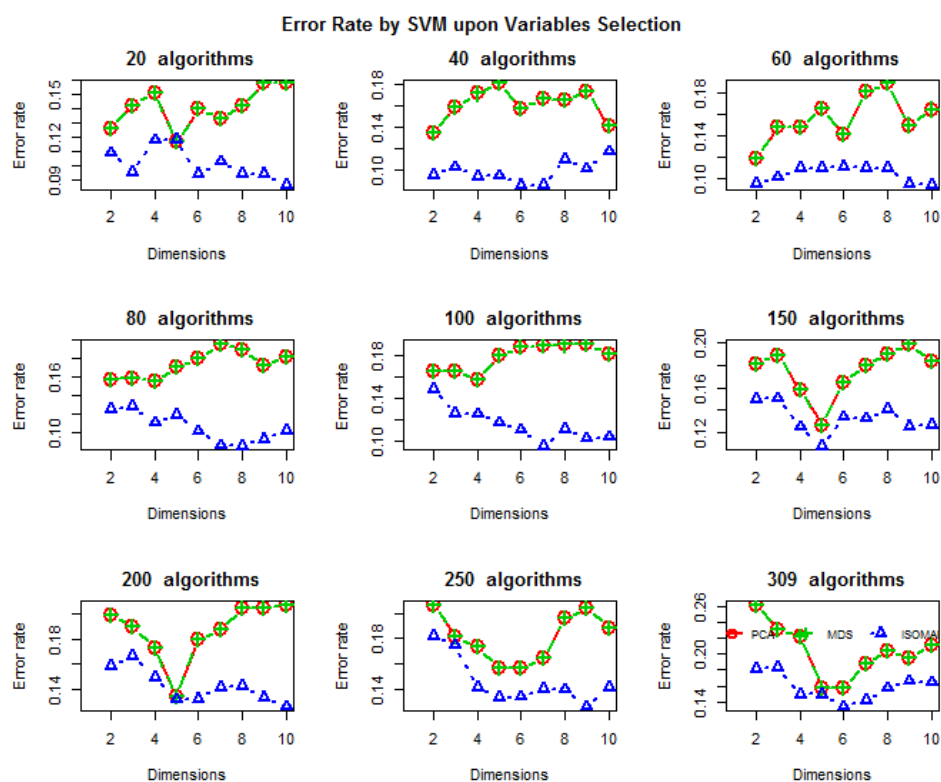
```



同樣地，將 309 個訊號演算法以主成分分析、多元尺度法分析及 ISOMAP 維度縮減後，分別取前 2 至 10 個維度作 SVM，預測 129 個治療後的聲音是否有改善，進一步比對 SVM 的預測與專家判斷的結果，探究預測的錯誤率大小。

無論所取的維度大小為何，ISOMAP 的分類錯誤率均比 PCA 與 MDS 小，其中又以取 ISOMAP 前 6 個維度的錯誤率最小。

然而，此 309 個信號演算法並非所有都對治療是否改善的結果具顯著的區辨力，因此需做變數的篩選，篩選出—在不同改善結果，演算值差異較大；在相同的結果下，差異較小。故將所有變數依照群間、群內差異比值由大至小排序，選擇群間、群內變異比值較大的變數。以下為不同變數篩選個數下，經過維度縮減後，SVM 的分類錯誤率：



分別篩選前 20、40、60、80、100、200、250 及 309 個最具區辨力的變數，即群間、群內變異比值最大的變數，再分別利用 PCA、MDS 與 ISOMAP 維度縮減後的維度去做分類，分類錯誤率如上圖。由圖可知，無論選擇變數的多寡、維度的多寡，ISOMAP 的分類錯誤率幾乎是最小的。因此 ISOMAP 為當中最適合的維度縮減方法。

以下接著將上圖中不同篩選個數—ISOMAP 所取各個維度的錯誤率繪製在一起，討論較適合的變數篩選個數。

```
> # 3. Feature selection
> # Compare isomap error rate in diff. number of var.
selected
> svm.algor.isomap.select.fun <- function(algor.num) {
+   LSVT <- d.algor.ord[ , 1:algor.num]
+   # c). ISOMAP projected value
+   iso.score.k <- apply(as.matrix(2:20), 1,
function(neighbor) {
+     lsvt.isomap <- isomap(dist(scale(LSVT)),
+                           ndim = 10, k = neighbor)$points
+     dim accur <- apply(as.matrix(2:10), 1, function(x)
+       lapply(test.id.set, svm accur.fun,
+             lsvt.isomap, x) %>% unlist %>% mean)
+     dim accur
+   })
+   svm.iso accur <- apply(iso.score.k, 1, max)
+   svm.iso accur
+ }
> svm.algor.isomap.select <-
+   rbind(rep(NA, 9),
+         apply(as.matrix(c(20, 40, 60, 80, 100,
+                           150, 200, 250, 309)),
+               1, svm.algor.isomap.select.fun))
>
> library(fields)
> matplot(1-svm.algor.isomap.select, pch = 16,
+         type = "l", lwd = 4:3, lty = 2:1,
+         xlim = c(0.5, 10.5), ylim = c(0.06, 0.28),
+         col = tim.colors(9), xlab = "Dimensions",
+         ylab = "Error rate",
+         main = "ISOMAP Classification Error Rate (SVM)",
+         xaxt = "n")
> axis(1, at = 2:10, 2:10)
> legend("topright", legend = rev(c(20, 40, 60, 80, 100,
150,
+                                   200, 250, 309))),
```

```

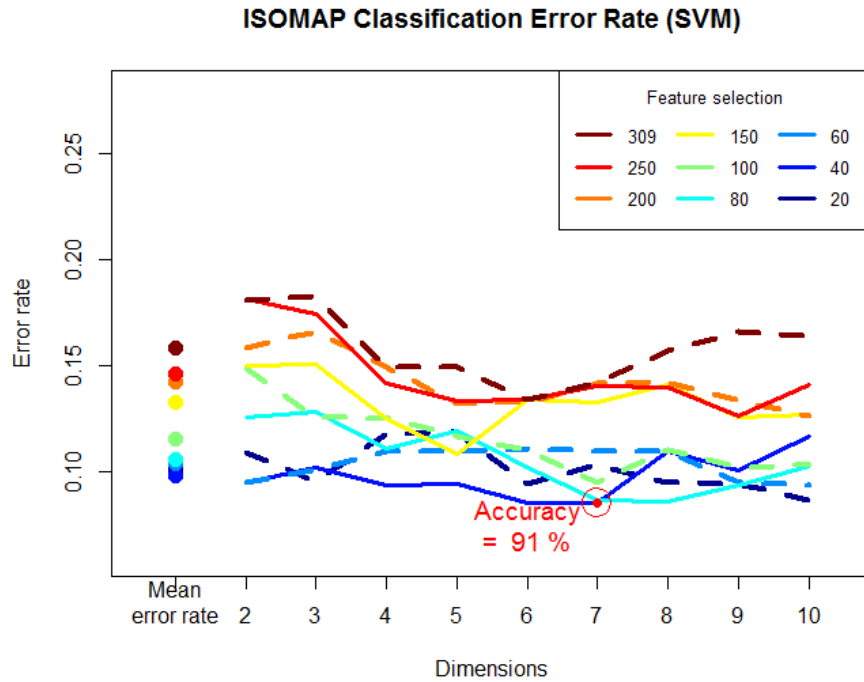
+         lwd = 3, col = rev(tim.colors(9)), ncol = 3,
+         title = "Feature selection", cex = 0.8)
> points(rep(7, 2),
+         rep(1-svm.algor.isomap.select[7, 2], 2),
+         pch = c(16, 1), col = 2, cex = c(1, 3)) #
(Continue...)
> # Thus, the first 40 sig. var.(features) are selected
> # to classify trt. consequences (SVM)
> # after Dimension Reduction (ISOMAP dim.7).
>
> # 4. Data that contains only the 40 most sig. var.
> # data.sig.var <- d.algor.ord[ , 1:40]
>
> # 5. ISOMAP dim.7, SVM accuracy
> ## Isomap dim.1-7
> # a). Find the most proper k(# of neighbors)
> svm accur.given.k <- apply(as.matrix(2:20), 1,
function(x) {
+   iso.pts <- isomap(dist(scale(data.sig.var)),
+                       ndim = 7, k = x)$points
+   accuracy <- lapply(test.id.set, svm accur.fun,
+                       iso.pts, 7) %>% unlist %>% mean
+   accuracy
+ })
> svm.best.k <- (2:20)[which.max(svm accur.given.k)]
>
> # b). Isomap
> svm.lsvt.isomap <- isomap(dist(scale(data.sig.var)),
+                             ndim = 7, k = svm.best.k)
> # c). classification by SVM
> tenfold.svm.pred.class <- lapply(test.id.set,
function(test.id) {
+   data <- svm.lsvt.isomap$points
+   x.train <- data[-test.id, ]
+   x.test <- data[test.id, ]
+   y.train <- class.f[-test.id]
+   y.test <- class.f[test.id]
+   train.set <- data.frame(x.train, y.train)
+
+   svm.model <- svm(y.train ~ ., data = train.set)

```

```

+   pred <- predict(svm.model, x.test)
+   pred
+ })
> svm.pred.class[unlist(test.id.set)] <-
+   unlist(tenfold.svm.pred.class)
> # svm.pred.class
>
> svm.iso.best accur <- mean(svm.pred.class ==
+                             class) # 91.27%
>
> # (continue...accuracy written in)
> text(6, 1-svm.algor.isomap.select[7, 2]-0.01,
+      paste("Accuracy\n= ", round(100*svm.iso.best accur),
+            "%"),
+      col = 2, cex = 1.2)
> # Mean error rate in Different numbers of var. selected
> svm.mean.err.rate <- 1-apply(svm.algor.isomap.select[-
1, ], 2, mean)
> par(new = T)
> plot(rep(1, 9), svm.mean.err.rate,
+      col = tim.colors(9), pch = 16, cex = 1.5,
+      xlab = "", ylab = "", xaxt = "n", yaxt = "n",
+      bty = "n", xlim = c(0.5, 10.5), ylim = c(0.06, 0.28))
> axis(1, at = 1, "Mean\nerror rate", tick = "n")
>

```



隨著最具區辨力的變數個數從 309、250、……篩選至 60、40 個，做維度縮減後分類，錯誤率會下降，但篩選至 20 個時錯誤率又上升，再次說明有部分的變數—信號演算法不具足夠的區辨力，以至於考量那些演算法的值時會使得整體分類錯誤率增加。

由圖所示，變數篩選前 40 個最具區辨力的演算法，再做 ISOMAP 降維，並取前 7 個維度，此時預測發聲是否改善的錯誤率最低，僅 9%，即有 91% 的預測準確率，結果與判別分析方法極為接近。

3.3.3 治療後有(無)改善的聲音特徵

在 3.3.2.1 節提到了經篩選出前 40 個最具區辨力的演算法，再透過 ISOMAP 前 7 個維度的值，能使得錯判率最低。除了找到使得錯誤率最小化的分類預測方法，另外，甚麼樣特性的聲音會被預測為治療有或沒有改善也十分重要。

在此節掌握了有無改善的聲音的主要特徵後，下一節將提到可以

在預測某些型態的聲音治療後沒有改善後，藉由維度縮減最具區辨力的演算法的值，更進一步判斷聲音不合格的因素；提供給 LSVT 療程專家作為未來治療調整的方向，故此節利用判別分析法的後驗機率先找出症狀有無改善的聲音的主要特徵。

一、 判別分析的後驗機率與 ISOMAP 維度投影值的關聯

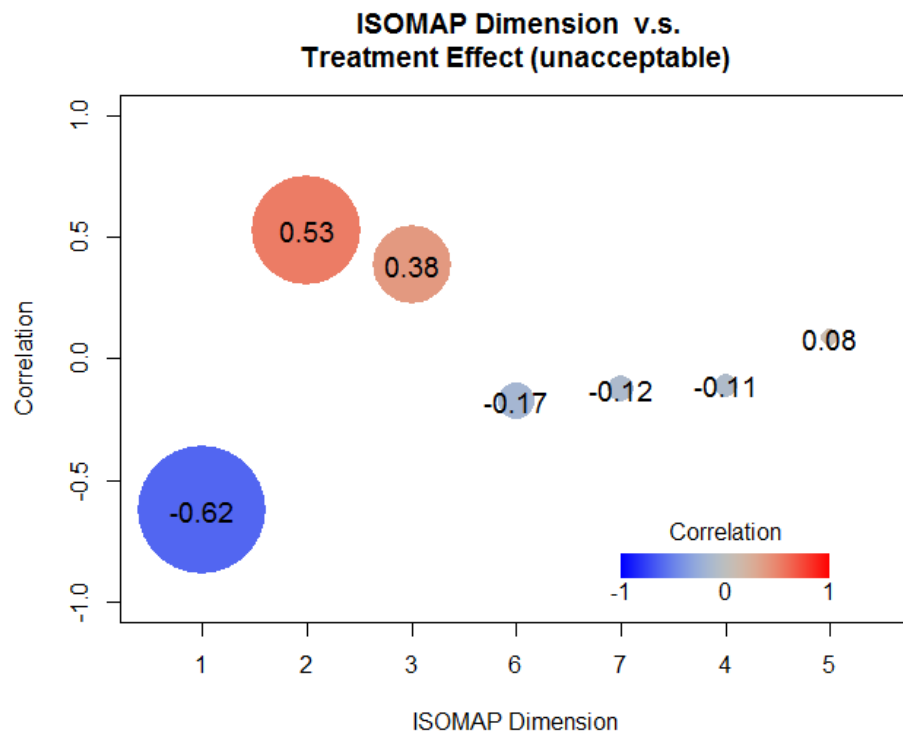
後驗機率是預測治療結果的依據，由其與 ISOMAP 7 個維度投影值的關聯性找出每個維度投影值大小、方向對預測結果的影響。

```
> # 6. Find the cause of accept./unaccept. of phonation
> # class(LDA posterior) => dim.val.(isomap$points)
> # => algorithm [measured by correlation coefficients]
>
> # a). class(LDA posterior) => dim.val.(isomap$points)
> # Effects of isomap$points (dim.)
> # on the class (LSVT treatment effect)
> # Measured by the corr. between isomap$points &
> # posterior prob.(the prob. that the phonation is
> # accept./unaccept.)
> cor.post.dim <- apply(lsvt.isomap$points, 2, function(x)
+ cor(x, lda.isomap$posterior[, 2]))
> ord.dim.eff <- order(abs(cor.post.dim), decreasing = T)
> reord.cor <- cor.iso.dim.post[ord.dim.eff]
>
> library(fields)
>
> col.red.blue <- two.colors(start = "blue",
+ middle = "gray",
+ end = "red")
> trans <- function(x) {round((x+1)*127.5)+1}
>
> plot(1:7, reord.cor, cex = 20 * (abs(reord.cor)),
+ xaxt = "n", xlim = c(0.5, 7.5), ylim = c(-1, 1),
+ xlab = "ISOMAP Dimension", ylab = "Correlation",
+ main = " ISOMAP Dimension v.s.
```

```

+ Treatment Effect (unacceptable)",
+   col = col.red.blue[trans(reord.cor)], pch = 16)
> axis(1, at = 1:7, ord.dim.eff)
> text(1:7, reord.cor, round(reord.cor, 2), cex = 1.2)
> # the post[2] bigger, the prob. of unaccept. larger.
> # so corr.(+), dim.val. big => unaccept.
> #      (-)              => accept.
> # legend
> rasterImage(t(col.red.blue), 5, -0.9, 7, -0.8)
> text(c(5, 6, 6, 7), c(-0.95, -0.7, -0.95, -0.95),
+      c(-1, "Correlation", 0, 1))
>

```



圖中數值表示沒有改善的聲音的後驗機率與 ISOMAP 各維度投影值的相關係數。其中前三個維度對預測治療結果的影響力最大，其與治療後症狀未改善的正反向關係如下：

維度	Dim. 1	Dim. 2	Dim. 3
V. S. 症狀未 改善	↓	↑	↑

聲音在 ISOMAP 第 1 維度的投影值越小或是在第 2、3 維度的投影值越大，表示該名患者該頻率、振幅的聲音越可能不被接受，即治療結果沒有成效越不明顯，症狀並未改善。

二、維度投影值與演算法的關聯及維度的定義

```

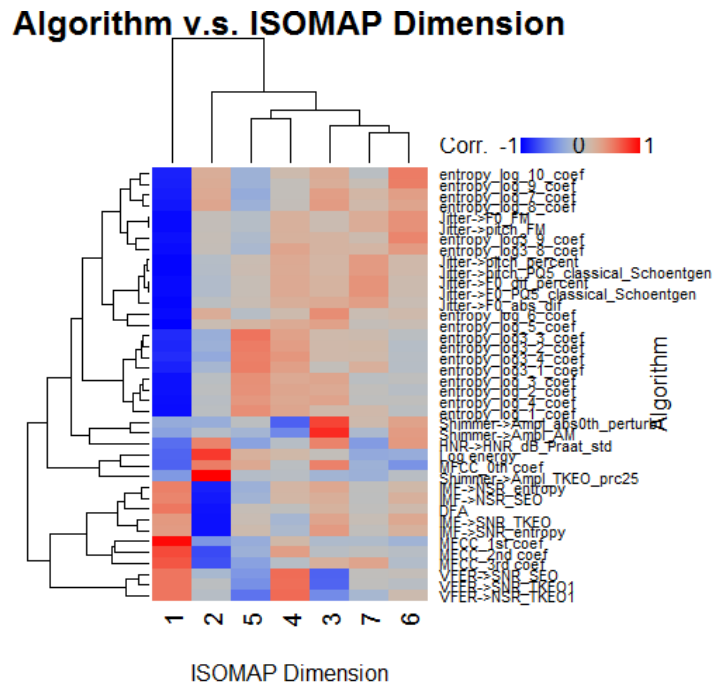
> # b). dim.val.(isomap$points) => algorithm
> #      Sig. algorithm
> cor.dim.var <- apply(data.sig.var, 2, function(x)
+   cor(x, lsvt.isomap$points))
> # heatmap
> heatmap(trans(t(cor.dim.var)),
+   col = col.red.blue,
+   Rowv = NULL,
+   margins = c(5, 10),
+   xlab = "ISOMAP Dimension", ylab = "Algorithm",
+   main = "Algorithm v.s. ISOMAP Dimension")
> # Medium or High Correlation is considered
> positive.var <- apply(cor.dim.var, 1, function(x)
+   names(which(x >= 0.3)))
> negative.var <- apply(cor.dim.var, 1, function(x)
+   names(which(x <= -0.3)))
>
> #####
> # Dim1: 信號能量的穩定與否，信號噪音比，
> #      小波波段 1 & 3，信號噪音比，
> #      聲帶振動頻率、振幅離散(偏離)趨勢
> # + MFCC_1st,2nd,3rd, IMF, DFA
> # - entropy_log_1~10, entropy_log3_1~5,8,9, HNR,
> # Jitter->F0/pitch, Shimmer->Ampl

```

```

>
> # Dim2: 信號能量的穩定與否, 振幅偏離趨勢
> #          信號能量的穩定與否, 信噪比, 小波波段 3 前段, 振幅
> # + MFCC_0th, Log energy, Shimmer->Ampl_TKEO_prc25
> # - MFCC_2nd,3rd, , IMF, DFA, entropy_log3_2~4
> #   Shimmer->Ampl
>
> # Dim3: 信號能量的穩定與否, 信噪比, 振幅
> #          信噪比
> # + MFCC_0th, IMF, Shimmer->Ampl_AM
> # - VFER->SNR_SEO,SNR_TKEO1,NSR_TKEO1
>
> # Dim4: 信噪比
> #          振幅偏離
> # + VFER->NSR_TKEO1
> # - Shimmer->Ampl_AM, abs0th_perturb
>
> # Dim5: 信噪比
> # + x
> # - VFER->SNR_SEO,SNR_TKEO1,NSR_TKEO1
>
> # Dim6:
> # + x
> # - x
>
> # Dim7:
> # + x
> # - x
> #####
>

```



前述首三個維度對預測治療結果的影響力最大，又由各個維度與演算法關聯性的熱圖看出此三個維度受某些演算法影響特別大，因此主宰前三個維度投影值的演算法便是影響預測結果的主要因素，換言之，這些主宰的演算法衡量的發聲特徵便是判斷治療結果是否改善的主要依據。

以下表格記錄主宰前三個維度的演算法與其意義，並歸納維度的意義，與症狀未改善的關係：

維度	Dim. 1		Dim. 2		Dim. 3	
方向	+	-	+	-	+	-
演算法	MFCC_1 st , 2 nd , 3 rd	entropy_1 og_1, 3 Jitter- >F0/pitch	MFCC_0 th	IMF DFA	Shimmer- >Ampl	VFER

說明	信號能量的穩定性	小波波段 1 & 3 波動, 聲帶振動 頻率偏離 (不穩定)	信號能量的不穩定性	信號噪音比	聲帶振動 振幅偏離 (不穩定)	信號噪音比
維度 意義	波動穩定性 能量穩定性 週期性(音調)		能量波動性 噪音含量(聲帶閉合不全)		週期偏離性(音量) 噪音含量(聲帶閉合不全、咬合)	
維度 定義	能量、音調持續穩定性		噪音(音調、音量、聲帶)		音量不穩定性	
V. S. 症狀未 改善	↓		↑		↑	

經由各個維度的主宰演算法變數，ISOMAP 第一個維度衡量聲音能量、音調的持續穩定性；第二個維度衡量來自於音量、音調無法控制及聲帶閉合不全帶來的噪音問題；第三個維度則是衡量音量的波動性。

3.3.3.1 判別分析下的 ISOMAP 投影值

```

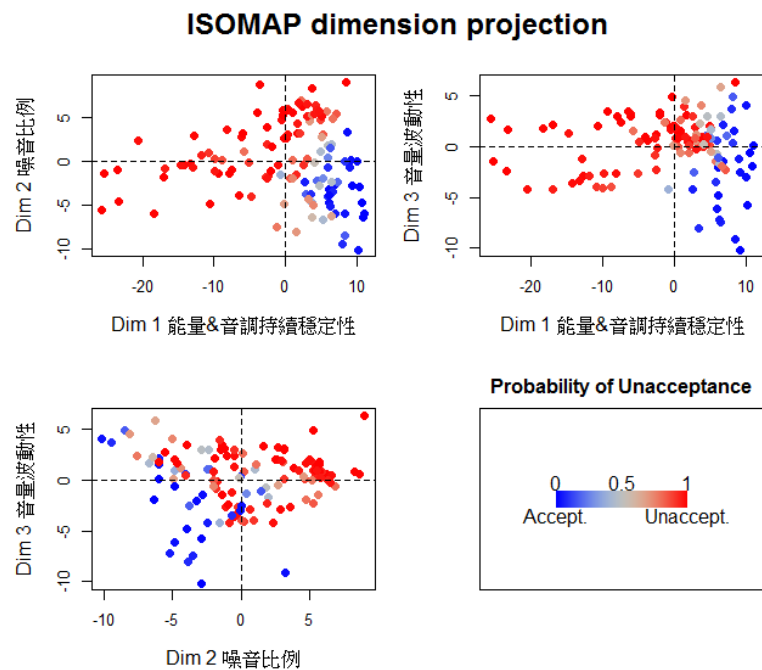
> # 7. scores
> iso.dim.def <- c("能量&音調持續穩定性",
+                  "噪音比例", "音量波動性")
> # a). 2D, risk (probability) of unaccept.
> par(mfrow = c(2, 2))
> par(mai = c(0.7, 0.7, 0.6, 0.2))
> apply(matrix(c(1, 1, 2, 2, 3, 3), ncol = 2), 1,
+        function(x) {

```

```

+       plot(lsvt.isomap$points[ , x[1]],
+           lsvt.isomap$points[ , x[2]],
+           col = col.red.blue[
+               trans(lda.isomap$posterior[ , 2]*2-1)],
+           pch = 16, cex = 1.2,
+           xlab = paste("Dim", x[1], iso.dim.def[x[1]]),
+           ylab = paste("Dim", x[2], iso.dim.def[x[2]]),
+           cex.lab = 1.2)
+       abline(h = 0, v = 0, lty = 2)
+   }
+ )
NULL
> plot(0, xlab = "", ylab = "", type = "n", xaxt = "n",
+     yaxt = "n", main = "\nProbability of Unacceptance",
+     xlim = c(-1, 1), ylim = c(-1, 1))
> rasterImage(t(col.red.blue), -0.5, -0.1, 0.5, 0.1)
> text(c(-0.5, 0, 0.5, -0.5, 0.5),
+     c(0.2, 0.2, 0.2, -0.2, -0.2), cex = 1.2,
+     c(0, 0.5, 1, "Accept.", "Unaccept."))
> title("\nISOMAP dimension projection",
+     outer = T, cex.main = 1.8)
>

```



聲音能量、音調持續地穩定是最好判斷患者治療狀況有無改善的準則，當治療後聲音能量、音調仍無法持續穩定，治療被預測為無效的機率很高；然而當能量、音調、音量都很穩定，噪音比例卻很高時，治療結果也不算有效。音量大小的控制能力則難以被用來評估治療是否有效。

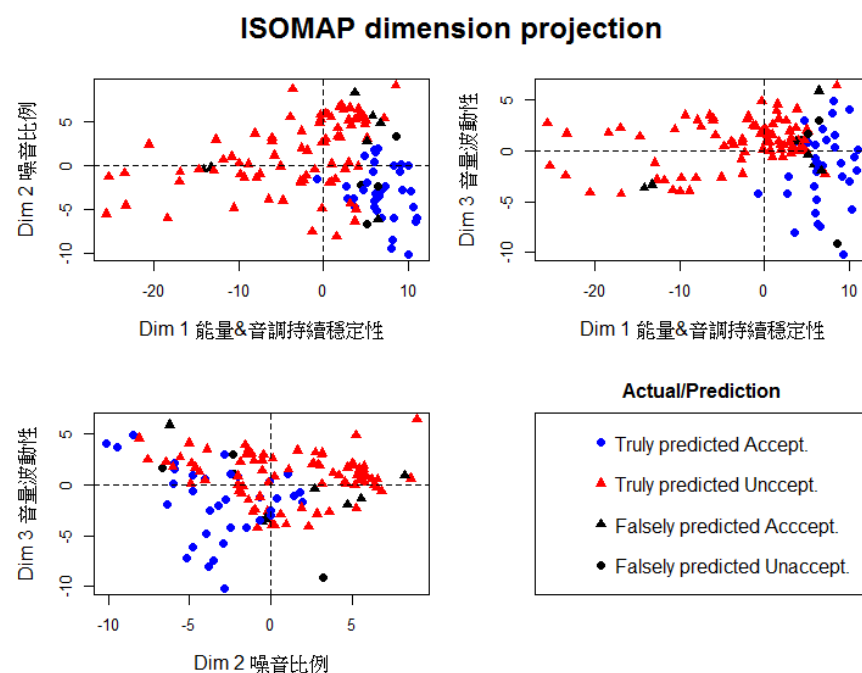
```
> # b). 2D, Actual/pred. class
> pred.class <- c(1, 2)[lda.isomap$class]
> actual.pred.class <- ifelse((class == 1 &
+                             pred.class == 2),
+                             3, ifelse((class == 2 &
+                             pred.class == 1),
+                             4, class))
> #   actual pred
> # 1 accept accept
> # 2 u       u
> # 3 a       u
> # 4 u       a
>
> par(mfrow = c(2, 2))
> par(mai = c(0.7, 0.7, 0.6, 0.2))
> apply(matrix(c(1, 1, 2, 2, 3, 3), ncol = 2), 1,
+        function(x) {
+          plot(lsvt.isomap$points[, x[1]],
+               lsvt.isomap$points[, x[2]],
+               col = c(4, 2, 1, 1)[actual.pred.class],
+               pch = c(16, 17, 17, 16)[actual.pred.class],
+               cex = 1.2, cex.lab = 1.2,
+               xlab = paste("Dim", x[1], iso.dim.def[x[1]]),
+               ylab = paste("Dim", x[2], iso.dim.def[x[2]]))
+          abline(h = 0, v = 0, lty = 2)
+        })
> )
NULL
> plot(0, xlab = "", ylab = "", type = "n", xaxt = "n",
+       yaxt = "n", main = "\nActual/Prediction")
```



```

> legend("center", c("Truly predicted Accept.",
+                    "Truly predicted Uncept.",
+                    "Falsely predicted Acccept.",
+                    "Falsely predicted Unaccept."),
+       col = c(4, 2, 1, 1), pch = c(16, 17, 17, 16),
+       bty = "n", pt.cex = 1.2, cex = 1.2)
> title("\nISOMAP dimension projection",
+       outer = T, cex.main = 1.8)
>

```



被預測錯誤的聲音中，幾乎都是屬於能量、音調持續穩定的聲音。少部分聲音雖然能量、音調穩定、噪音含量高，但其實是屬於有改善的聲音；部分聲音穩定、噪音含量低，卻屬於治療沒有效果的聲音。

3.3.3.2 SVM 下的 ISOMAP 投影值

```

> # 7. scores (colored by SVM 10-fold CV)
> # iso.dim.def <- c("能量&音調持續穩定性",

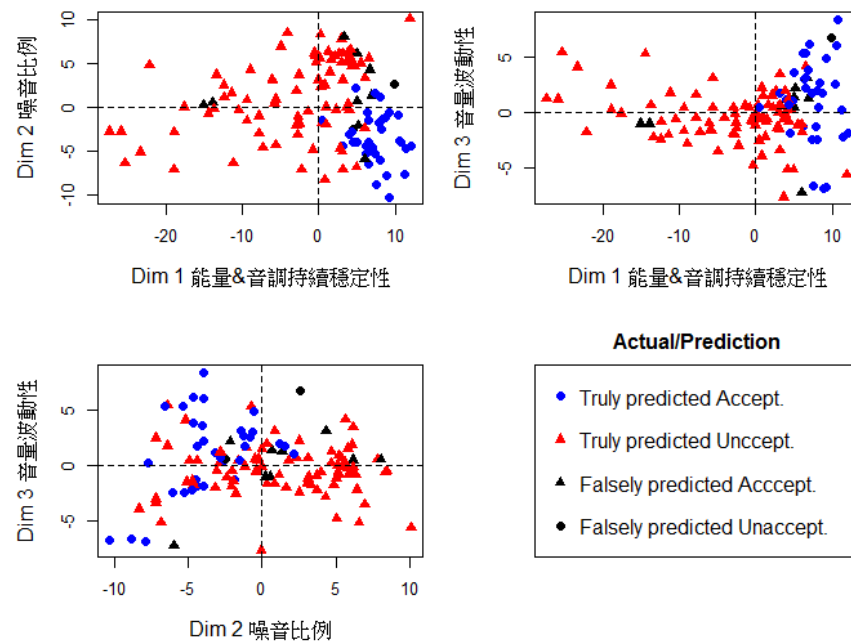
```

```

> # "噪音比例", "音量波動性")
>
> # 2D, Actual/pred. class
> svm.pred.actual <- ifelse((class == 1 &
+                           svm.pred.class == 2),
+                           3, ifelse((class == 2 &
+                           svm.pred.class == 1),
+                           4, class))
> # actual pred
> # 1 accept accept
> # 2 u      u
> # 3 a      u
> # 4 u      a
>
> par(mfrow = c(2, 2))
> par(mai = c(0.7, 0.7, 0.6, 0.2))
> apply(matrix(c(1, 1, 2, 2, 3, 3), ncol = 2), 1,
+       function(x) {
+         plot(svm.lsvt.isomap$points[ , x[1]],
+              svm.lsvt.isomap$points[ , x[2]],
+              col = c(4, 2, 1, 1)[svm.pred.actual],
+              pch = c(16, 17, 17, 16)[svm.pred.actual],
+              cex = 1.2, cex.lab = 1.2,
+              xlab = paste("Dim", x[1], iso.dim.def[x[1]]),
+              ylab = paste("Dim", x[2], iso.dim.def[x[2]]))
+         abline(h = 0, v = 0, lty = 2)
+       })
+ )
NULL
> plot(0, xlab = "", ylab = "", type = "n", xaxt = "n",
+      yaxt = "n", main = "\nActual/Prediction")
> legend("center", c("Truly predicted Accept.",
+                    "Truly predicted Uncccept.",
+                    "Falsely predicted Acccept.",
+                    "Falsely predicted Unaccept."),
+      col = c(4, 2, 1, 1), pch = c(16, 17, 17, 16),
+      bty = "n", pt.cex = 1.2, cex = 1.2)
> title("\nISOMAP dimension projection(SVM)",
+      outer = T, cex.main = 1.8)
>

```

ISOMAP dimension projection(SVM)



SVM 分類的結果也與判別分析極為相近，被預測錯誤的聲音中，幾乎都是屬於能量、音調持續穩定的聲音。少部分聲音雖然能量、音調穩定、噪音含量高，但其實是屬於有改善的聲音；部分聲音穩定、噪音含量低，卻屬於治療沒有效果的声音。

3.3.4 PD 患者發聲治療後未改善的症狀

由前一節知道不合格的聲音可能是因為能量、音調、音量的不穩定，也可能是噪音含量高，此節目的在於一預測聲音治療後沒有改善後，更進一步判斷該聲音不合格的主要因素，能提供給 LSVT 療程專家作為未來治療調整的方向。

```
> # 8. Unaccept. phonation symptom type
> #   unaccept. isomap$points clustered
> # a). Unaccept. phonation isomap$points
```

```

> unaccept.iso.pt <- lsvt.isomap$points[
+   which(class == 2), ]
>
> # b). Clustering Measurment
> library(clValid)
> intern <- clValid(unaccept.iso.pt, 2:10,
+                   clMethods = c("hierarchical",
+                                 "kmeans", "pam"),
+                   validation = "internal")
> # summary(intern) # view results
> stab <- clValid(unaccept.iso.pt, 2:10,
+                 clMethods = c("hierarchical",
+                               "kmeans", "pam"),
+                 validation = "stability")
> # summary(stab)
>
> # c). Hierarchical, K = 2
> hclust.iso.pts <- hclust(dist(unaccept.iso.pt),
+                           members = NULL)
> # Clustered isomap pts plot, k = 2
> cut.h.cluster <- cutree(hclust.iso.pts, k = 2)
>
> par(mfrow = c(2, 2))
> par(mai = c(0.7, 0.7, 0.6, 0.2))
> apply(matrix(c(1, 1, 2, 2, 3, 3), ncol = 2), 1,
+        function(x) {
+          plot(unaccept.iso.pt[ , x[1]],
+               unaccept.iso.pt[ , x[2]],
+               col = c("purple", "orange")[cut.h.cluster],
+               pch = 16, cex = 1.2,
+               xlab = paste("Dim", x[1], iso.dim.def[x[1]]),
+               ylab = paste("Dim", x[2], iso.dim.def[x[2]]),
+               cex.lab = 1.2)
+          abline(h = 0, v = 0, lty = 2)
+        })
NULL
> plot(0, xlab = "", ylab = "", type = "n",
+       xaxt = "n", yaxt = "n", main = "",
+       xlim = c(-1, 1), ylim = c(-1, 1))

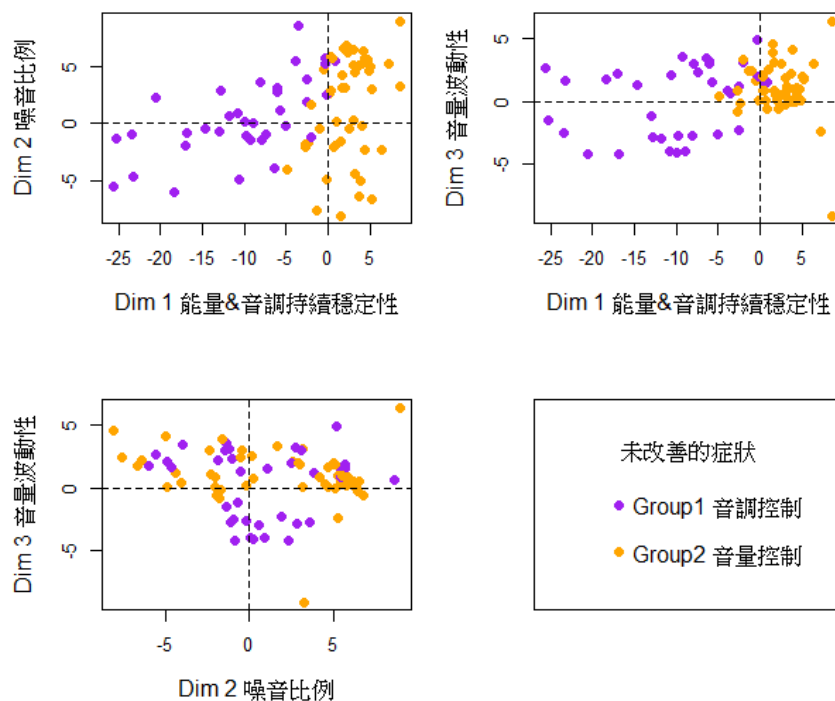
```

```

> legend("center", c("Group1 音調控制",
+                    "Group2 音量控制"),
+       title = "未改善的症狀", pch = 16,
+       col = c("purple", "orange"), bty = "n",
+       cex = 1.2, pt.cex = 1.2)
> title("\nISOMAP dimension projection (Unacceptance)",
+       outer = T, cex.main = 1.8)
>

```

ISOMAP dimension projection (Unacceptance)



被預測為治療後沒有改善的聲音，依照內外部效度、可靠度作層次法分群，共分為兩群。第一群聲音的能量、音調不穩定的症狀較為嚴重，第二群則是音量控制，未來療程能首先個別針對音調、音量的控制加以治療。

3.3.5 分類—SVM

繼前一節利用 Leave-one-out LDA 以及 10-Fold SVM 分類後，此節則運用考慮抽樣放回、重複試驗 100 次的 10-Fold SVM，以及 Leave-one-out SVM 分類，同樣地期望經由變數的篩選以及維度縮減使得分類錯誤率最小化。

3.3.5.1 重複試驗 100 次的 10-Fold SVM

一、分類錯誤率最小化下的變數選取以及維度縮減方法

```
> ## Feature & Dimension Reduction method Selection
> # (criterion: Max. classification(SVM) accuracy)
>
> # 1. Data updated by
> #   Ordered significant variables (algorithm)
>
> # d.algor.ord <- data.frame(data[, order.bw])
>
> # 2. SVM, 10-fold CV, function
> library(caret)
> set.seed(12345)
> test.id.set.rep <- apply(as.matrix(1:100), 1, function(x)
+   createFolds(y = 1:dim(data)[1], k = 10))
>
> library(e1071)
> svm accur.fun <- function(test.id, data, dim) {
+   x.train <- data[-test.id, 1:dim]
+   x.test <- data[test.id, 1:dim]
+   y.train <- class.f[-test.id]
+   y.test <- class.f[test.id]
+   train.set <- data.frame(x.train, y.train)
+
+   svm.model <- svm(y.train ~ ., data = train.set)
+
+   pred <- predict(svm.model, x.test)
```

```

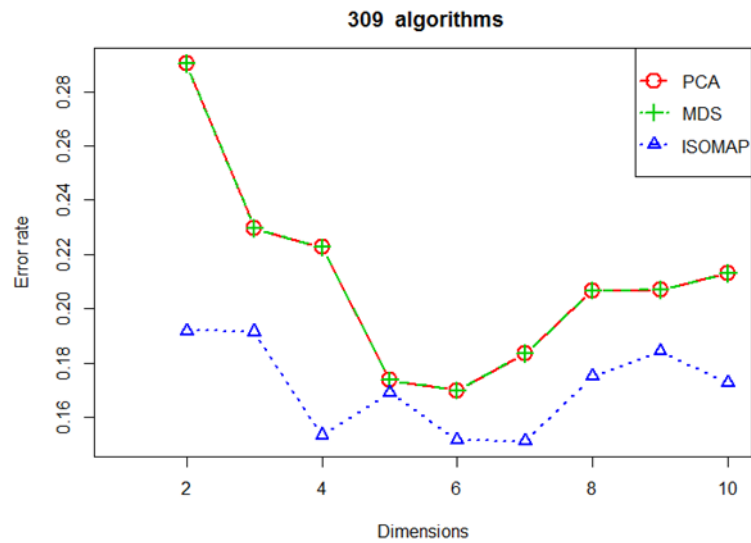
+   confus.mat <- table(pred, y.test)
+   accuracy <- sum(diag(prop.table(confus.mat)))
+   accuracy
+ }
>
> # 2. Dimension Reduction method Selection (matplot.)
> library(dplyr)
> algor.DR.method.select.svm <- function(algor.num) {
+   LSVT <- d.algor.ord[ , 1:algor.num]
+
+   # a). PC scores
+   # library(FactoMineR)
+   # library(ggplot2)
+   # library(factoextra)
+   lsvt.pca <- PCA(LSVT, scale.unit = T, graph = F,
+                   ncp = 10)
+   eig <- get_eigenvalue(lsvt.pca)
+   var <- get_pca_var(lsvt.pca)
+   ind <- get_pca_ind(lsvt.pca)
+   lsvt.pc.score <- ind$coord[ , 1:10]
+   rep.pc accur <- apply(as.matrix(1:100), 1,
function(rep)
+     apply(as.matrix(2:10), 1, function(x)
+       lapply(test.id.set.rep[[rep]], svm accur.fun,
+             lsvt.pc.score, x) %>% unlist %>% mean))
+   svm.pc accur <- apply(rep.pc accur, 1, mean)
+
+   # b). MDS projected value
+   mds.subj <- cmdscale(dist(scale(LSVT)), k = 10)
+   rep.mds accur <- apply(as.matrix(1:100), 1,
function(rep)
+     apply(as.matrix(2:10), 1, function(x)
+       lapply(test.id.set.rep[[rep]], svm accur.fun,
+             mds.subj, x) %>% unlist %>% mean))
+   svm.mds accur <- apply(rep.mds accur, 1, mean)
+
+   # c). ISOMAP projected value
+   # library(vegan)
+   iso.score.k <- apply(as.matrix(2:20), 1,
function(neighbor) {

```

```

+   lsvt.isomap <- isomap(dist(scale(LSVT)),
+                         ndim = 10, k = neighbor)$points
+   rep accur <- apply(as.matrix(1:100), 1, function(rep)
+     apply(as.matrix(2:10), 1, function(x)
+       lapply(test.id.set.rep[[rep]], svm accur.fun,
+             lsvt.isomap, x) %>% unlist %>% mean))
+   dim accur <- apply(rep accur, 1, mean)
+   dim accur
+ })
+ svm.iso accur <- apply(iso.score.k, 1, max)
+
+ svm accur <- rbind(rep(NA, 3),
+                   data.frame(PCA = svm.pc accur,
+                               MDS = svm.mds accur,
+                               ISOMAP = svm.iso accur))
+ matplot(1 - svm accur, pch = c(1, 3, 2), type = "b",
+         col = 2:4, lwd = 2, cex = c(2, 1.5, 1.2),
+         main = paste("\n", algor.num, " algorithms"),
+         xlab = "Dimensions", ylab = "Error rate")
+ }
> # Before feature selection
> algor.DR.method.select.svm(dim(data)[2])
> legend("topright", c("PCA", "MDS", "ISOMAP"),
+       pch = c(1, 3, 2), lwd = 2, col = 2:4,
+       lty = 2:4, pt.cex = c(2, 1.5, 1.2))
> par(mfrow = c(3, 3))
> par(mai = c(0.6, 0.6, 0.5, 0.2))
> rep.svm.algor.DR.method.select <-
+   apply(as.matrix(c(20, 40, 60, 80, 100, 150, 200, 250,
+ 309)),
+       1, algor.DR.method.select.svm)
>

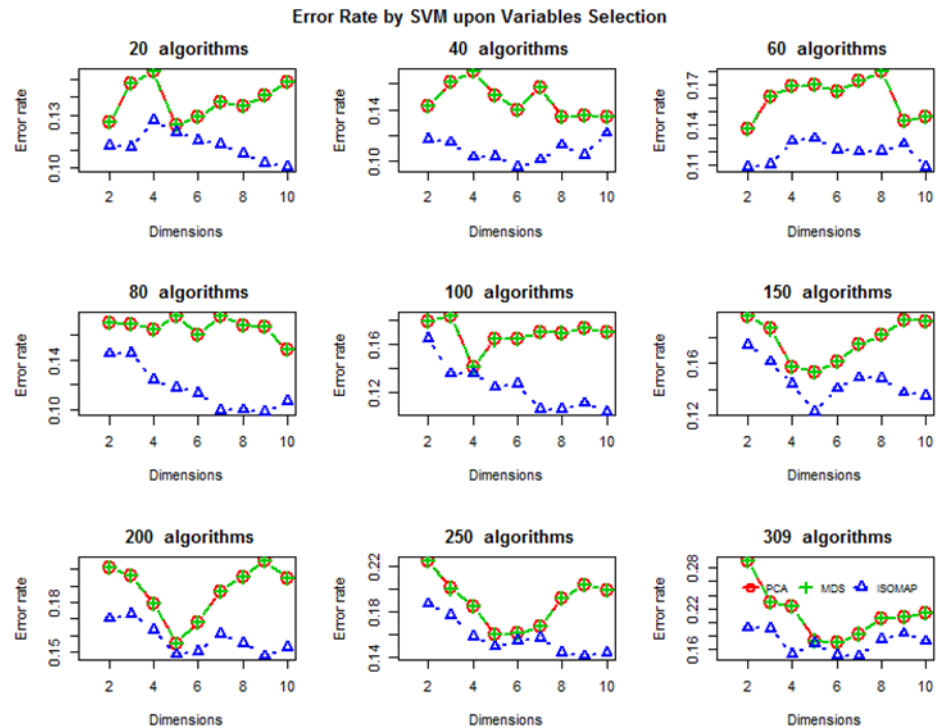
```

將總共 309 個訊號演算法以主成分分析、多元尺度法分析以及 ISOMAP 維度縮減後，分別取前 2 至 10 個維度作判別分析，預測 129 個發聲經治療後是否有改善，進一步比對判別分析的預測與專家判斷的結果，探究預測的錯誤率大小。

無論所取的維度大小為何，ISOMAP 的分類錯誤率均比 PCA 與 MDS 小，其中又以取 ISOMAP 前 7 個維度的錯誤率最小。

然而，此 309 個信號演算法並非所有都對治療是否改善的結果具顯著的區辨力，因此需做變數的篩選，篩選出—在不同改善結果，演算值差異較大；在相同的結果下，差異較小。故將所有變數依照群間、群內差異比值由大至小排序，選擇群間、群內變異比值較大的變數。以下為不同變數篩選個數下，經過維度縮減後，10-fold SVM 重複 100 次試驗的平均分類錯誤率：



先分別篩選前 20、40、60、80、100、200、250 及 309 個最具區辨力的變數，即群間、群內變異比值最大的變數，再分別利用 PCA、MDS 與 ISOMAP 維度縮減後的維度去做分類，分類錯誤率如上圖。由圖可知，無論選擇變數的多寡、維度的多寡，ISOMAP 的分類錯誤率幾乎是最小的。因此 ISOMAP 為當中最適合的維度縮減方法。

以下接著將上圖中不同篩選個數—ISOMAP 所取各個維度的錯誤率繪製在一起，討論較適合的變數篩選個數。

```
> # 3. Feature selection
> # Compare isomap error rate in diff. number of var.
  selected
> rep.svm.algor.isomap.select.fun <- function(algor.num) {
+   LSVT <- d.algor.ord[ , 1:algor.num]
+   # c). ISOMAP projected value
+   iso.score.k <- apply(as.matrix(2:20), 1,
+ function(neighbor) {
+   lsvt.isomap <- isomap(dist(scale(LSVT)),
```

```

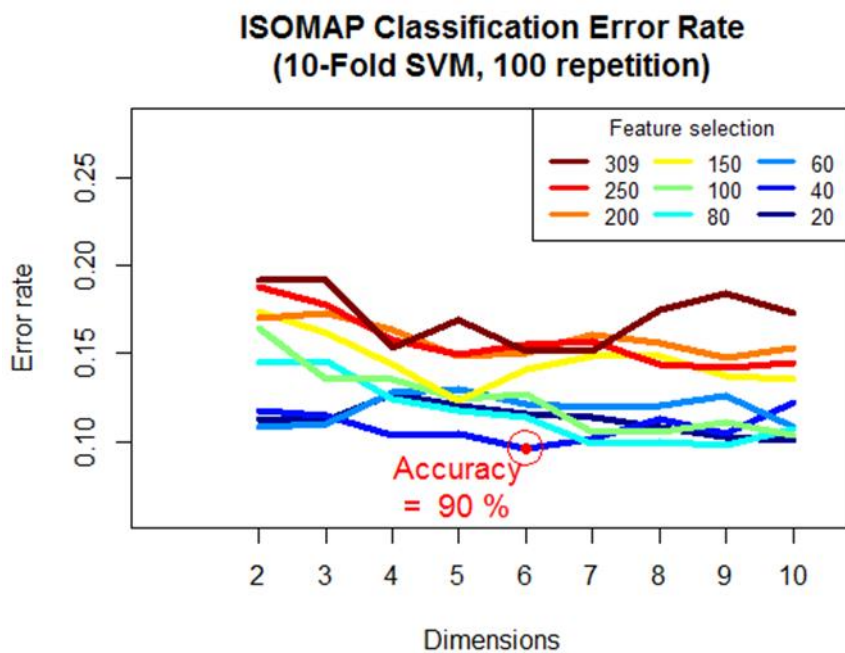
+             ndim = 10, k = neighbor)$points
+   rep accur <- apply(as.matrix(1:100), 1, function(rep)
+     apply(as.matrix(2:10), 1, function(x)
+       lapply(test.id.set.rep[[rep]], svm accur fun,
+         lsvt.isomap, x) %>% unlist %>% mean))
+   dim accur <- apply(rep accur, 1, mean)
+   dim accur
+ })
+ svm.iso accur <- apply(iso.score.k, 1, max)
+ svm.iso accur
+ }
> rep.svm.algor.isomap.select <-
+   rbind(rep(NA, 9),
+     apply(as.matrix(c(20, 40, 60, 80, 100,
+       150, 200, 250, 309)),
+       1, rep.svm.algor.isomap.select.fun))
>
> library(fields)
> matplot(1-rep.svm.algor.isomap.select, pch = 16,
+   type = "l", lwd = 4, lty = 1,
+   xlim = c(0.5, 10.5), ylim = c(0.06, 0.28),
+   col = tim.colors(9), xlab = "Dimensions",
+   ylab = "Error rate",
+   main = "ISOMAP Classification Error Rate
+ (10-Fold SVM, 100 repetition)",
+   xaxt = "n")
> axis(1, at = 2:10, 2:10)
> legend("topright", legend = rev(c(20, 40, 60, 80, 100,
150,
+       200, 250, 309)),
+   lwd = 3, col = rev(tim.colors(9)), ncol = 3,
+   title = "Feature selection", cex = 0.8)
> points(rep(6, 2),
+   rep(1-rep.svm.algor.isomap.select[6, 2], 2),
+   pch = c(16, 1), col = 2, cex = c(1, 3)) #
(Continue...)
> text(5, 1-rep.svm.algor.isomap.select[6, 2]-0.02,
+   paste("Accuracy\n= ",
+     round(100*(rep.svm.algor.isomap.select[6, 2])),
+   "%"),

```

```

+ col = 2, cex = 1.2)
> # Thus, the first 40 sig. var.(features) are selected
> # to classify trt. consequences (SVM)
> # after Dimension Reduction (ISOMAP dim.6).
>
> # 4. Data that contains only the 40 most sig. var.
> # data.sig.var <- d.algor.ord[ , 1:40]
>

```



隨著最具區辨力的變數個數從 309、250、……篩選至 60、40 個，做維度縮減後分類，錯誤率會下降，但篩選至 20 個時錯誤率又上升，再次說明有部分的變數—信號演算法不具足夠的區辨力，以至於考量那些演算法的值時會過度配適，整體分類錯誤率增加。

由圖所示，變數篩選前 40 個最具區辨力的演算法，再做 ISOMAP 降維，並取前 6 個維度，此時預測發聲是否改善的錯誤率最低，僅 10%，即有 90% 的預測準確率。

3.3.5.2 Leave-one-out 10-Fold SVM

一、分類錯誤率最小化下的變數選取以及維度縮減方法

```
> ## Feature & Dimension Reduction method Selection
> # (criterion: Max. classification(SVM) accuracy)
>
> # 1. Data updated by
> #   Ordered significant variables (algorithm)
>
> # d.algor.ord <- as.matrix(data)[ , order.bw]
>
> # 2. SVM, CV(Leave one out), function
> library(e1071)
> cv.svm accur.fun <- function(test.id, data, dim) {
+   x.train <- data[-test.id, 1:dim]
+   x.test <- data[test.id, 1:dim]
+   y.train <- class.f[-test.id]
+   y.test <- class.f[test.id]
+   train.set <- data.frame(x.train, y.train)
+
+   svm.model <- svm(y.train ~ ., data = train.set)
+
+   pred <- predict(svm.model, x.test)
+   confus.mat <- table(pred, y.test)
+   accuracy <- sum(diag(prop.table(confus.mat)))
+   accuracy
+ }
>
> # 2. Dimension Reduction method Selection (matplot.)
> library(dplyr)
> cv.algor.DR.method.select.svm <- function(algor.num) {
+   LSVT <- d.algor.ord[ , 1:algor.num]
+   # a). PC scores
+   # library(FactoMineR)
+   # library(ggplot2)
+   # library(factoextra)
+   lsvt.pca <- PCA(LSVT, scale.unit = T, graph = F,
+                   ncp = 10)
+   eig <- get_eigenvalue(lsvt.pca)
+   var <- get_pca_var(lsvt.pca)
```

```

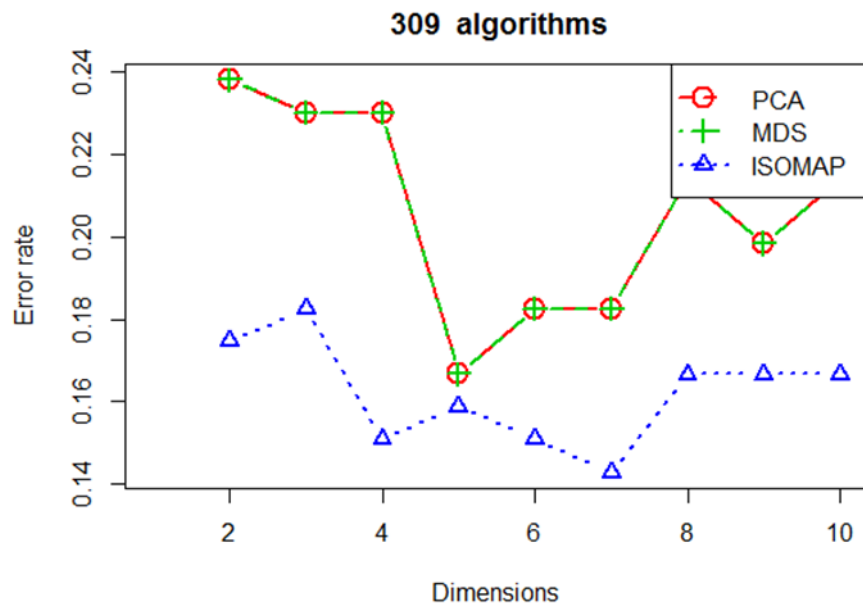
+   ind <- get_pca_ind(lsvt.pca)
+   lsvt.pc.score <- ind$coord[ , 1:10]
+   svm.pc accur <- apply(as.matrix(2:10), 1, function(x)
+     apply(as.matrix(1:dim(LSVT)[1]), 1, cv.svm.accur.fun,
+       data.frame(lsvt.pc.score), x) %>% mean)
+
+   # b). MDS projected value
+   mds.subj <- cmdscale(dist(scale(LSVT)), k = 10)
+   svm.mds.accur <- apply(as.matrix(2:10), 1, function(x)
+     apply(as.matrix(1:dim(LSVT)[1]), 1, cv.svm.accur.fun,
+       data.frame(mds.subj), x) %>% mean)
+
+   # c). ISOMAP projected value
+   # library(vegan)
+   iso.score.k <- apply(as.matrix(2:20), 1,
function(neighbor) {
+     lsvt.isomap <- isomap(dist(scale(LSVT)),
+       ndim = 10, k = neighbor)$points
+     dim.accur <- apply(as.matrix(2:10), 1, function(x)
+       apply(as.matrix(1:dim(LSVT)[1]), 1,
cv.svm.accur.fun,
+         data.frame(lsvt.isomap), x) %>% mean)
+     dim.accur
+   })
+   svm.iso.accur <- apply(iso.score.k, 1, max)
+
+   svm.accur <- rbind(rep(NA, 3),
+     data.frame(PCA = svm.pc.accur,
+       MDS = svm.mds.accur,
+       ISOMAP = svm.iso.accur))
+   matplot(1 - svm.accur, pch = c(1, 3, 2), type = "b",
+     col = 2:4, lwd = 2, cex = c(2, 1.5, 1.2),
+     main = paste("\n", algor.num, " algorithms"),
+     xlab = "Dimensions", ylab = "Error rate")
+ }
> # Before feature selection
> cv.algor.DR.method.select.svm(dim(d.algor.ord)[2])
> # After feature selection (ISOMAP is always the best)
> par(mfrow = c(3, 3))
> par(mai = c(0.6, 0.6, 0.5, 0.2))

```

```

> cv.svm.algor.DR.method.select <-
+   apply(as.matrix(c(20, 40, 60, 80, 100, 150, 200, 250,
+ 309)),
+         1, cv.algor.DR.method.select.svm)
>

```

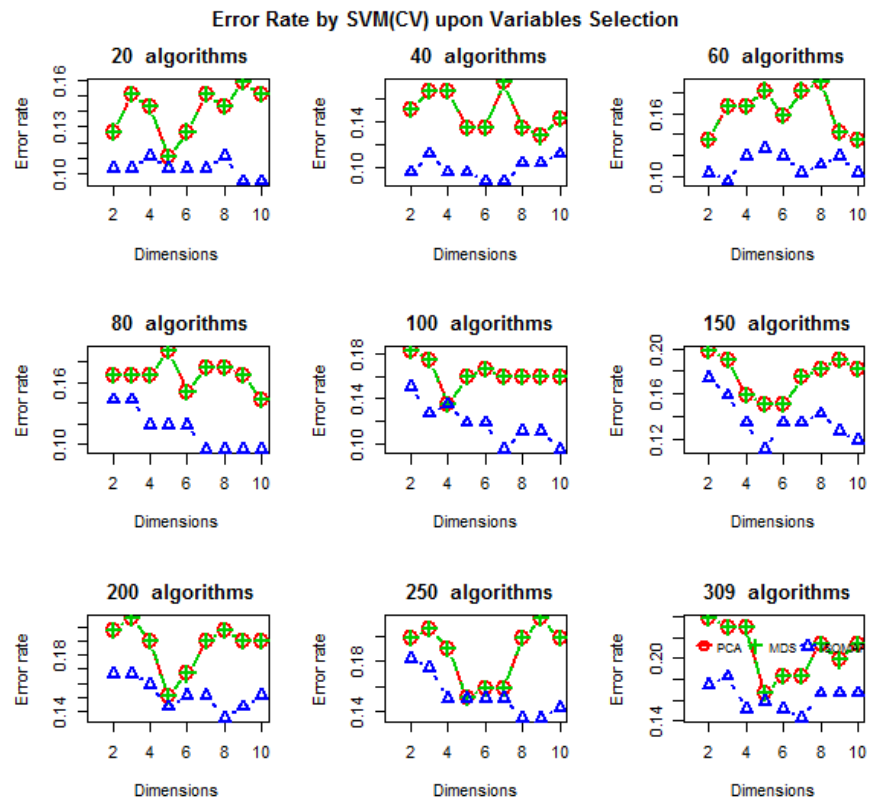


將總共 309 個訊號演算法以主成分分析、多元尺度法分析以及 ISOMAP 維度縮減後，分別取前 2 至 10 個維度作判別分析，預測 129 個發聲經治療後是否有改善，進一步比對判別分析的預測與專家判斷的結果，探究預測的錯誤率大小。

無論所取的維度大小為何，ISOMAP 的分類錯誤率均比 PCA 與 MDS 小，其中又以取 ISOMAP 前 7 個維度的錯誤率最小。

然而，此 309 個信號演算法並非所有都對治療是否改善的結果具顯著的區辨力，因此需做變數的篩選，篩選出—在不同改善結果，演算值差異較大；在相同的結果下，差異較小。故將所有變數依照群間、群內差異比值由大至小排序，選擇群間、群內變異比值較大的變

數。以下為不同變數篩選個數下，經過維度縮減後，Leave-one-out 10-Fold SVM 的平均分類錯誤率：



先分別篩選前 20、40、60、80、100、200、250 及 309 個最具區辨力的變數，即群間、群內變異比值最大的變數，再分別利用 PCA、MDS 與 ISOMAP 維度縮減後的維度去做分類，分類錯誤率如上圖。由圖可知，無論選擇變數的多寡、維度的多寡，ISOMAP 的分類錯誤率幾乎是最小的。因此 ISOMAP 為當中最適合的維度縮減方法。

以下接著將上圖中不同篩選個數—ISOMAP 所取各個維度的錯誤率繪製在一起，討論較適合的變數篩選個數。

```
> # 3. Feature selection
> # Compare isomap error rate in diff. number of var.
selected
```



```

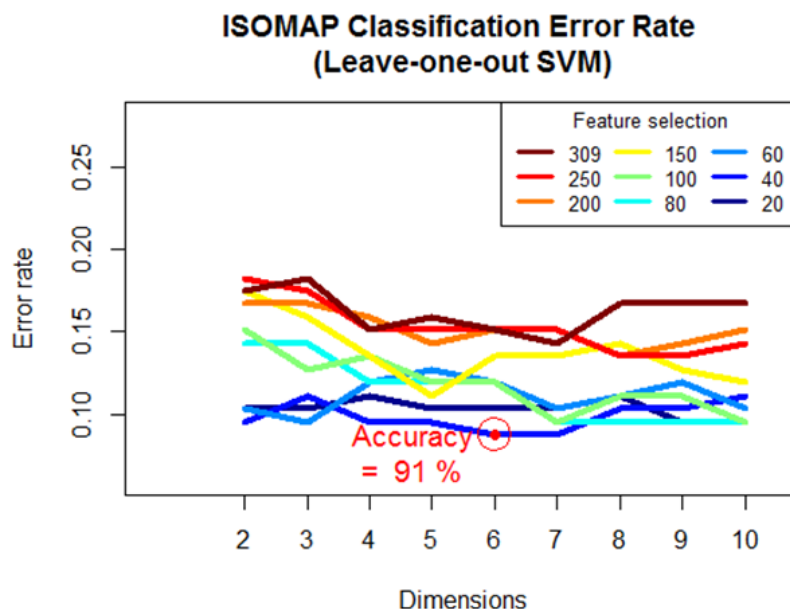
> cv.svm.algor.isomap.select.fun <- function(algor.num) {
+   LSVT <- d.algor.ord[ , 1:algor.num]
+   # c). ISOMAP projected value
+   iso.score.k <- apply(as.matrix(2:20), 1,
function(neighbor) {
+     lsvt.isomap <- isomap(dist(scale(LSVT)),
+                           ndim = 10, k = neighbor)$points
+     dim accur <- apply(as.matrix(2:10), 1, function(x)
+       apply(as.matrix(1:dim(LSVT)[1]), 1,
cv.svm accur.fun,
+         data.frame(lsvt.isomap), x) %>% mean)
+     dim accur
+   })
+   svm.iso accur <- apply(iso.score.k, 1, max)
+   svm.iso accur
+ }
> cv.svm.algor.isomap.select <-
+   rbind(rep(NA, 9),
+         apply(as.matrix(c(20, 40, 60, 80, 100,
+                           150, 200, 250, 309)),
+               1, cv.svm.algor.isomap.select.fun))
>
> library(fields)
> matplot(1-cv.svm.algor.isomap.select, pch = 16,
+         type = "l", lwd = 4, lty = 1,
+         xlim = c(0.5, 10.5), ylim = c(0.06, 0.28),
+         col = tim.colors(9), xlab = "Dimensions",
+         ylab = "Error rate",
+         main = "ISOMAP Classification Error Rate
+ (Leave-one-out SVM)",
+         xaxt = "n")
> axis(1, at = 2:10, 2:10)
> legend("topright", legend = rev(c(20, 40, 60, 80, 100,
150,
+                               200, 250, 309)),
+       lwd = 3, col = rev(tim.colors(9)), ncol = 3,
+       title = "Feature selection", cex = 0.8)
> points(rep(6, 2),
+        rep(1-cv.svm.algor.isomap.select[6, 2], 2),
+        pch = c(16, 1), col = 2, cex = c(1, 3)) #

```

```

(Continue...)
> text(4.7, 1-cv.svm.algor.isomap.select[6, 2]-0.01,
+     paste("Accuracy\n= ",
round(100*cv.svm.algor.isomap.select[6, 2]), "%"),
+     col = 2, cex = 1.2)
> # Thus, the first 40 sig. var.(features) are selected
> # to classify trt. consequences (SVM)
> # after Dimension Reduction (ISOMAP dim.6).
>
> # 4. Data that contains only the 40 most sig. var.
> # data.sig.var <- d.algor.ord[ , 1:40]
>

```



隨著最具區辨力的變數個數從 309、250、……篩選至 60、40 個，做維度縮減後分類，錯誤率會下降，但篩選至 20 個時錯誤率又上升，再次說明有部分的變數—信號演算法不具足夠的區辨力，以至於考量那些演算法的值時會過度配適，整體分類錯誤率增加。

由圖所示，變數篩選前 40 個最具區辨力的演算法，再做 ISOMAP 降維，並取前 6 個維度，此時預測發聲是否改善的錯誤率最低，僅

9%，即有 91%的預測準確率。

3.3.5.1 及此節分別利用 10-fold SVM 重複 100 次以及 Leave-one-out SVM 兩者的最佳變數篩選個數均為前 40 個最具區辨力的演算法，並取 ISOMAP 維度縮減後前 6 個維度做預測，預測結果最佳，然而後者的預測準確率較前者高出約 1%，且後者每筆資料都曾作為模型的訓練集與測試集，故選擇 Leave-one-out SVM。

3.3.5.3 治療後有(無)改善的聲音特徵

前一小節提到了經篩選出前 40 個最具區辨力的演算法，再透過 ISOMAP 前 6 個維度的值，能使得錯判率最低。除了找到使得錯誤率最小化的分類預測方法，另外，甚麼樣特性的聲音會被預測為治療有或沒有改善也十分重要。

在此節掌握了有無改善的聲音的主要特徵後，下一節將提到可以在預測某些型態的聲音治療後沒有改善後，藉由維度縮減最具區辨力的演算法的值，更進一步判斷聲音不合格的因素；提供給 LSVT 療程專家作為未來治療調整的方向，此節利用 Leave-one-out SVM 的預測分類機率先找出症狀有無改善的聲音的主要特徵。

一、SVM 的預測分類機率與 ISOMAP 維度投影值的關聯

分類機率是預測治療結果的依據，由其與 ISOMAP 6 個維度投影值的關聯性找出每個維度投影值大小、方向對預測結果的影響。

```
> # 5. SVM, CV(Leave one out), isomap dim=6
> # a). Find the most proper k(# of neighbors)
> library(dplyr)
> cv.svm accur.given.k <- apply(as.matrix(2:20), 1,
function(x) {
```

```

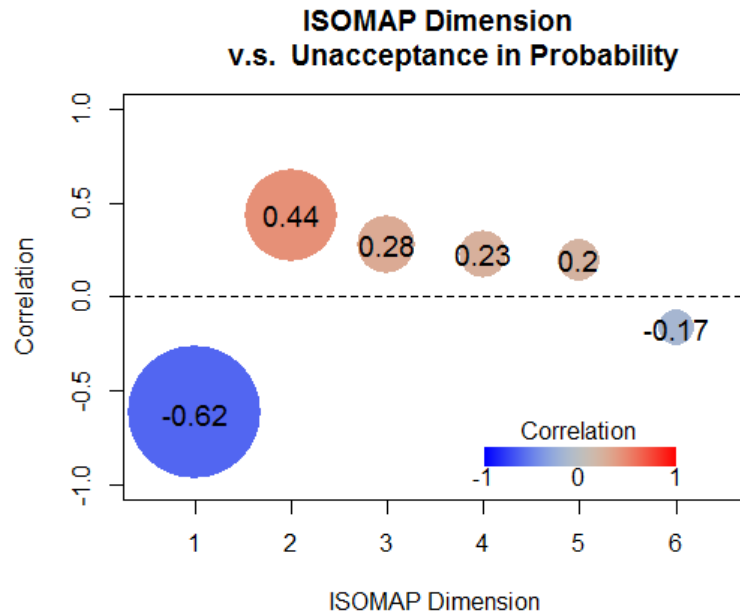
+   iso.pts <- isomap(dist(scale(data.sig.var)),
+                     ndim = 6, k = x)$points
+   accuracy <- apply(as.matrix(1:dim(data.sig.var)[1]), 1,
+                     cv.svm accur.fun,
+                     data.frame(iso.pts), 6) %>% mean
+   accuracy
+ })
> cv.svm.best.k <- (2:20)[which.max(cv.svm accur.given.k)]
>
> # b). Isomap
> cv.svm.lsvt.isomap <- isomap(dist(scale(data.sig.var)),
+                               ndim = 6, k = cv.svm.best.k)
> # c). classification by SVM
> library(e1071)
> cv.svm.prob.fun <- function(test.id) {
+   data <- data.frame(cv.svm.lsvt.isomap$points)
+   x.train <- data[-test.id, ]
+   x.test <- data[test.id, ]
+   y.train <- class.f[-test.id]
+   y.test <- class.f[test.id]
+   train.set <- data.frame(x.train, y.train)
+
+   svm.model <- svm(y.train ~ ., data = train.set,
+                    probability = T)
+
+   pred <- predict(svm.model, x.test, probability = T)
+   c(attr(pred, "probabilities"), pred)
+ }
> cv.svm.prob.fun.out <-
t(apply(as.matrix(1:dim(data.sig.var)[1]),
+       1, cv.svm.prob.fun))
> cv.svm.prob.class <- apply(cv.svm.prob.fun.out, 1,
function(x) {
+   if ((x[3] == 1) && (x[1] < x[2]) ||
+       (x[3] == 2) && (x[1] > x[2])) {
+     replace <- x[1]
+     x[1] <- x[2]
+     x[2] <- replace
+   }
+   x

```

```

+ }) %>% t
> colnames(cv.svm.prob.class) <- c("Acceptable",
+                                   "Unacceptable",
+                                   "class")
> # 6. Prob.of unaccept. [corr.to] iso.dim.
> cor.dim.unaccept.prob <-
+   apply(cv.svm.lsvt.isomap$points, 2, function(dim)
+     cor(dim, cv.svm.prob.class[ , 2]))
>
> plot(1:6, cor.dim.unaccept.prob,
+      cex = 20 * (abs(cor.dim.unaccept.prob)),
+      xlim = c(0.5, 6.5), ylim = c(-1, 1),
+      xlab = "ISOMAP Dimension", ylab = "Correlation",
+      main = " ISOMAP Dimension
+      v.s. Unacceptance in Probability",
+      col = col.red.blue[trans(cor.dim.unaccept.prob)], pch
= 16)
> abline(h = 0, lty = 2)
> text(1:6, cor.dim.unaccept.prob,
+      round(cor.dim.unaccept.prob, 2), cex = 1.2)
> rasterImage(t(col.red.blue), 4, -0.9, 6, -0.8)
> text(c(4, 5, 5, 6), c(-0.95, -0.7, -0.95, -0.95),
+      c(-1, "Correlation", 0, 1))
>

```



圖中數值表示沒有改善的聲音的機率與 ISOMAP 各維度投影值的相關係數。治療後沒有改善的機率與聲音在 ISOMAP 第 1、6 維度的投影值為反向關係；與 2 至 5 維度的投影值為正向關係。表示若某聲音演算後的值經 ISOMAP 縮減維度後，第 1、6 維度投影值越小或是在第 2 至 5 維度的投影值越大，該聲音治療後沒有改善的可能性越大。前兩維與 SVM 預測分類機率存在中度相關，第三維度接近中度相關，後三個維度則為低度相關，故知後三個維度的雖亦能提供分類的訊息，但主要的訊息是來自前三個維度，換言之，前三個維度影響預測結果極大。接下來便要找出此三個維度的代表意義。

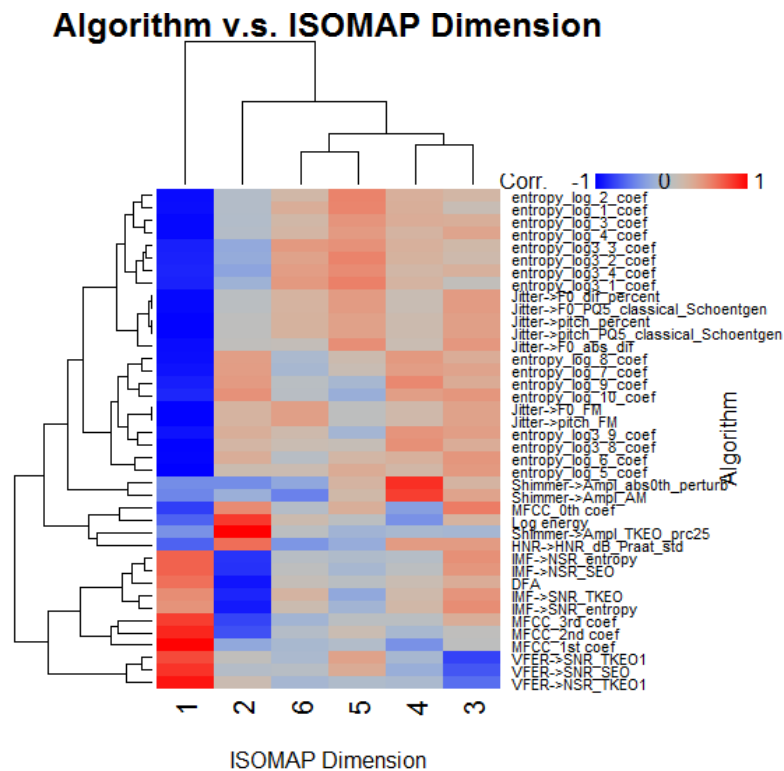
二、維度投影值與演算法的關聯及維度的定義

```
> # 7. isomap$points [corr.to] algorithm
> #     Sig. algorithm
> cv.svm.iso.dim <- cv.svm.lsvt.isomap$points
>
> library(fields)
> col.red.blue <- two.colors(start = "blue",
+                             middle = "gray",
```

```

+                                     end = "red")
> trans <- function(x) {round((x+1)*127.5)+1}
>
> cor.dim.var <- apply(data.sig.var, 2, function(x)
+   cor(x, cv.svm.iso.dim))
> # heatmap
> heatmap(trans(t(cor.dim.var)),
+   col = col.red.blue,
+   Rowv = NULL,
+   margins = c(4, 11),
+   xlab = "ISOMAP Dimension", ylab = "Algorithm",
+   main = "Algorithm v.s. ISOMAP Dimension")
> # legend
> rasterImage(t(col.red.blue), 5.1, 1, 6.4, 1.1)
> text(c(4.98, 4.5, 5.7, 6.5), rep(1.05, 4),
+   c(-1, "Corr.", 0, 1))
>
> # High Correlation is considered
> cv.svm.positive.var <- apply(cor.dim.var, 1, function(x)
+   names(which(x >= 0.5)))
> cv.svm.negative.var <- apply(cor.dim.var, 1, function(x)
+   names(which(x <= -0.5)))
> # view~. => def. dim.
>
> iso.dim.def <- c("能量、音調持續穩定性",
+   "噪音(聲帶)、低音量波動性",
+   "噪音(聲帶、咬合)")
>

```



上述分類訊息主要來自於前三個維度，由各個維度與演算法關聯性的熱圖看出前三個維度則受某些演算法影響特別大。

以下表格記錄主宰前三個維度的演算法與其意義、歸納維度的意義，與症狀未改善的關係：

維度	Dim. 1		Dim. 2		Dim. 3
方向	+	-	+	-	-
演算法	MFCC_1 st , 2 nd , 3 rd VFER	entropy_1 og_1, 3 Jitter- >F0/pitch	Shimmer- >Ampl prc25	IMF DFA	VFER
說明	信號能量的穩定性 信號噪音	小波波段 1 & 3 波 動,	低音量振 幅偏離(不 穩定)	信號噪音 比	信號噪音比

	比	聲帶振動 頻率偏離 (不穩定)			
維度 意義	波動穩定性 能量穩定性 週期性(音調)		噪音含量 音量(小)不穩定		噪音含量(聲帶閉合不全、咬合)
維度 定義	能量、音調持續穩定性		噪音(聲帶閉合不全) 音量(小)不穩定		噪音(聲帶閉合、咬合問題)
V. S. 症狀未 改善	↓		↑		↑

經由各個維度的主宰演算法變數，ISOMAP 第一個維度衡量聲音能量、音調的持續穩定性；第二個維度衡量來自於聲帶閉合不全帶來的噪音以及音量(小)波動性；第三個維度則衡量來自於聲帶閉合、咬合問題交互所帶來的噪音。

一、治療有(無)改善的聲音特徵

```

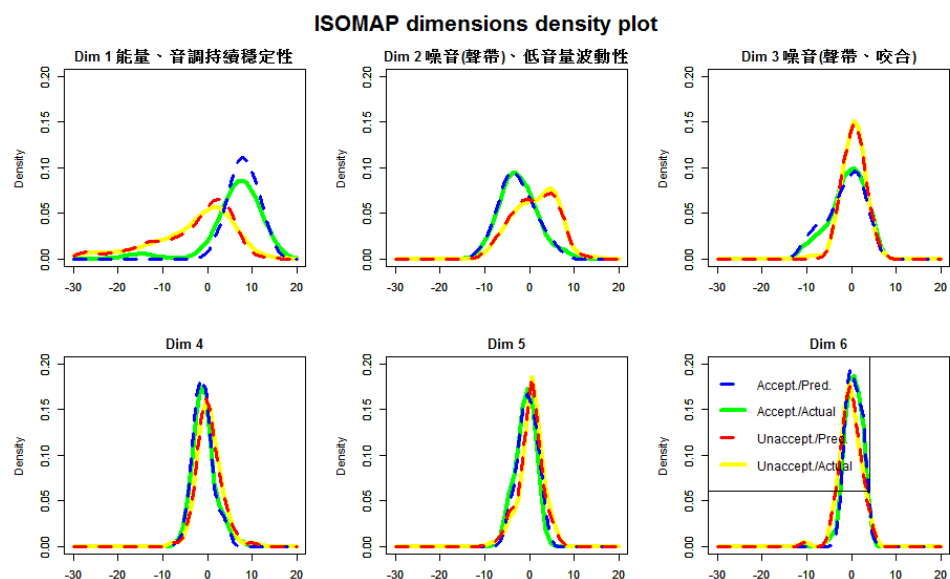
> # 8. plot
> # Density plot (iso.pts' distr. given pred.class)
> par(mfrow = c(2, 3))
> par(mai = c(0.3, 0.6, 0.6, 0.2))
> apply(as.matrix(1:6), 1, function(x) {
+   dim <- cv.svm.lsvt.isomap$points[ , x]
+   # Actual
+   sm.density.compare(dim, class.f,
+                       col = c("green", "yellow")[class],
+                       lty = rep(1, 2), lwd = 4,
+                       xlim = c(-30, 20),
+                       ylim = c(0, 0.2), xaxt = "n",

```

```

+             yaxt = "n", bty = "n",
+             xlab = "", ylab = "")
+   par(new = T)
+   # Pred.
+   sm.density.compare(dim, cv.svm.prob.class[ , 3],
+                       col = c(4, 2)[cv.svm.prob.class[ , 3]],
+                       lty = rep(2, 2), lwd = 3,
+                       xlim = c(-30, 20),
+                       ylim = c(0, 0.2),
+                       xlab = "")
+   titles <- ifelse(is.na(iso.dim.def[x]) == 1,
+                    paste(paste("\n\nDim", x)),
+                    paste("\n\nDim", x, iso.dim.def[x]))
+   title(titles)
+ })
NULL
>
> legend("topleft", c("Accept./Pred.", "Accept./Actual",
+                     "Unaccept./Pred.", "Unaccept./Actual"),
+       col = c("blue", "green", "red", "yellow"),
+       lty = c(2, 1, 2, 1), lwd = c(3, 4, 3, 4),
+       bg = cm.colors(1, alpha = 0), cex = 1.1)
> title("\nISOMAP dimensions density plot",
+       outer = T, cex.main = 2)
>

```

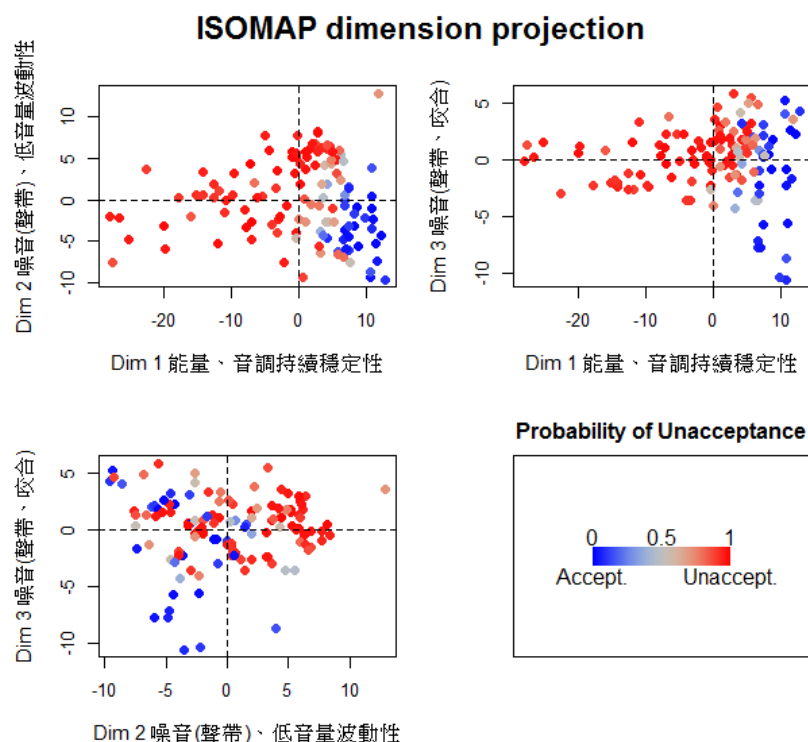


上圖為實際及預測聲音有無改善的 ISOMAP 各維度投影值的分布。藍色虛線為預測治療後有改善的聲音；綠實線為實際治療後有改善的聲音；紅虛線為預測治療後沒有改善的聲音；黃實線則為實際治療後無改善的聲音，可以看見 Leave-one-out SVM 預測的結果與實際治療結果相近。

有無改善的聲音在前兩個維度的差異最大，第三個維度次之，其中，治療後沒有改善的聲音能量、音調持續穩定性較小，由聲帶或咬合問題產生的噪音越大，低音量時的波動越大。

```
> # 2D prob.
> par(mfrow = c(2, 2))
> par(mai = c(0.7, 0.7, 0.6, 0.2))
> apply(matrix(c(1, 1, 2, 2, 3, 3), ncol = 2), 1,
+       function(x) {
+         plot(cv.svm.iso.dim[ , x[1]],
+              cv.svm.iso.dim[ , x[2]],
+              col = col.red.blue[
+                trans(cv.svm.prob.class[ , 2]*2-1)],
+              pch = 16, cex = 1.2,
+              xlab = paste("Dim", x[1], iso.dim.def[x[1]]),
+              ylab = paste("Dim", x[2], iso.dim.def[x[2]]),
+              cex.lab = 1.1)
+         abline(h = 0, v = 0, lty = 2)
+       }
+ )
NULL
> plot(0, xlab = "", ylab = "", type = "n", xaxt = "n",
+      yaxt = "n", main = "\nProbability of Unacceptance",
+      xlim = c(-1, 1), ylim = c(-1, 1))
> rasterImage(t(col.red.blue), -0.5, -0.1, 0.5, 0.1)
> text(c(-0.5, 0, 0.5, -0.5, 0.5),
+      c(0.22, 0.22, 0.22, -0.22, -0.22), cex = 1.2,
+      c(0, 0.5, 1, "Accept.", "Unaccept."))
> title("\nISOMAP dimension projection",
```

```
+      outer = T, cex.main = 1.7)
>
```



樣本 126 個聲音在前三個 ISOMAP 維度兩兩的投影值如上圖，圖中顏色表示 SVM 預測無改善的機率大小，聲音被預測為沒有改善的機率越大，點的颜色越偏紅色，反之偏藍色。

同樣地，治療後沒有改善的聲音能量、音調持續穩定性明顯較小，能量、音調持續地穩定是判斷患者聲音的治療狀況有無改善最好的準則，當治療後聲音的能量、音調持續穩定性不夠，被預測為無改善的機率很高；然而當能量、音調很穩定、噪音比例卻很高時，被預測無改善的機率也很高。

治療後有改善的病患，除了發聲時的能量、音調較能持續保持穩定以外，因 PD 使聲帶閉合不全而產生的噪音問題透過治療而得到改善，但有包含咬合問題，而導致的噪音(維度三)則難以判斷治療結果，人人都有可能咬合不整齊，而導致發音時空氣流動產生不必要的

聲音。

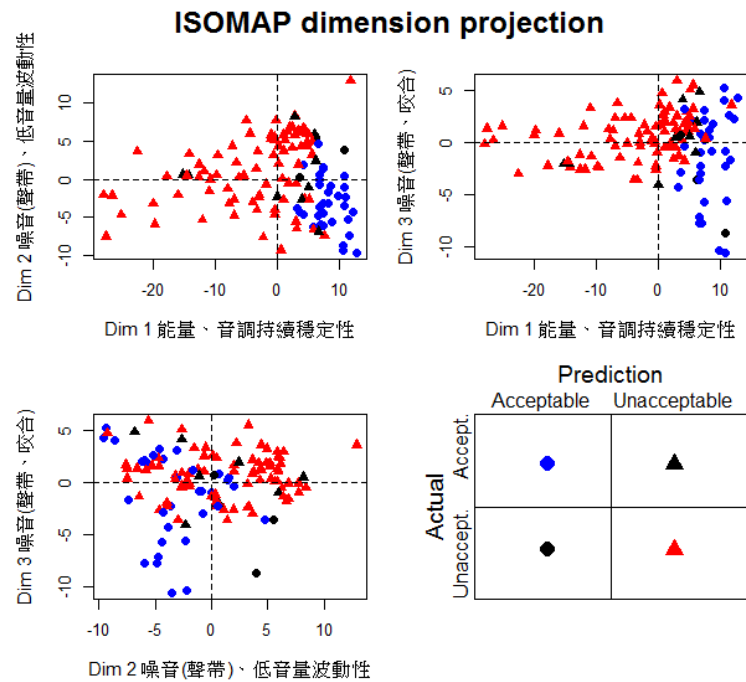
四、 易被預測錯誤的聲音特徵

```
> # 2D, pred./actual class
> pred.class <- cv.svm.prob.class[ , 3]
> cv.svm.actual.pred.class <-
+   ifelse((class == 1 & pred.class == 2), 3,
+         ifelse((class == 2 & pred.class == 1), 4, class))
>
> par(mfrow = c(2, 2))
> par(mai = c(0.7, 0.7, 0.6, 0.2))
> apply(matrix(c(1, 1, 2, 2, 3, 3), ncol = 2), 1,
+       function(x) {
+         plot(cv.svm.iso.dim[ , x[1]],
+              cv.svm.iso.dim[ , x[2]],
+              col = c(4, 2, 1, 1)[cv.svm.actual.pred.class],
+              pch = c(16, 17, 17,
16)[cv.svm.actual.pred.class],
+              cex = 1.2, cex.lab = 1.1,
+              xlab = paste("Dim", x[1], iso.dim.def[x[1]]),
+              ylab = paste("Dim", x[2], iso.dim.def[x[2]]))
+         abline(h = 0, v = 0, lty = 2)
+       })
+ )
NULL
> plot(c(-0.5, 0.5, -0.5, 0.5),
+      c(0.5, 0.5, -0.5, -0.5),
+      pch = c(16, 17, 16, 17),
+      col = c(4, 1, 1, 2),
+      xlab = "", ylab = "", cex = 2,
+      xaxt = "n", yaxt = "n", main = "",
+      xlim = c(-1, 1), ylim = c(-1, 1))
> abline(v = 0, h = 0)
> mtext(c("Actual", "Prediction"),
+       side = 2:3, padj = c(-1.5, -1.5),
+       adj = c(0.5, 0.5), cex = 1.2)
> mtext(c("Accept.", "Unaccept.",
+         "Acceptable", "Unacceptable"),
```

```

+   side = c(2, 2, 3, 3),
+   padj = rep(-0.2, 4),
+   adj = c(1, 0, 0.1, 0.9))
> title("\nISOMAP dimension projection",
+       outer = T, cex.main = 1.8)
>

```



藍色及紅色的點為預測正確的聲音，黑色為預測錯誤的聲音。雖然能量、音調持續地穩定是判斷患者聲音的治療狀況有無改善最好的準則，但仍少數會被預測錯誤。被預測錯誤的聲音中，幾乎都是屬於原本為治療後有改善但被錯判成沒有改善的聲音；實際上沒有改善的聲音較不會被錯判。會被錯判的聲音大致上有兩種，一是能量、音調持續穩定性不高不低，二是能量、音調持續穩定性高但噪音比例太高。

五、 症狀嚴重程度

```

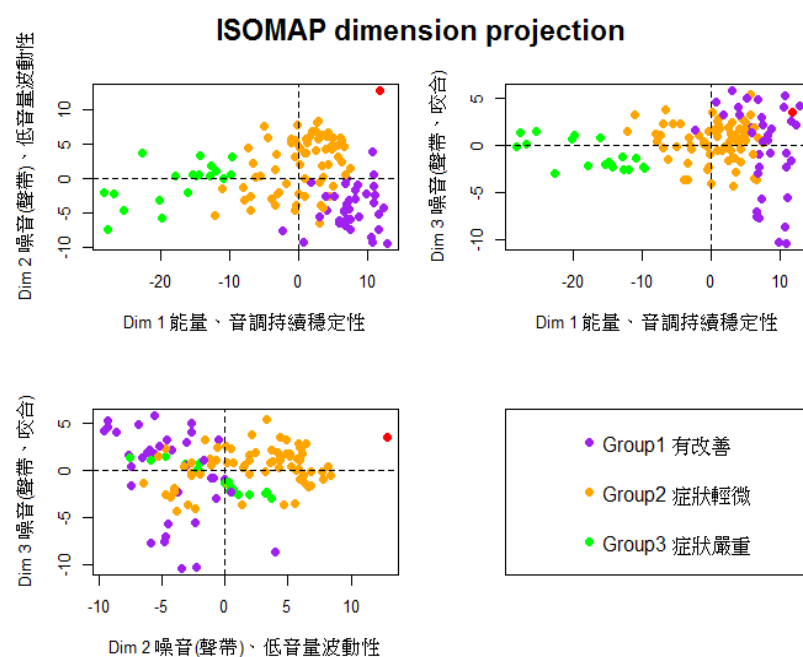
> # 9. isomap$points clustered

```

```

> # Clustered isomap pts plot, k = 4
> hclust.iso.pts <- hclust(dist(cv.svm.iso.dim),
+                           members = NULL)
> cut.h.cluster <- cutree(hclust.iso.pts, k = 4)
>
> par(mfrow = c(2, 2))
> par(mai = c(0.7, 0.7, 0.6, 0.2))
> apply(matrix(c(1, 1, 2, 2, 3, 3), ncol = 2), 1,
+        function(x) {
+          plot(cv.svm.iso.dim[ , x[1]],
+               cv.svm.iso.dim[ , x[2]],
+               col = c("purple", "orange",
+                       "green", "red")[cut.h.cluster],
+               pch = 16, cex = 1.2,
+               xlab = paste("Dim", x[1], iso.dim.def[x[1]]),
+               ylab = paste("Dim", x[2], iso.dim.def[x[2]]),
+               cex.lab = 1)
+          abline(h = 0, v = 0, lty = 2)
+        }
+ )
NULL
> plot(0, xlab = "", ylab = "", type = "n",
+      xaxt = "n", yaxt = "n", main = "",
+      xlim = c(-1, 1), ylim = c(-1, 1))
> legend("center", c("Group1 有改善",
+                     "Group2 症狀輕微",
+                     "Group3 症狀嚴重"),
+       pch = 16, bty = "n", cex = 1.2, pt.cex = 1.2,
+       col = c("purple", "orange", "green"))
> title("\nISOMAP dimension projection",
+       outer = T, cex.main = 1.7)
>

```



上圖為以 ISOMAP 所取的前六個維度作群集分析後分群的結果，群一為有改善的聲音，群二為尚未改善但症狀較輕微的聲音，群三則屬於症狀較為嚴重，聲音最不穩定、最無法控制的一群。故除了利用 ISOMAP 投影值去做分類，預測治療後症狀是否有改善，亦可透過投影值的大小分辨病患的聲音症狀屬於輕微或嚴重，可針對症狀的輕重找到最適合的治療方式。