# 國 立 交 通 大 學

## 資訊工程學系

## 碩 士 論 文

以叢集為基礎的支撐向量機學習及其應用於語者辨識

The Cluster-based Learning of Support Vector Machines and Its Application in Text-Independent Speaker Identification

研 究 生：孫聖育

指導教授：傅心家　教授

中 華 民 國 九 十 三 年 七 月

# 以叢集為基礎的支撐向量機學習及其應用於語者辨識

# The Cluster-based Learning of Support Vector Machines and Its Application in Text-Independent Speaker Identification


研 究 生：孫聖育　　Student：Sheng-Yu Sun

指導教授：傅心家 教授　　Advisor：Prof. Hsin-Chia Fu

國 立 交 通 大 學

資 訊 工 程 學 系


碩 士 論 文

A Thesis

Submitted to Department of Computer and Information Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science and Information Engineering

July 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年七月

# 以叢集為基礎的支撐向量機學習及其應用於語者辨識

學生：孫聖育　　指導教授：傅心家 教授

國立交通大學資訊工程學系

## 摘　要

擁有充分的統計學習理論基礎的支撐向量學習機(Support Vector Machine)在分類與辨識的問題上有相當好的表現水準，例如：圖型辨識(Pattern Recognition)和語者辨識(Speaker Identification)等。然而， 訓練 SVM 時需要大量的記憶體來計算且花費很多時間。針對大資料量這類的例子，我們提出了一個以叢集為基礎(Cluster-based)的 SVM 利用叢集的概念將待訓練的資料做初步的篩選，挑選出位於每個叢集外圍上面的資料，也就是對於 SVM 的切割平面(separating hyperplane)有較大影響的資料，以達到加速訓練的效果，進而減少支撐向量的個數而提升辨識的效率。我們將其應用於語者辨識的問題上，在辨識率幾乎不受影響的情況下，訓練資料減少了約 75%且訓練時間減少了約 85%。此外，所得到的支撐向量(support vector)總數也減少了 4 倍左右的量，使得辨識的效率大幅的提升。此外，我們將 SVM 分類功能實際應用於電視新聞內容上的氣象播報偵測，也達到了很好的結果。

# The Cluster-based Learning of Support Vector Machines and Its Application in Text-Independent Speaker Identification

Student : Sheng-Yu Sun Advisor : Prof. Hsin-Chia Fu

Institute of Computer Science and Information Engineering
National Chiao Tung University

## Abstract

Based on Statistical learning theory, Support Vector Machine(SVM) is a powerful tool for various classification problems, such as pattern recognition and speaker identification etc. However, training SVM consumes large memory and long computing time. This paper proposes a cluster-based learning methodology to reduce training time and the memory size for SVM. By using k-means based clustering technique, training data at boundary of each cluster were selected for SVM learning. We applied this technique to text-independent speaker identification problems. Without deteriorating recognition performance, the training data and time can be reduced up to 75% and 85% respectively. Furthermore, The amount of support vectors of SVM models are the quarter of full-SVM such that the recognition action is more effective. Finally, we apply our proposed method to the case of detecting the weather-forecasting segments in a news program.

# 誌　謝

首先我要感謝老師在課業研究及生活上的指導與關心，讓我在碩士班這兩年來受益良多。感謝賴柏伸、曾政龍、陳岳宏、徐永煜等學長對於我的問題給予的解答與幫忙，尤其是政龍跟柏伸，你們不厭其煩的幫我解惑及解決困難，感謝信憲、博傑學長和留圓學姐給予的幫忙，還有俊銘跟子源的相互打氣與加油，讓我可以較快的適應不同的環境，揚智、宗儒和宜玲學弟妹適時的加入以及給予的協助，增添了生活上的樂趣。此外，我還要感謝父母親以及姊姊跟弟弟，感謝爸媽讓我衣食無缺可以專心於學業跟研究上，還有姊姊跟弟弟的幫忙分擔家計跟照顧父母。還有謝謝大學同學們，聽我訴苦，一同吃飯聊天紓解壓力；還有王小川老師和王逸如老師百忙之中願意抽空蒞臨指導論文口試。再者，感謝老天爺給我的福氣，讓我可以順順利利、平平安安。

# 目　　錄

頁次

# 表　　目　　錄

# 圖　　目　　錄

# 第一章 簡介

## 1.1 動機 ( Motivation )

對於語者辨識來說，使用 GMM 的方法已經被驗證可以達到相當好的準確性[1]；然而，GMM 的 models 建立過程受所收集到的訓練資料的分布與完整性影很很大，如果收集到的訓練資料的分布無法涵蓋到所有資料的分布情況，則 GMM 的辨識能力將大打折扣。支撐向量機( Support Vector Machine )是一種區別性的辨別器( discriminative classifier )，而 GMM 是一種機率性的辨別器( generative probability classifier )[2]。因為機率有著不確定的因素存在，無法較明確的掌控其辨識率。而 SVM 有個明確的錯誤上界( error upper bound )的理論值以及較完善的數學理論基礎，可調整一些因素(如：支撐向量( support vectors )的個數或其 VC(Vapnik-Chervonenkis)維度等)，使得錯誤上界變小，進而能夠預期實際上的測試錯誤率變小[3][4]，且 SVM 的最終目的是尋找資料的最佳切割平面，不是去 model 訓練資料的分布，所以較不受訓練資料分布的全面性與否而影響其結果，因此支撐向量機是個最近相當被重視的一個方法。根據 Cover 的 separability 定理[1][5]不同類別的資料如果其原本是無法被線性分割的，可以經由將其轉換到不同的空間或提高到更高維度的空間，使其可以被線性分割。而支撐向量機的主要

[1] A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space.

精神就是試圖將資料轉換到另一個空間中，找到適當的分割平面( separating hyperplane )來將不同類別的資料作合適的區分。雖然支撐向量機有著讓人喜歡的特性及穩固的理論基礎[3]，但是支撐向量機在求解的過程中，需要大量的記憶體，隨著維度越高，資料量越大，其所需要的記憶體也越大，因而需要的計算複雜度也越高，一旦所需記憶體超過系統可提供的記憶體總量時，SVM 的訓練時間會因為訓練過程中常常需要存取到硬碟而導致時間變得非常的漫長。為了克服這個問題，如何節省訓練資料量及加速支撐向量機是一個值得研究的課題。

本論文提出了以叢集為基礎( cluster-based )的資料篩選法，使用 K-means clustering 來達到有效的資料篩選以達到縮減訓練資料的目的，進而加速支撐向量機的訓練速度。

## 1.2 章節組織 ( Thesis Organization )

本篇論文接下來的組織如下，在第二章，我們會先對支撐向量機( SVM )作一些基本的介紹，然後將一些加速 SVM 的相關研究簡介一下；第三章介紹一下叢集的方法以及我們所提出的架構；第四章描述實驗與結果分析；第五章是將其應用於新聞氣象報導的偵測與辨識；最後第六章會對本論文作一個結論以及未來工作的探討。

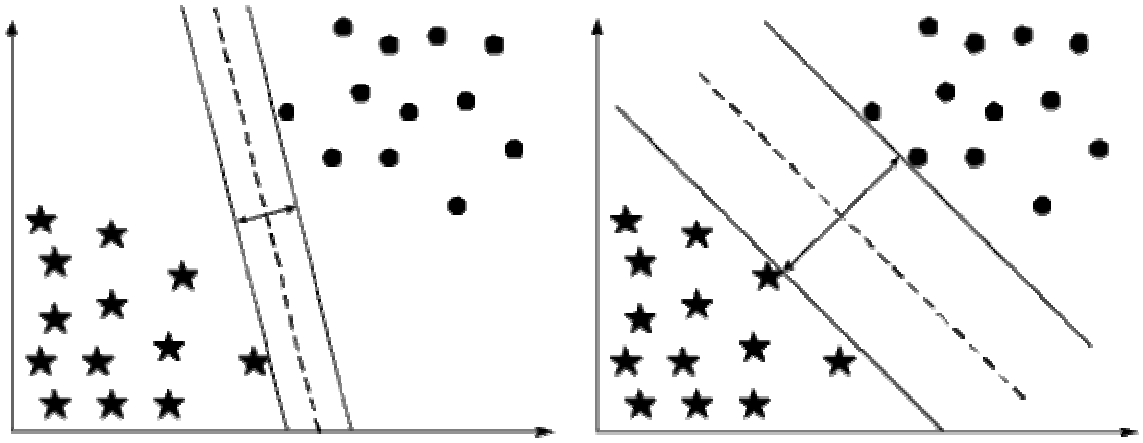
# 第二章 相關研究

為了加快 SVM 的訓練速度以及節省訓練 SVM 時所需的記憶體空間，有很多方法及策略被提出，其中主要的訴求主題分為兩類，一類為針對 QP 問題( Quadratic Programming problem )上的加速方法，另一類為對訓練資料的篩選或分群的加速法。

## 2.1 概述支撐向量機

本論文只對支撐向量機( Support Vector Machine, SVM )作一個概略性的敘述，有興趣的讀者可以參考 [3][4]。SVM 是一個尋找兩個不同類別資料間的最佳切割平面( Optimal Separating Hyperplane, OSH )或者是估測資料間的最佳迴歸式( Regression )。本論文採用的是 SVM 在尋找最佳切割平面上的功能。SVM 之所以會被重視且看好的原因，是因為 SVM 在求解時相當於在解一個 quadratic programming problem( QP problem )，而對於 QP 問題來說，已經有相當厚實的數學理論基礎在支撐著它，只要其滿足一些特性就能夠確保所找到的解是全域最佳解( global solution )而不是局部解( local solution ) [12]。基本上，SVM 可以分成可分割( separable )與不可分割( non-separable )兩種類型，其又可以細分成線性可分( linearly separable )，線性不可分( linearly non-separable )，非線性可分( non-linearly separable )和非線性不可分( non-linearly non-separable )四種狀

況。



(a) (b)

圖 2-1 (a) 兩類資料的切割平面但 margin 較小。(b) 較大 margin 的切割平面，也是這兩類資料的最佳切割平面，因其具備最大的 margin。

### 2.1.1 線性的支撐向量機( Linear Support Vector Machine )

以最簡單的線性可分的案例來說，給定一組資料集合$\{\mathbf{x}_i, y_i\}, i = 1,\ldots, n, \mathbf{x}_i \in \mathbf{R}^d, y_i \in \{-1, +1\}$, $\mathbf{x}_i$ 代表第 $i$ 筆資料的特徵向量, $y_i$ 表示其所屬的類別。我們希望能夠找到一個能夠分開 "+1" 與 "-1" 這兩類資料的平面( hyperplane )。考慮這類平面函數的家族：

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = \mathbf{w}^T \mathbf{x} + b \tag{2.1}$$

SVM 就是要從中找到合適的 $\mathbf{w}^*$ 和 $b^*$ 使得 $sgn(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) = sgn(y_i)$，而 $f(\mathbf{x}) = \mathbf{w}^* \cdot \mathbf{x} + b^*$ 就是 SVM 的最佳切割平面。所謂的最佳切割平面就是能夠完全的區分資料且擁有最大 margin 的平面，而 margin 就是各類別中最靠近切割平面的資料到切割平面的最短距離, 如圖 2-1。為了求得最大 margin 的切割平

面，我們必須解決以下的含有限制條件的最佳化問題( optimization problem with constraints )：

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2$$
$$subject \quad to$$
$$y_i(\mathbf{w}\cdot\mathbf{x}+b)\geq 1,\ i=1,\cdots,n \tag{2.2}$$

我們稱(2.2)式為最佳化問題的 primal form，而因為大部分時候 primal form 較難解，通常都被轉換到 dual form 的型式，試圖能夠較簡單的來解此問題。藉由引進 Lagrange multipliers 來將 primal form 轉到 dual form，轉換方式如下：

$$L(\mathbf{w},b,\mathbf{\Lambda})=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{n}\lambda_i[y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1] \tag{2.3}$$

其中 $\mathbf{\Lambda}=(\lambda_1,\cdots,\lambda_n)$, $\forall\ \lambda_i\geq 0$ 是 Lagrange multipliers。該式分別對 **w** 和 *b* 微分並令其微分值為零，得到下列兩式

$$\frac{\partial L(\mathbf{w},b,\mathbf{\Lambda})}{\partial \mathbf{w}}=\mathbf{w}-\sum_{i=1}^{n}\lambda_i y_i\mathbf{x}_i=0 \tag{2.4}$$

$$\frac{\partial L(\mathbf{w},b,\mathbf{\Lambda})}{\partial b}=\sum_{i=1}^{n}\lambda_i y_i=0 \tag{2.5}$$

將兩式代入(2.3)式，我們可以獲得

$$F(\mathbf{\Lambda})=\sum_{i=1}^{n}\lambda_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j\mathbf{x}_i\cdot\mathbf{x}_j \tag{2.6}$$

如此原本的最小化 primal QP 問題變成了最大化的 dual QP 問題，如下式：

$$\max_{\Lambda} F(\mathbf{\Lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

$$\textit{subject to} \tag{2.7}$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

$$\lambda_i \geq 0,\ i = 1, \cdots, n$$

由 (2.7) 式解出 $\mathbf{\Lambda}^* = (\lambda_1^*, \cdots, \lambda_n^*)$ 之後，根據 (2.5) 式可以得到 $w^* = \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{x}_i$ ，而 $b^* = -\frac{\max_{y_i=-1}(w^* \cdot \mathbf{x}_i) + \min_{y_i=+1}(w^* \cdot \mathbf{x}_i)}{2}$ ，然後得到最佳切割平面 $f(\mathbf{x}) = \mathbf{w}^* \cdot \mathbf{x} + b^* = \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{x}_i \cdot \mathbf{x} + b^*$ 。根據 KKT complementarity conditions，最佳解 $\mathbf{\Lambda}^*$， $\mathbf{w}^*$， $b^*$ 必需滿足

$$\lambda_i^* [y_i (\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1] = 0,\ \ i = 1, \cdots, n \tag{2.8}$$

因此，只有當 $y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) = 1$ 時，其相對應的 $\lambda_i^*$ 才有可能大於零，我們定義 $\lambda_i^* > 0$ 所相對應的資料 $\mathbf{x}_i$ 為支撐向量( Support Vector )，而最佳切割平面可以改寫成

$$f(\mathbf{x}) = \sum_{i \in SV} \lambda_i^* y_i \mathbf{x}_i \cdot \mathbf{x} + b^* \tag{2.9}$$

從上式可以看出，構成 SVM 的最佳切割平面並不需要由所有的資料，我們只需要知道支撐向量和其相對應的 Lagrange multipliers 就可以知道我們的最佳切割平面。換句話說，如果沒有了支撐向量，那麼這個平面也就會隨之瓦解。

但是並非所有的資料都可以被線性分割，所以上述的型式就沒辦法適用不可分割的情況。針對這樣的問題，如果我們還是想要尋找一個適當的線性切割平面，則勢必要容忍一些誤差的狀況發生，因此 SVM 引進了一個新的變

數 ***slack variable***，$\xi_i$。則尋找最大 margin 變成了尋找最大的 margin 但是必需有最小的錯誤，其最佳化問題的 primal form 如下：

$$\min_{\mathrm{w}} \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} c_i \xi_i^p$$

$$\textit{subject to}$$

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i,\ i = 1, \cdots, n$$

$$\xi_i \geq 0, i = 1, \cdots, n \tag{2.10}$$

其中 $c_i > 0,\ i = 1, \cdots, n$ 是錯誤的懲罰參數( penalty parameters )，用來控制每個錯誤資料的傷害程度。通常為了簡單起見，我們都設定 $\mathrm{p} = 1,\ c_i = C\ \forall\ i$。

當 $\mathrm{p} = 1,\ c_i = C\ \forall\ i$，其 Lagrange 方程式如下：

$$L(\mathbf{w}, b, \Xi, \mathbf{\Lambda}, \Gamma) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\lambda_i[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{n}\gamma_i\xi_i \tag{2.11}$$

其中 $\mathbf{\Lambda} = (\lambda_1, \cdots, \lambda_n)$, $\mathbf{\Gamma} = (\gamma_1, \cdots, \gamma_n),\ \forall\ \lambda_i \geq 0,\ \gamma_i \geq 0$ 為 Lagrange multipliers。分別對 $\mathbf{w}, b,\ \Xi$ 微分且令其結果為零，然後再將得到的式子代入 primal form 中，可以得到其 dual form，如下：

$$\max_{\Lambda} F(\mathbf{\Lambda}) = \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

$$\textit{subject to}$$

$$\sum_{i=1}^{n}\lambda_i y_i = 0$$

$$0 \leq \lambda_i \leq C,\ i = 1, \cdots, n \tag{2.12}$$

與線性可分的最佳化問題差異處只在限制條件中 Lagrange multipliers, $\lambda_i$, 被 $C$ 所限制住。

### 2.1.2 非線性的支撐向量機 ( Nonlinear Support Vector Machine )

在實際的情況下，使用線性的切割平面(或稱辨別器( classifier ))只能解決一小部份的問題，大部分的情況都是非線性分割的問題。SVM 如何來解決此類問題呢? 對於原本無法被線性分割的資料，我們可以提升其維度使其在高維度的空間中可以被線性分割，於是 SVM 使用了一個非線性的轉換，將原本的資料非線性的轉換到另一個維度的空間上，然後在該空間上找到其最佳的切割平面。我們將資料原本所在的空間稱為輸入空間( input space )而稱轉換到的空間為特徵空間( feature space )。令轉換函數如下：

$$\phi : \mathbf{R}^d \rightarrow F$$

我們將資料經由ϕ轉換到特徵空間後，切割平面方程式變成

$$f(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b \tag{2.13}$$

非線性 SVM 的最佳化問題的 dual form 變成了

$$\max_{\Lambda} F(\mathbf{\Lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

$$\textit{subject to}$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

$$0 \le \lambda_i \le C,\ i = 1, \cdots, n \tag{2.14}$$

很明顯的與(2.12)式的差別只在於內積的對象改變了；原本在輸入空間中的內積行為轉換到特徵空間中的內積。

為了節省從輸入空間轉換到特徵空間的運算量，SVM 定義了 kernel function, $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j)$。為了確保(2.14)式有解，且找到的解是全域最佳解( global optimal solution )，函數 $F(\Lambda)$ 必須是 convex function [12]。因為 $F(\Lambda)$ 是 quadratic function，所以 kernel 矩陣 K(,) 必須半正定( positive semi-definite )或正定( positive definite )，因此 kernel function 必須滿足 Mercer's condition [3][12]，也就是

$$\sum_{i,j=1}^{n} K(\mathbf{x}_i, \mathbf{x}_j)\rho_i\rho_j \geq 0, \forall\ \mathbf{x}_i, \mathbf{x}_j\ and\ \forall\ \rho_i, \rho_j \in \mathbf{R} \tag{2.15}$$

因為 kernel function 可以表現出資料在特徵空間中的內積行為，不僅可以節省轉換所需要的運算量，也可以不需要真的造出轉換函數。雖然 kernel function 有著這些優點，但是要真正找到一個合適且滿足 Mercer's condition

表 2-1 kernel functions

| Classifier Type | Kernel function |
|---|---|
| Polynomial | $K(\mathbf{x}_i, \mathbf{x}) = [g(\mathbf{x}_i \cdot \mathbf{x}) + b]^p$ |
| Radial basis function (RBF) | $K(\mathbf{x}_i, \mathbf{x}) = \exp\{-\frac{\lvert\mathbf{x} - \mathbf{x}_i\rvert^2}{2\sigma^2}\}$ |
| Sigmoid function | $K(\mathbf{x}_i, \mathbf{x}) = \tanh[g(\mathbf{x}_i \cdot \mathbf{x}) + b]$ |

的函數並不容易，表 2-1 中列了幾個已知的 kernel functions 供參考，其中 RBF kernel 及 polynomial kernel 是較常被使用且有不錯表現能力的 kernel。

## 2.2 採用分解方法來加速 SVM

因為在求 SVM 的分割平面( separating hyperplane )基本上相當於在解一個 QP 問題( Quadratic Programming problem )。而且解 QP 問題時需要處理到 kernel matrix 的運算，而 kernel matrix 的大小與訓練資料的總數平方成正比[3][4]。待訓練的資料量越多時，kernel matrix 就越大，所需要的記憶體也就越多，一旦所需記憶體超過系統可用的記憶體空間，會導致求解困難而使得求解的速度變慢。

為了解決 kernel matrix 可能過大而導致系統記憶體無法滿足其需要的問題，Boser、Guyon 和 Vapnik 提出了"chunking method"[4][6]。"chunking method"將一個大的 QP 問題拆解成數個小的 QP 問題，然後依序解決每一個小的 QP 問題，來達到最終的目的：找出所有 Lagrange multipliers 非零的資料。因此其將訓練資料集合( training data set )分成工作集( working set )和非工作集合( non-working set )，工作集合中包含了 M 個違反 KKT 條件( Karush-Kuhn-Tucker conditions )的資料，其餘的則屬於非工作集合，然後以工作集合的資料去解其 QP 問題。如此反覆的進行，每次只保留該組工作集合的支撐向量( Support Vectors )，然後再從非工作集合中挑選 M 個違反 KKT 條件的資料，與保留下來的支撐向量形成新的工作集合，直到所有 Lagrange multipliers 非零的資料都被確認為止。

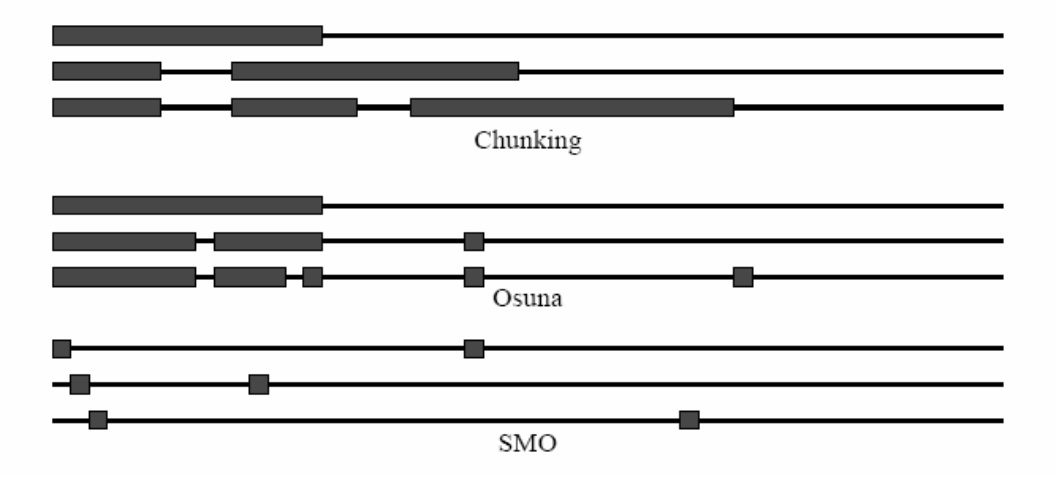



圖 2-2　三種訓練 SVM 的方法：Chunking, Osuna 的演算法和 SMO。對於每一種方法都展現三個步驟，每一條水平線表示訓練資料集合，而長條方塊代表在該步驟所最佳化的工作集合。對於 Chunking 來說，進行每一個步驟時都會加入固定數量的資料到工作集合中，而且會捨棄掉非支撐向量的資料，因此，工作集合的大小會越來越大。對於 Osuna 的演算法來說，工作集合的大小保持固定，也就是每次捨棄掉的資料與新加入的資料數量都一樣。對於 SMO 來說，工作集合中都只維持兩筆資料。

因為``chunking method``仍然有可能導致工作集合中的資料量太多，因此 Osuna 等人提出了另一種分解演算法( decomposition algorithm ) [7]。Osuna 等人建議工作集合的大小保持固定，也就是每次從工作集合中加入與移除的資料量要保持一樣。此外，Osuna 等人也證明了一個大 QP 問題可以被拆解成一些小的 QP 問題，只要每次的迭代都至少有一個違反 KKT 條件的資料被加入到工作集合中，則都會降低整體的目標函數( objective function )的值，因此將大的 QP 問題拆解成一些小的 QP 問題保證會收斂。

雖然 Osuna 解決的工作集合可能會過大的問題，但是其提出的方法在解 QP 問題時仍然會涉及到數值運算上的精確度的困難。因此 Platt 提出了 SMO ( Sequential Minimal Optimization ) [8]，其依據 Osuna 所證明出的將大的 QP 問題

拆解成小的 QP 問題保證可以收斂的定理，認為工作集合的資料總數只需要為 2，如此對於每個小的 QP 問題都可使用解析的方法( analytical method)來解而不涉及數值運算上的 QP 最佳化問題。上述三個方法的不同處可以由圖 2-2 來分辨。

## 2.3 其他加速方法

待訓練的資料量如果超過系統可用的記憶體量，支撐向量機( SVM )的求解過程就會遇到很大的問題。Marcelo Barros de Almeida 等人提出了 SVM-KM 的方法[10]，先對每類待訓練的資料作 K-means clustering 之後， 挑選每個叢集的中心點當作該叢集的代表資料，收集每個叢集的代表資料形成接下來支撐向量機(SVM)的訓練資料，以此來達到對訓練資料量的縮減而加快支撐向量機的訓練速度。雖然此方法能夠對訓練資料量做大量的縮減而大幅的加快訓練速度，但因為[10] 建議K-means clustering 中 K 的選擇最好等於資料量的 1/5，而對於 K-means clustering 來說，K 定的越大，所需的時間就越長。因此，雖然 SVM-KM 可以大幅節省支撐向量機的訓練時間，但是其前處理所花費的時間會隨著 K 越大，所需的時間越多，整理的時間並不都會有所節省。

此外，Pavlov 等人提出了 Boost-SMO 的演算法來加速 SVM 的訓練速度[11]，利用 boosting 演算法將訓練資料集合分成一些子集合，然後針對每一個子集合訓練出一個 SVM model，再對所得到的 SVM models 作線性的組合來作為

最後判斷的依據。這樣的方法容易受到訓練資料子集合的分布所影響，而導致最後的結果偏差太大。

# 第三章 藉由叢集來篩選資料 ( Data Selection by Clustering )

叢集( clustering )是分類資料的一種方法，因為位於同一群的資料通常擁有相似的特性，所以我們可以挑選一些位於同一群的資料來代表該群中所有的資料，藉以達到資料縮減的作用。

K-means clustering[14]，SOM( Self-Organizing Map )[15]，和 GMM ( Gaussian Mixture Models )等都是叢集( clustering )的方法。因為我們的目的是要加速 SVM 的訓練，因此叢集的速度是我們需要考慮的，因此本論文採用 K-means clustering 來當作我們分群的方法。因為速度上的考量, 我們採用了一些較有效率的 K-means clustering 方法 [13][14]。由 Alsabti 等人提出的加速版本的演算法[14]是將資料儲存於 kd-tree 的結構[16]中，kd-tree 以每個叢集的中心資料當作其根部( root )，而距離中心越近的資料儲存於越靠近根部的節點( node )，然後在處理分群動作時，只計算與根部較近的一些節點的資料，藉以達到運算量的縮減而加快 K-means clustering 的速度。本論文中採用此版本的 K-means clustering 演算法。

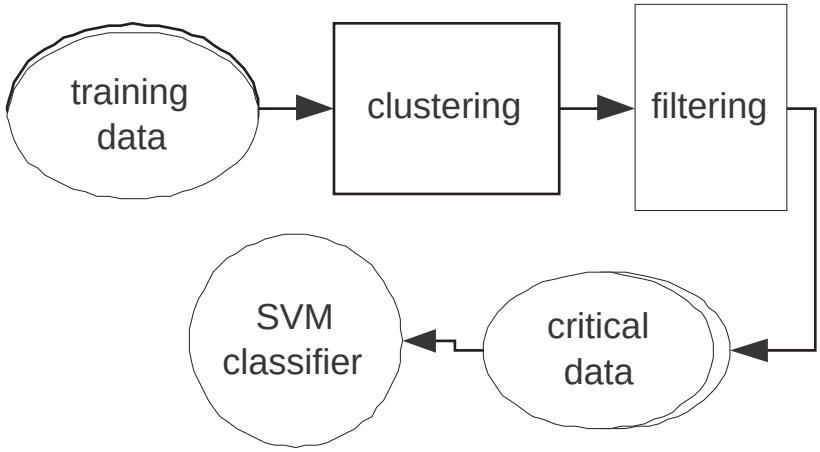
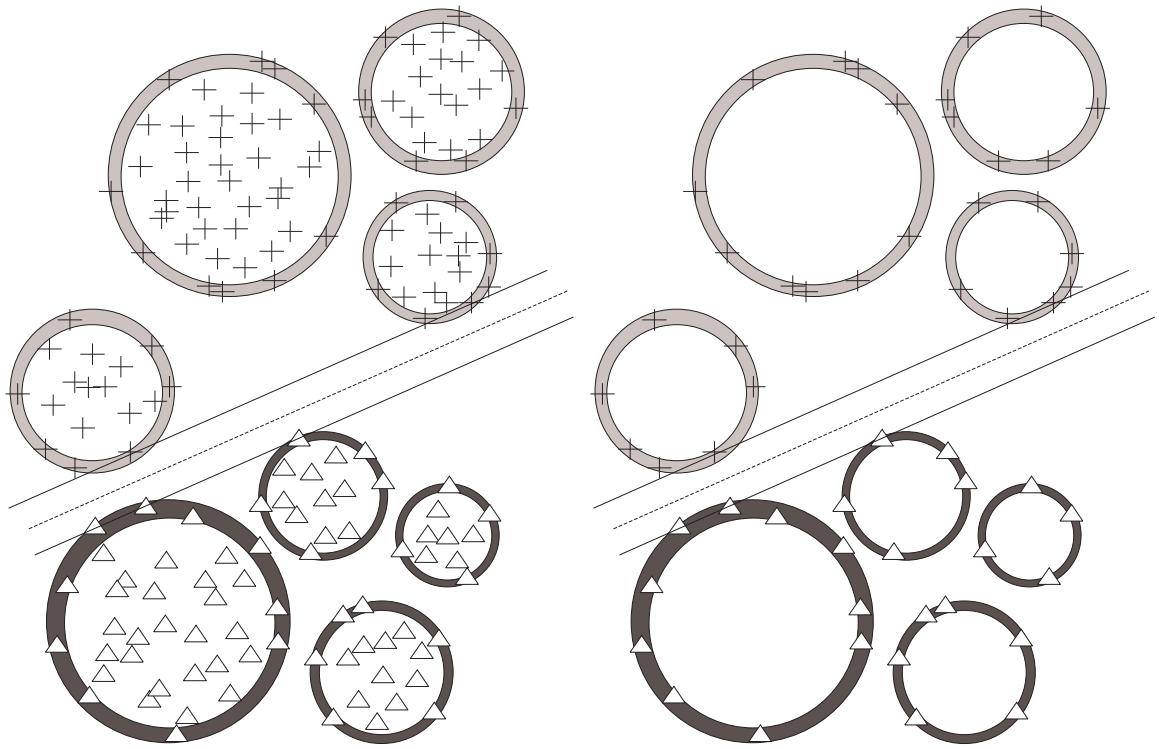



圖 3-1　　我們的方法架構圖

## 3.1　我們所提出的方法架構

圖 3-1 為我們所提出的方法的流程圖。首先，我們先對每一類別的資料作分群，然後挑選對 SVM 的辨別器( classifier )有較大影響的資料。 挑選的演算法“**filtering**”，如下敘述：

```
Let CS be the set of critical data
Initial CS ← ∅
for each cluster Cj
   for each data xij , where xij ∈ Cj
      if distance(xij, mj) > Tj , where mj is the centroid
                          of Cj and Tj is a threshold
         CS ← xij
   end for
   CS ← mj
end for
```



(a) (b)

圖 3-2 (a) 二分類的資料分佈，依據原始的資料所訓練出的 SVM 切割平面。淺灰色與深灰色區域爲每一個叢集的外圍部份。(b) 設定每一類分成 4 個叢集(cluster), 然後挑選每個叢集的外圍資料當作 SVM 的訓練資料所得到的切割平面。

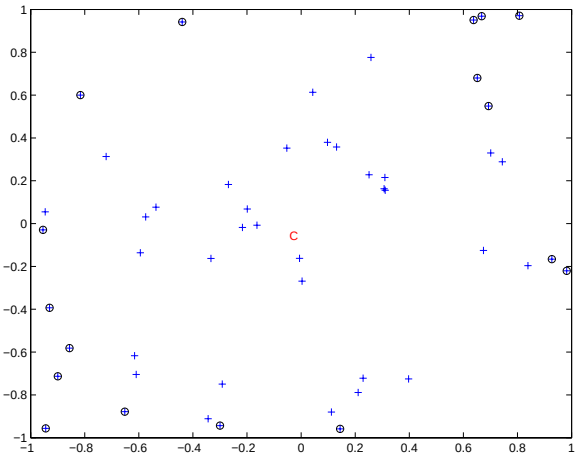
每一個叢集( cluster ) $C_j$ 都會搭配一個門檻值( threshold ) $T_j$，$T_j$ 的大小與每個叢集( cluster )呈正相關，叢集的半徑越大，其相對應的門檻值就越大。因為對於 SVM 的切割平面有決定性影響的資料(即支撐向量)大多是位於每類資料的邊界上，所以我們挑選位於每個叢集( cluster )的外圍的資料當作 SVM 的訓練資料。我們稱經由”filtering”挑選出來的資料為關鍵資料( critical data )。圖 3-2(a) 表示原始資料的分布狀況及依據原始的資料所訓練出的 SVM 切割平面，淺灰色及深灰色區域我們視為叢集( cluster )的外圍部分。圖 3-2 (b) 表示只依據外圍區域上的資料所訓練出的 SVM 切割平面。雖然圖 3-2 (a) 與圖 3-2 (b) 的切割平面是一樣的，但是它們決定出該平面所需要的資料量有所差異，圖 3-2 (b) 所需的資料量只

是圖 3-2 (a) 的一小部份，很顯然地，圖 3-2 (b) 在訓練過程中是比較有效率的。

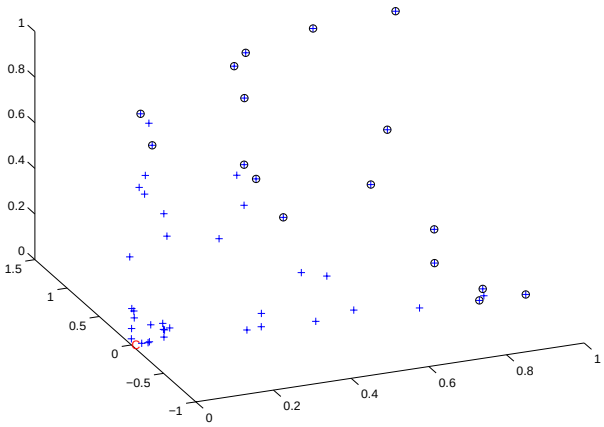
雖然利用 SVM 是將位於輸入空間( input space )中的資料轉換到特徵空間( feature space )，然後依據資料在特徵空間中分布尋找最佳的切割平面，但是我們相信資料的原始分布在被轉換到特徵空間後其分布的相對關係大多還是會維持。我們以 50 筆假造的資料( synthetic data )為例，其輸入空間為 2 維的空間，如圖 3-3 (a) 。我們利用一個非線性轉換的函數將這 50 筆資料轉到 3 維的特徵空間中，如圖 3-3 (b) ，轉換函數如下：

$$\phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2), \ \textit{where} \ \mathbf{x} = (x_1, x_2) \tag{3.1}$$

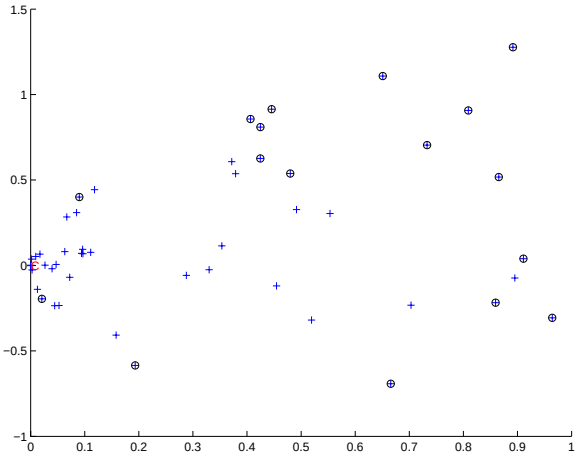
由圖 3-3 (b) 我們可以看到，以(3.1)式非線性轉換函數將原本屬於 2 維空間的資料轉換到 3 維空間後，其原本位於外圍的資料在 3 維的特徵空間中也大多落於外圍。此外，我們也發現這 50 筆資料的中心點經過轉換後，在特徵空間中也落於外圍的一角。又因為同屬於一個叢集( cluster )的資料會有相似的特性，且其中心點常被拿來代表該叢集( cluster )，因此一個叢集( cluster )的中心點也會是我們的關鍵資料( critical data )。圖 3-3 (c) 以不同的角度呈現轉換後資料的分布情形，從俯視圖更可以清楚的看出原本位於外圍的資料以及中心點經過轉換後，確實在特徵空間中大多落於外圍。




(a) 在 2 維的輸入空間中，50 筆假造的資料的分布狀況。



(b) 將資料轉換到 3 維空間的資料分布圖，使用的轉換函數爲 $\phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2),\ where\ \mathbf{x} = (x_1, x_2)$ 。



(c) (b)的俯視圖，我們可以看到挑選的那些資料經過轉換後大多落於外圍。

圖 3-3　2 維空間的資料經過非線性轉換到 3 維的特徵空間中後的資料分布情況。標記爲圓形和"C"的資料表示我們挑選的資料。

因此我們先對每一類的資料作分群的動作(採用 K-means clustering )，然後再經過"filtering"從 K 個叢集中挑選出我們的關鍵資料( critical data )來當作我們訓練 SVM 時的訓練資料。

## 3.2 多類別分類的問題 ( Multi-Classification Problems )

如果我們要解決的問題是二元問題的話，也就是只有兩種類別的分類問題，那麼我們可以直接將 SVM 應用到此問題上，因為 SVM 本質上就是被設計在解二元分類問題的。但是實際上，我們面對的問題大多不會只有"1"或"2"，而是多類別的分類問題，例如：要區分是"蘋果"，"西瓜"，"香蕉"等水果，或分辨是阿拉伯數字的"0"，"1"，…，"9"等數字。我們要如何去利用 SVM 來達到 N 類問題的分辨上呢? 在本論文中，我們以二元 SVM( binary SVM )為基礎，對於 N 類的問題我們會建立$C_2^n$=N*(N-1)/2 個 SVM models，每個 model 都只負責區分兩種類別，然後最後判定是屬於哪一類的方式採用多數決，也就是看這 N*(N-1)/2 個 models 中哪一類別的得票數最多。上述的方法就是所謂的 "one-against-another method"，還有另外一種方法，"one-against-rest method"，但效果比前者要差[17]。

# 第四章 實驗與結果討論

為了驗證我們提出的資料篩選方法對於辨識的正確性大致可以保持住，且確實可以加快大筆資料的訓練速度，本論文先採用 TCC-300 麥克風語音資料庫[18][19]來當作我們實驗的資料庫，來實驗語者辨識( Text-Independent Speaker Identification )的正確性。該資料庫利用麥克風錄製了 300 人的語音資料，聲音資料的格式如表 4-1：

表 4-1　TCC-300 麥克風語音資料庫的聲音格式

| | |
|---|---|
| 取樣頻率(sampling rate) | 16 KHz |
| 每個樣本的位元數(bits per sample) | 16 |

我們總共會從資料庫中任意挑選 40 個語者(20 個男生, 20 個女生)的語料來當作我們實驗的語者語料。我們所採用的語音特徵( features )為 13 維的 MFCC( Mel-Frequency Cepstral Coefficient )和其 Delta-MFCC，總共 26 維的語音特徵向量來代表一個單位長度( frame )的語音特徵, 在抽取特徵向量時，我們會先捨去語料中靜音( silence )的部分，然後單位長度( frame size )為 256 個 samples，每次平移半個 frame 的方式抽取 30 秒的語料來當作一個語者的訓練語料特稱向量集合。以 5 秒的語料長度來當作測試語料，共 1000 筆。訓練時，每個語者的訓練資料會有 3700 多筆，40 個語者的總訓練資料共 149760 筆。

在所有的實驗中，我們使用 LIBSVM [20] 這個套件來訓練 SVM models。

針對 TCC-300 語音資料庫，我們選用 polynomial kernel function，其 order 為 2，polynomial 的常數係數設為 4，而 SVM 的懲罰參數( penalty parameter )設為 250，藉以達到較好的準確度。

在將資料丟進 SVM 訓練前，我們會先將每筆訓練資料的每個維度的值標準化( normalize )到 0 和 1 的區間，因為 SVM 是以距離為基礎的判斷方法，為了讓每個維度所貢獻的距離影響性一樣，不致於因為某個維度的值特別大或特別小而左右了距離的大小，將每個維度的值標準化到 0 和 1 的區間可以消除此因素；另外，將每個維度的值標準化到 0 和 1 區間中，也可以減少 SVM 在計算過程中可能導致的計算精確度問題。此外，我們採用"one-against-another method"來建立我們的多類別 SVM models。

訓練好 SVM models 之後，我們以 5 秒來當作每次測試的語料基本長度，並以多數決( majority-voting )的方式來判定測試語料屬於哪個語者，判斷的準則如下：

Assume $\exists$ n frames in a test-unit utterance
, i.e, $\exists$ n feature vectors $\mathbf{x}_i$, i = 1~n

$$\text{Let} \quad \text{target} = \arg\max_j \left| \mathbf{x}_i : \mathbf{x}_i \in \text{class}_j, \forall\ \mathbf{x}_i \right|$$

$$\text{then} \quad \text{the utterance} \in \text{class}_{\text{target}}$$

表 4-2 在語者總數爲 20 個的情況下，當 K-means clustering 的 K 值設定爲 20 時，針對不同的篩選比率，在測試了 500 筆語料後的實驗結果。

| T | 準確度( accuracy ) | 標準差 | 挑選資料比例( % ) |
|---|---|---|---|
| 0.95 | 0.69 | 0.341 | 1.76 |
| 0.9 | 0.80 | 0.260 | 3.12 |
| 0.85 | 0.922 | 0.164 | 5.57 |
| 0.8 | 0.962 | 0.097 | 9.88 |
| 0.75 | 0.974 | 0.07 | 16.17 |
| 0.7 | 0.99 | 0.025 | 24.91 |
| 0.65 | 0.996 | 0.012 | 36.10 |
| 0.6 | 0.994 | 0.019 | 49.26 |
| 0.55 | 0.996 | 0.017 | 62.72 |
| 0.5 | 0.996 | 0.017 | 75.58 |

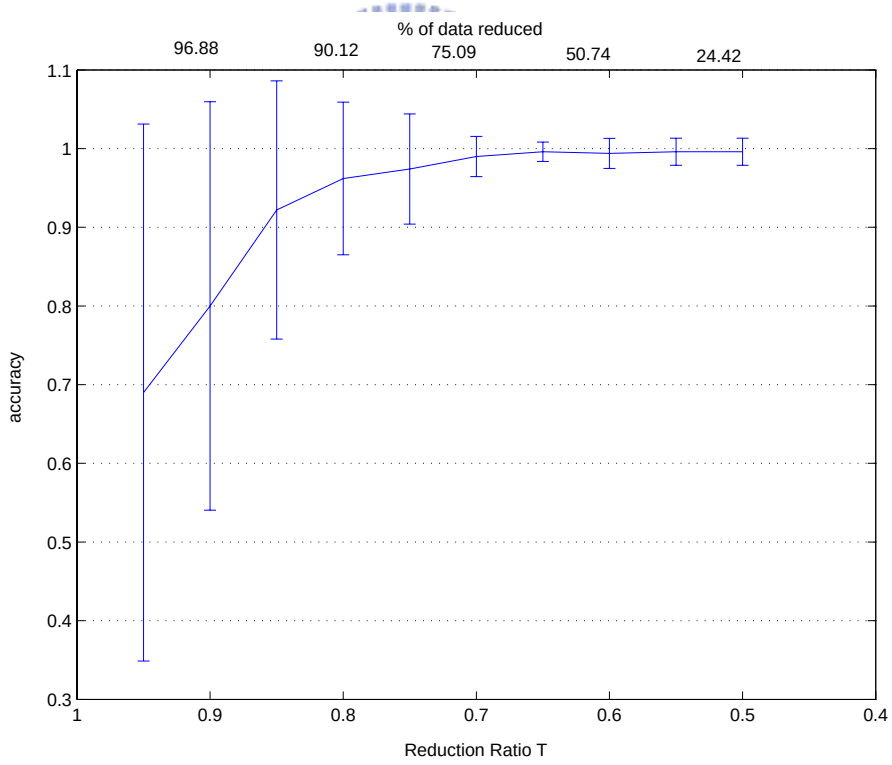



圖 4-1 在 20 個語者及 K=20 的情況下，不同的篩選比率 T 與相對的語者辨識準確度的曲線圖。"% of data reduced"表示節省的資料量百分比。每個語者的訓練語料長度爲 30 秒，測試語料單位長度爲 5 秒，共 500 筆測試語料。當 $T \le 0.7$ 時，辨識準確度的平均值以及標準差分別都高於 99%及低於 0.03。

## 4.1 不同的 K 值及門檻值的實驗比較

我們使用不同的篩選比率 T ( $=T_j/r_j$, 其中 $r_j$ 代表叢集 $C_j$ 的半徑長度)(參看第四章中的"filtering"演算法)和不同 K 值的 K-means clustering，以 20 個語者作了一系列的實驗分析，藉以尋找理想的 K 值與篩選比率 T。由表 4-2 和圖 4-1 可以看出，當篩選比率為為 0.7 時, 挑選的資料總數為原資料總數的 1/4，也就是節省了 3/4 的資料量，可以達到 0.99 的準確度和 0.025 的標準差。當篩選比率越來越小時，節省的資料量越來越少，但是準確度與 T=0.7 的差異不大，只有不到 1% 的差異且標準差也只相差 0.01 左右。因為訓練資料量的多寡，會影響到 SVM 的訓練時間；資料量越多，需要越多的訓練時間。因此為了能夠讓 SVM 的訓練時間能夠顯著的節省，在兼顧準確度的考量下，我們建議篩選比率 0.7 為理想的選擇。

接著，我們藉由選擇不同的 K 值來觀察 K-means clustering 的 K 值對 SVM 的效能的影響。由表 4-3 和圖 4-2 可以看到，當 K 值在 12 和 100 之間變動時，準確度變動範圍在 0.992 與 0.94 之間。但是，準確度的標準差的變動性卻有不同程度的差別，當 K 介於 18 和 21 之間時，其相對應的標準差是較小的(約 0.03 左右)，超出這個範圍的標準差就變得較大，如圖 4-2。我們猜測資料的真實群集( clusters )個數就落在這個範圍內。因此我們建議在使用我們的方法時，最好的參數設定值可以先以 K=20 及 T=0.7 來作試驗。

表 4-3 在語者總數爲 20 個而篩選比率爲 0.7 的情況下，不同 K 值在測試 500 筆語料後的實驗結果。

| K | 準確度( accuracy ) | 標準差 | 挑選資料比例( % ) |
|---|---|---|---|
| 12 | 0.976 | 0.072 | 22.03 |
| 13 | 0.968 | 0.059 | 22.70 |
| 14 | 0.978 | 0.067 | 23.19 |
| 15 | 0.962 | 0.096 | 23.46 |
| 16 | 0.984 | 0.042 | 23.72 |
| 17 | 0.948 | 0.109 | 24.65 |
| 18 | 0.988 | 0.025 | 24.53 |
| 19 | 0.984 | 0.035 | 24.02 |
| 20 | 0.990 | 0.025 | 24.91 |
| 21 | 0.992 | 0.027 | 25.05 |
| 22 | 0.978 | 0.057 | 25.94 |
| 23 | 0.978 | 0.051 | 26.10 |
| 30 | 0.972 | 0.087 | 28.02 |
| 40 | 0.972 | 0.104 | 31.05 |
| 50 | 0.976 | 0.094 | 33.08 |
| 60 | 0.964 | 0.061 | 35.76 |
| 100 | 0.942 | 0.198 | 43.88 |

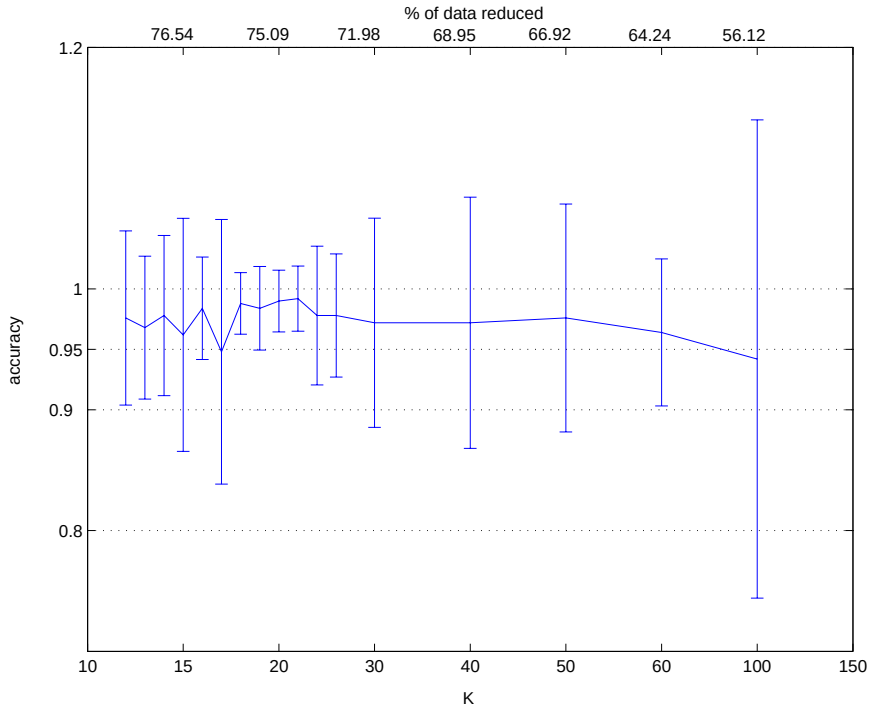



圖 4-2 在 20 個語者及篩選比率 T=20 的情況下，不同 K 值與相對的語者辨識準確度的曲線圖。當 18 ≤ K ≤ 21 時，正確率的標準差與其他的 K 值相較下是比較小的(大約在 0.03 左右)。

表 4-4　　SVM-KM 的實驗結果。

| K | 準確度( accuracy ) | 標準差 | 挑選資料比例( % ) |
|---|---|---|---|
| 20 | 0.186 | 0.241 | 0.53 |
| 50 | 0.772 | 0.312 | 1.31 |
| 100 | 0.826 | 0.338 | 2.63 |
| 150 | 0.93 | 0.233 | 3.94 |
| 200 | 0.948 | 0.215 | 5.25 |
| 250 | 0.954 | 0.198 | 6.57 |
| 300 | 0.962 | 0.163 | 7.88 |
| 350 | 0.96 | 0.172 | 9.20 |
| 400 | 0.964 | 0.155 | 10.51 |
| 700 | 0.974 | 0.112 | 18.39 |

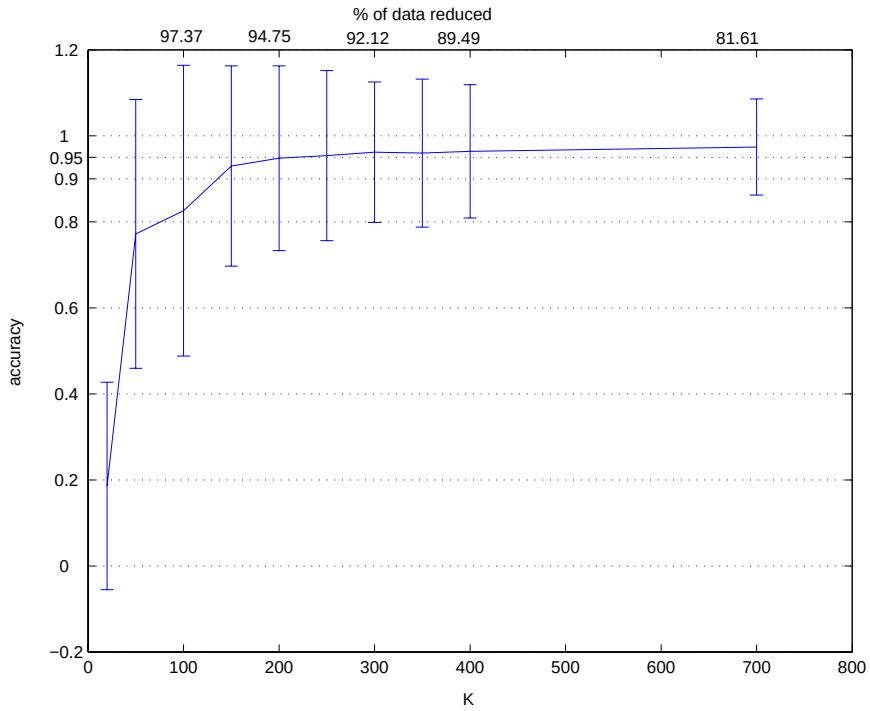



圖 4-3　　SVM-KM 方法對於不同 K 值的辨識準確度曲線圖(20 個語者)。此法可以達到不錯的準確度，但是其相對應的標準差也較大。

## 4.2　與 SVM-KM 的比較

為了試驗我們的方法確實值得被採用，我們以 SVM-KM[10]方法來做比較。因為[10]對於 SVM-KM 中 K 的建議值為 n/5( n 為資料量 )，我們針對不同的 K

表 4-5　我們提出的方法，SVM-KM(350)及 full-SVM 之比較數據結果。

| | **我們的方法( K=20 , T=0.7 )** | SVM-KM (K=350) | Full-SVM |
|---|---|---|---|
| 節省的資料量(%) | **75.09** | 90.8 | 0 |
| 篩選所花時間(min) | **4.67** | 800 | 0 |
| 訓練所花時間(min) | **23** | 2.5 | 185 |
| 整體花費時間(min) | **27.67** | 802.5 | 185 |
| 準確度(%) /標準差 | **99 / 0.025** | 96 / 0.172 | 100 |
| 辨識時間(sec) / SV 數量 | **5.846 / 6972** | 2.474 / 6651 | 23.387 / 59967 |

作了一系列的實驗，如表 4-4 和圖 4-3。當 K 小於 100 時，其效能相較下是不理想的。當 K ≥ 150 後，辨識的正確度才超過 90%(但還是低於我們所提出的方法( K=20，T=0.7， 準確度=99% ))，但其相對應的標準差都偏大( 0.112 ~ 0.233 )。此外因為 SVM-KM 所要求的 K 值對於大資料量來說， K 也要越大。我們的實驗中，對於每位語者的訓練語料的資料量大約在 3700 左右，我們以 K=350 為例，比較了其整個篩選資料以及訓練完成之後整體所需的時間如表 4-5。當 K=350 時，K-means clustering 平均花費的時間為 40 分鐘，20 個語者需要 800 分鐘，雖然訓練 SVM 只需要 2.5 分鐘，可是整體花費的時間卻是 full-SVM (沒有經過資料的篩選動作)的 4.3 倍左右，且是我們的方法(K=20，T=0.7)的 29 倍，而我們的方法( K=20, T=0.7 )所花費的時間是 full-SVM 的 0.15 倍，快了 85%左右，且準確度只降了 1%。此外我們的方法最後所得到的 SVM models 的辨識時間也比 full-SVM 快了 4 倍左右。由此可知，雖然在兼顧準確度的前提下，我們的方法所節省的資料量沒有 SVM-KM 多，但是整體所需的時間卻遠遠少於 SVM-KM，也確實的達到了加速的效果。此外 SVM-KM 的準確度標準差與我們的方法相較下都偏高，顯示我們的方法比 SVM-KM 有更好的強健性( robustness )。

表 4-6　　語者辨識的準確度。

| 語者總數 | **我們的方法( K=20, T=0.7 )** | Full-SVM | GMM(32) |
|---|---|---|---|
| 5 | **99.6 %** | 100 % | 100 % |
| 10 | **98.6 %** | 100 % | 98.9 % |
| 20 | **99.0 %** | 100 % | 93.8 % |
| 40 | **97.6 %** | 100 % | 80.3 % |

## 4.3　與未經過資料篩選的 SVM 及 GMM 的比較

為了驗證我們的方法的辨識準確度可以達到水準，我們針對不同的語者數量與 full-SVM 和 GMM(32) 作了實驗比較，如表 4-6。我們的方法( K=20，T=0.7 ) 與傳統的 full-SVM 相較之下，其辨識準確度是相當的，但是整體的速度快了 85% 左右，資料量也減少了 75%左右。此外，雖然在語者總數小於 10 人時，GMM(32) 的辨識準確度略高於我們的方法，可是當語者總數大於 20 人時，GMM(32)的準確度卻大幅的下降。從這些數據結果可以看出 (1) 我們的方法具有好的強健性 ( robustness )；(2) 我們提出的方法幾乎不影響 SVM 的辨識準確度；(3) 對於大資料量的應用上，我們的方法確實可以有效的提升 SVM 的訓練速度；(4) SVM 相較於 GMM 在辨識上擁有較好的強健性( robustness )。

雖然我們的方法所得到的辨識準確度，相較於 full-SVM 有些微的下降(大約 2%左右)，但是大幅的加快了訓練的速度及減少了資料量，進而減少了支撐向量

表4-7　　40 個語者的 SVM models 的支撐向量總數與辨識所需時間。

| | 我們的方法( K=20, T=0.7 ) | Full-SVM |
|---|---|---|
| 支撐向量總數 | **34762** | 126109 |
| 辨識時間(sec) | **15.741** | 68.970 |

( support vectors )的個數，表 4-7。因為 SVM 的辨識時間與支撐向量的總數成正比，也就是支撐向量越多，辨識花費的時間也越多，由表 4-7 中我們提出的方法辨識所需的時間比 full-SVM 快了 4 倍多。因此，如果準確度這個因素相較於其他因素是最重要的，那麼可以藉由我們的方法先快速的尋找出訓練 SVM 合適的參數。如果希望能夠使辨識的速度也加快，那麼可以使用我們的方法藉以達到較快的辨識效率且維持好的辨識準確度。

## 4.4　其他資料庫的實驗結果

由 TCC-300 語音資料庫的實驗結果可以知道，我們的方法對於語者辨識上的應用是有幫助且可行的，為了驗證我們的方法並非只適用在語者辨識上，我們從 UCI Machine Learning Repository[21] 下載了一些資料來測試，分別是 Letter Recognition Database 和 Optical Recognition of Handwritten Digits。首先，Letter Recognition Database 共含有 20000 筆資料，每筆資料的維度為 16 維，分 A~Z 共 26 類。我們從每類的資料中選取 500 筆資料當作訓練資料，因此共有 13000

筆的訓練資料，7000 筆的測試資料。對於我們的方法中的參數設定，我們採用 TCC-300 語音資料庫的結果，也就是 K=20, T=0.7，而 SVM 的參數設定上，因為準確度上的考量，我們選用 RBF kernel function，然後 c = 32, g = 0.5 ( g = $\frac{1}{2\sigma^2}$ )，實驗結果如表 4-8。

表 4-8　UCI-Lettet Recognition Database : 總共 26 類，每類 500 筆共 13000 筆的訓練資料, ，7000 筆的測試資料，選用 RBF kernel function, c = 32, g = 0.5 的實驗結果。

| | **Our method ( K=20, T=0.7 )** | SVM-KM (20) | SVM-KM (50) | SVM-KM (100) | SVM-KM (150) | SVM-KM (180) | Full-SVM |
|---|---|---|---|---|---|---|---|
| % of reduced data | **63.76** | 96 | 90 | 80 | 70 | 64 | 0 |
| Selecting time (s) | **19.86** | 19.32 | 29.23 | 40.97 | 54.38 | 60 | 0 |
| Training time (s) | **3.04** | 0.4 | 0.71 | 1.34 | 2.14 | 2.67 | 10.54 |
| Total time (s) | **22.9** | 19.72 | 29.94 | 42.31 | 56.52 | 62.67 | 10.54 |
| Accuracy (%) | **94.33** | 88.4 | 92.26 | 93.93 | 94.73 | 94.96 | 96.77 |

再者，Optical Recognition of Handwritten Digits 共含有 5620 筆資料，每筆資料的維度為 64 維，分數字“0”~“9”共 10 類，其中有 3823 筆訓練資料，1797 筆測試資料。我們採用的 K-means clustering 的 K 值一樣為 20，門檻值 T 同樣為 0.7，而因為辨識率的關係，SVM 採用 polynomial kernel function, order=4, c=32, g=0.5，實驗結果如表 4-9。

表 4-9　　UCI-Optical Recognition of Handwritten Digits : 總共 10 類，3823 筆訓練資料，1797 筆測試資料，選用 polynomial kernel function with order 4, c = 32, g = 0.5 的實驗結果。

| | **Our method ( K=20, T=0.7 )** | SVM-KM (20) | SVM-KM (50) | SVM-KM (100) | SVM-KM (150) | SVM-KM (200) | Full-SVM |
|---|---|---|---|---|---|---|---|
| % of reduced data | **44.02** | 94.77 | 86.92 | 73.84 | 60.76 | 47.68 | 0 |
| Selecting time (s) | **22.77** | 21.7 | 24.5 | 42.65 | 50.83 | 63.8 | 0 |
| Training time (s) | **1.24** | 0.2 | 0.31 | 0.52 | 0.75 | 1 | 2.56 |
| Total time (s) | **24.01** | 21.9 | 24.81 | 43.17 | 51.58 | 64.8 | 2.56 |
| Accuracy (%) | **98.16** | 96.27 | 97.16 | 97.83 | 97.89 | 97.94 | 98.11 |

由表 4-8 和表 4-9，其驗證了我們的方法不僅適用於語者辨識上，也適用於其他問題的辨識上。此外我們的方法相較於 SVM-KM 來說，由準確率和整體花費時間的觀點來看，我們的方法有較好的強健性。雖然這兩個實驗所花費的整體訓練時間並沒有達到加速的目的，但是也沒有因為訓練資料的減少而使得準確度大幅的下降。因此，我們的方法對於大資料量的例子才能真正展現其效果，而對於小資料量的例子使用現今的 SVM 訓練方法已經可以快速的達到目的。

# 第五章 應用：從新聞中切出氣象播報片段

天氣的好壞容易影響到一個人的心情，進而影響到當天的工作或學習的成效。如果能夠確實掌握好天氣的動態，心理較能夠有事先的調適與準備，使天氣對於自己的心情影響降到最低，並且不會因為氣候的變化而打亂的計劃。另一方面還可以藉由氣象主播的解說與分析，增加自己在氣候上的知識。因為這些原因，我們想讓使用者能夠立即的得知天氣消息，且有具備生動且知識性的氣象播報。因此藉由 SVM 在語者辨識上面的能力，將新聞節目裡面的氣象報導獨立出來，讓一般大眾可以快速的得知具備生動與知識性的氣象播報，而不是單純的只獲得天氣狀況的數據。

在接下來的應用當中，我們以華視新聞為對象，並且假設氣象播報時有獨立的氣象主播在報導，我們希望偵測出華視新聞中氣象播報的開始與結束時間，然後將該段內容抽取出來成為可以直接觀看的片段。

## 5.1 應用前的處理分析

為了能夠偵測出氣象播報的時間，我們需要訓練一些 SVM models 來分辨不同的語者，然後再利用這些 models 來對新聞做辨識，將新聞的內容對不同的語者做分類，以找到氣象主播說話的起始與結束點。當我們決定這麼做時，首先

會面臨到一個問題，因為一個完整的新聞節目中，出現的語者不只一人，且每天出現的語者也不見得都一樣(每天新聞的外景主播以及事件的主角人物，甚至新聞節目中穿插的廣告)，所以為了達到有效的對新聞節目中語者的分類且能夠將氣象主播的片段突顯出來，待訓練語者的選擇將會影響到新聞中語者的分類好壞。此外，如果訓練的語者數量太多， 會影響到判斷的速度。因此，為了兼顧準確度以及速度，待訓練語者的選取及語者數量的決定是整個過程中首要注意的問題。

本實驗中，我們總共選擇了 10 個語者來訓練出我們的 SVM models，10 個語者分別為李四端、周明華、莊開文、張彭雯、徐俊相、陳來發、王欣怡、竹幼婷、一名外場男記者和外場女記者，不包含廣告中的人物，因為新聞中，每則廣告出現的時機次數都不定，且廣告的變化性相當大，所以如果把廣告中人物的聲音加入我們的訓練語料中並不是一個好的做法，對於新聞的語者分類也不合適，且廣告不是我們所要尋找的目標，因此我們不特別去針對廣告做分類。我們選擇的語者的條件主要考量為出現的可能性與時間性。在一段新聞節目中，主播是肯定且出現機會最高的一個語者，因此為了對新聞內容做一個適當的切割分類，新聞主播一定是我們要訓練的語者。在我們 10 個語者中，李四端、周明華、莊開文、張彭雯和徐俊相等五個都是可能的新聞主播。另外，因為新聞節目中，一定會有新聞現場或負責講述該則新聞事件始末的記者，所以我們也將一些該類記者的語料加入我們的訓練對象中(我們分別選擇了一個男性的外

場記者和女性的外場記者)。顯然的，氣象主播是必然的訓練對象，所以我們把陳來發、王欣怡和竹幼婷三個氣象主播也加入了我們的訓練語料中。

我們採用 12 orders 的 Mel-Frequency Cepstral Coefficients ( MFCC )和 Delta-MFCC 共 24 維，而 frame size 為 512 個 samples 來當作語者的 features。在抽取 features 前我們將靜音( silence )的部分捨去，增加所抽取 features 的獨特性， 以利於 SVM models 的準確性。每個語者的訓練語料長度為 30~40 秒，K-means clustering 的 K=20、T=0.7，以"one-against-another method"建立多類別的 SVM models，採用 RBF kernel function 而懲罰參數( penalty parameter ) C=32、$\sigma^2$ =1。

為了讓待辨識的語料長度足以被準確的判斷，我們所設定的辨識單位長度不能夠太短，但為了讓我們所找出的氣象報導的起始與結束時間準確，我們所設定的辨識單位長度不能太長，因此我們選定 3 秒的長度為我們的辨識語料長度，如此可以兼顧判斷的準確性以及氣象報導起始與結束時間的精準度。因此一個小時的新聞節目, 總共會有 1200 個待辨識單位，我們從這 1200 個辨識單位的結果來找出氣象播報的位置。

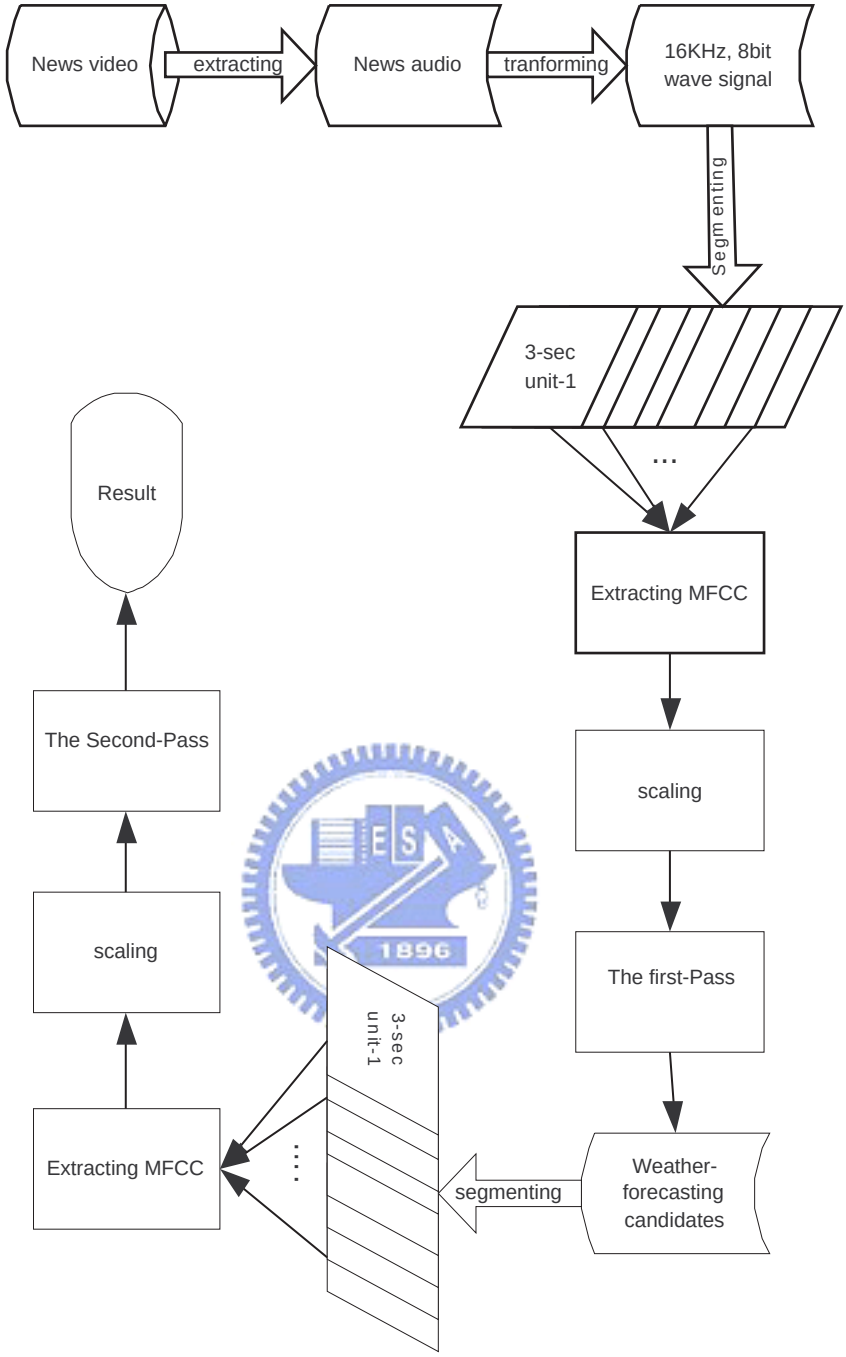



圖 5-1　　氣象播報偵測流程圖。

## 5.2　流程與方法

圖 5-1 為我們的偵測氣象播報片段的流程圖，我們依據聲音的特性來判斷氣象播報的位置，首先我們會從錄好的新聞影片(聲音的取樣頻率為 44.1K)抽取出

其聲音檔，並且將其轉換成 16K 取樣頻率及 8bps 的 wave 檔，以 frame size 為 512 的格式來抽取 features 供我們事先訓練好的 SVM models 來辨識。我們把 wave 轉換成 16K 的取樣頻率的用意為減少需要處理的資料量，且可以藉由降低取樣頻率以減少一些高頻的雜訊，而使得所抽取的 features 受雜訊影響的成分減少來達到較好的辨識率。對於語者辨識來說，辨識單位的聲音長度如果太短，其辨識的結果準確度會十分不理想，為了兼顧準確性以及偵測出的開始與結束時間的精準度，我們每次辨識的聲音單位為 3 秒的聲音 features。將每組待辨識的 features 丟進我們的 SVM models 中，會得到一個 label 及可靠度的數據，其中 label 表示該單位語料被判定為哪位語者而可靠度表示判定結果的可信度的高低(越接近 1 表示越可靠，越接近 0 表示越不可信)，待所有的待辨識語料的 features 都被辨識完後，會得到由 1200 個 (label, 可靠度) 所組成的數列(圖 5-2)，

| | |
|---|---|
| 2 0.647059 | 2 0.647059 |
| 2 0.294118 | 2 0.294118 |
| 2 0.661765 | 2 0.661765 |
| 1 0.348485 | 1 0.348485 |
| 1 0.511111 | 1 0.511111 |
| 1 0.564103 | 1 0.564103 |
| 2 0.571429 | 1 0 |
| 1 0.792208 | 1 0.792208 |
| 1 0.760563 | 1 0.760563 |
| 1 0.692308 | 1 0.692308 |
| 5 0.661765 | 1 0 |
| 3 0.757143 | 3 0.757143 |
| 3 0.764706 | 3 0.764706 |
| 6 0.788732 | 6 0.788732 |
| 6 0.74359 | 6 0.74359 |
| 1 0.731343 | 1 0.731343 |
| 1 0.714286 | 1 0.714286 |
| 7 0.805556 | 1 0 |
| 1 0.657143 | 1 0.657143 |
| 8 0.757143 | 8 0.757143 |
| 4 0.786667 | 8 0 |
| 8 0.813333 | 8 0.813333 |
| 1 0.742857 | 1 0.742857 |
| 1 0.746835 | 1 0.746835 |
| 9 0.753425 | 9 0.753425 |
| 9 0.690141 | 9 0.690141 |
| (a) | (b) |

圖 5-2　(a) (label, 可靠度)數列的例子。 (b)對(a)作 smoothing 之後的結果，我們將突然變化的部分作校正並將校正的 label 所對應的準確度設爲 0。

這組數列表示著每個單位的聲音屬於哪位語者。然後再經由平順化( smoothing )

的動作將這組數列做細微的修正，以符合一般語音表現的特性。何謂一般語音表現的特性？一般來說，一段影片中聲音的表現與變化是平緩的，突然變化的狀況是少見且不合理的，基本上拍攝與剪輯人員在處理多位語者的談話內容時，不會在急短的時間內讓一段由單一語者的聲音的語音內容中突然的變換或穿插另一語者的聲音(即使有，對我們來說也是無關緊要的)，且不同語者之間的轉換是平緩的，由前一語者慢慢的變換到另一語者。基於上述的特性，我們將得到的數列依據下列原則來進行修正，

```
if   label_j ≠ label_j-1
     if   label_j-1 = label_j+1
          label_j ← label_j-1
     else if   label_j ≠ label_j+1
          label_j ← label_j-1
end
```

其中 $label_{j-1}$, $label_j$, $label_{j+1}$ 表示在第 j-1, j, j 個辨識單位的 labels，在 smoothing 的過程中，我們一次觀察三個單位，如果第 j 個單位的 label 與其前後單位的 labels 不同時我們才會做調整，例如：1－3－1 ➔ 1－1－1 或 1－2－3 ➔ 1－1－3 。然後依據修正後的數列來判斷可能的氣象播報片段。我們假設一段氣象報導的時間至少在一分鐘以上，這也是相當合理的假設，因為一個完整且可以清楚表達訊息的報導，如果沒有足夠長的時間往往沒辦法清楚交代。此外一個完整的新聞節目出現的語者不計其數，對於不是我們訓練的 10 個語者的聲音的辨識單位，我們會給予一個假的 label。為了要從這一串數列(label，可靠度)中找出可能的氣象播報片段，我們必須設定一個門檻值( threshold )來把那些假的 label 的

對象給捨棄掉。假若一個不屬於我們訓練的語者的待辨識單元，即使會被判定成屬於某一個語者，但其相對應的可靠度肯定是比較低的。因此藉由門檻值的選擇，我們可以把大部分假的目標淘汰掉。但是門檻值的選擇是一個困難但又重要的一個決策，假如門檻值設定的太大，則我們可能會錯失目標物；

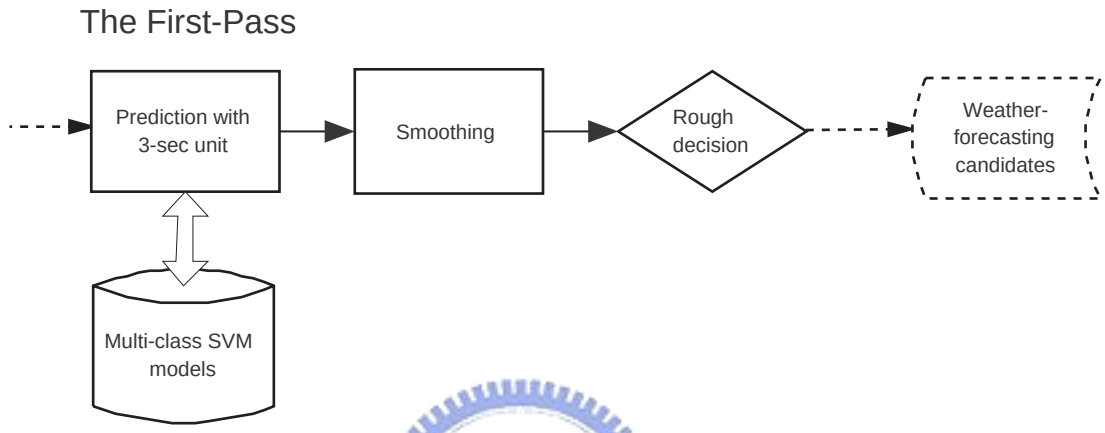



圖 5-3　　The First-Pass 流程圖。

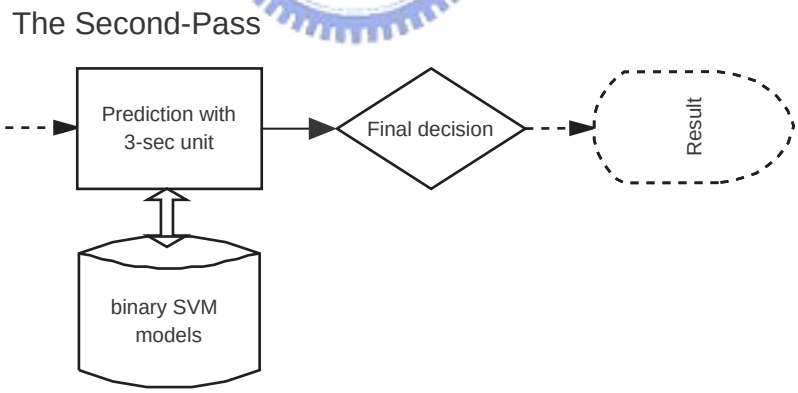



圖 5-4　　The Second-Pass 流程圖。

如果門檻值設定的太小，則會有很多錯誤的判定。因此，為了讓門檻值的選擇不影響到判定的結果，我們採用了 2-pass 的方法。第一個 pass 如圖 5-3，設定一個較寬鬆的門檻值(本實驗採用的門檻值為 0.65 )，經由"rough decision"找出

可能的氣象播報片段，然後再經由第二個 pass (圖 5-4) 的處理將錯誤的對象過濾掉，留下正確的氣象播報片段。"rough decision"的演算法如下：

假設氣象播報的聲音單位個數至少爲 D 個且門檻值爲 T
Step 1：根據 T 及氣象主播的類別尋找可能的氣象播報起始點。
Step 2：找到可能起始點後，再依據氣象主播類別繼續往下找可能的結束點。
Step 3：假設找到的可能氣象播報片段的聲音單位個數多於 D 個
則 將此段標記爲可能的氣象播報片段，且繼續處理 Step 4
否則 回到 Step 1 。
Step 4 ：根據氣象主播的類別將可能的起始點位置往前調整，然後回到 Step 1。

演算法 1 "rough decision"演算法。

我們依據"rough decision"演算法持續地去判斷可能的氣象播報片段，直到處理完所有的 labels。然後我們會得到可能的氣象播報片段，再利用一個 binary SVM model 對該聲音片段做重新的辨識，得到一組(label，可靠度)的數列，再對這組數列做"final decision"，其演算法如下：

假設氣象播報的聲音單位個數至少爲 K 個
Step 1：根據氣象主播的類別尋找氣象播報的起始點。
Step 2：找到起始點後，再根據氣象主播的類別繼續的往下找結束點。
Step 3：假設找到的氣象播報片段的聲音單位個數多於 K 個
則 將此段標記爲氣象播報片段。

演算法 2 "final decision"演算法

做完"final decision"之後所得到的結果就是我們想要的氣象播報片段的開始與結束時間。

對於 binary SVM model 來說，我們將氣象主播的語料視為一類，其他聲音

的語料視為另一個，其他聲音包含了廣告當中的聲音以及一些非氣象主播的人聲。我們總共蒐集了 4 分鐘左右的其他聲音的語料及 4 分鐘的氣象主播語料，採用的資料篩選及 SVM 的參數設定都與多類別的 SVM models 一樣。

表 5-1　　華視新聞的氣象播報偵測結果( 5/11 ~ 7/5 )。除了沒有專屬氣象主播的新聞以及錄製失敗的新聞( 6/14 午間新聞 )外，總共有 95 則的新聞節目，氣象播報的開始與結束時間 ± 3 秒是我們允許的誤差範圍。

| 正確 | 錯誤 | 準確率( % ) |
|---|---|---|
| 91 | 4 | 95.8 |

## 5.3　結果

我們蒐集了 5/11 ~ 7/5 共 56 天的華視午間及晚間新聞，除了沒有專屬氣象主播的新聞及錄製失敗的新聞(6/14 華視午間新聞)之外，共 95 則的新聞。因為我們設定的最小辨識單位長度為 3 秒鐘，所以最後的結果會有 ±3 秒的誤差。我們認為在這 ±3 秒的誤差都屬允許的範圍內，其實驗的結果如表 5-1。實驗的結果的正確率為 95.8 %，在 95 則新聞中總共有 4 則新聞不正確，其中有兩則新聞是因為開始或結束時間的誤差超過了我們設定的範圍而被判斷成錯誤的狀況，分別為 5/19 的午間新聞和 6/6 的晚間新聞。5/19 的午間新聞是因為判定的氣象播報結束時間比實際的時間多了 8 秒，而 6/6 的晚間新聞則是判定的開始時間比實際的時間提早了 21 秒，但是整段氣象播報的片段還是有在判定的片段中。所以整體來看我們的方法對於偵測氣象播報有相當好的效果，也驗證了 SVM 的

強大功能。此外，我們將這個方法實際的放進實驗室的新聞系統中，有興趣的人可以參觀 http://140.113.216.64/NewsQuery/main.asp 這個網址，我們以標題"華視氣象"代表偵測到的氣象播報片段。

# 第六章 結論與未來工作

我們提出了一個以叢集為基礎( cluster-based )的方法，對於大資料量的訓練過程的整體訓練時間可以達到有效的節省。我們挑選叢集的外圍資料以及其中心點來當作我們的訓練資料，藉由資料量的縮減來加速 SVM 的訓練。我們所設定的 cluster 數量不需要很多，因而可以達到快速的分群，且挑選出來的資料所訓練得到的 SVM models 對於辨識的準確度也和不經由資料的篩選所得到的 SVM models 相當。此外，我們提出的方法對於支撐向量的個數也達到了減少的作用，進而節省了辨識所需花費的時間。

我們也成功的將其應用到實際的新聞節目中，藉由訓練好的 SVM models 我們能夠準確的偵測出新聞中氣象播報的片段。除了利用 SVM models 之外，我們的 2-pass 方法也確實的幫我們避免掉門檻值選取的困難，而不錯過氣象播報的片段。

由於目前只由實驗驗證了我們的方法的可行性，尚未經由數學上的推導來加以證明，因此未來我們希望可以給出一個合理的數學式子以及有辦法經由一些算式之後給出一個明確而有依據的參數設定值。此外，希望能夠探討加速辨識速度上的問題，因為 SVM 在作辨識時，需要使用到支撐向量，如果能夠在支撐

向量的總數上達到有效率的縮減，或者降低特徵空間的維度，而降低 SVM 所找到的最佳切割平面的複雜度，達到加快辨識速度的效果。

# 參考文獻


1. D. A. Reynolds, R. C. Rose, ”Robust Text-Independent Speaker Identification using Gaussian Mixture Speaker Models”, IEEE Transactions on Speech and Audio Processing, Vol. 3, No. 1, January 1995.

2. Xin Dong, Wu Zhaohui, “Speaker Recognition Using Continuous Density Support Vector Machines”, ELECTRONICS LETTERS, 16th, August 2001.

3. Vladimir N. Vapnik, Statistical Learning Theory, John Wiley and Sons, Inc., New York, 1998.

4. Christopher J.C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, Data Mining and Knowledge Discovery, 1998.

5. T. M. Cover, “Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition”, IEEE Transaction on Electronic Computers, Vol. 14, pp. 326-334, 1965

6. B. E. Boser, I. M. Guyon, V. N. Vapnik, ”A Training Algorithm for Optimal Margin Classifiers”, In Proc. 5th ACM Workshop on Computational Learning Theory, pp. 144-152, 1992.

7. Edgar Osuna, Robert Freund, Federico Girosi, “An Improved Training Algorithm for Support Vector Machines”, In Proc. of the 1997 IEEE Workshop on Neural Network for Signal Processing, pp. 276-285, 1997.

8. John C. Platt, ”Fast Training of Support Vector Machines Using Sequential Minimal Optimization”, In Advances in Kernel Methods: Support Vector Learning, MIT Press 1998.

9. Michael Schmidt, Herbert Gish, ”Speaker Identification via Support Vector Classifiers” IEEE ICASSP, 1996.

10. Marcelo Barros de Almeida, Antônio de Pádua Braga, João Pedro Braga, “SVM-KM: speeding SVMs learning with a priori cluster selection and

k-means", IEEE 6 th Brazilian Symposium on Neural Networks, pp.162-167, 2000.

11. Dmitry Pavlov, Jianchang Mao, "Scaling-up Support Vector Machines Using Boosting Algorithm", International Conference of Pattern Recognition, Vol. 2, September, 2000.

12. Mokhtar S. Bazaraa, Hanif D. Sherali and C. M. Shetty, Nonlinear Programming: Theory and Algorithm, John Wiley and Sons, Inc., New York, 1993.

13. A.K. Jain, M.N. Murty, "Data Clustering: A Review", ACM Computing Surveys, vol. 31, no. 3, pp. 264-323, 1999.

14. K. Alsabti, S. Ranka, V. Singh, "An Efficient k-means Clustering Algorithm", Proc. First Workshop High Performance Data Mining, Mar. 1998.

15. Kohonen, T., "The self-organizing map", Proceedings of the IEEE ,Volume: 78 , Issue: 9 , pp. 1464-1480 , Sept. 1990.

16. J. L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching", Communications of the ACM, vol. 18, issue 9, pp. 509-517, September, 1975.

17. Chih-Wei Hsu and Chih-Jen Lin, "A Comparison of Methods for Multiclass Support Vector Machines", IEEE Transactions on Neural Networks, vol 13, pp. 415-425, 2002.

18. TCC-300 speech database.. Association for Computational Linguistics and Chinese Language Processing, Institute of Information Science, Academia Sinica, Nangkang, Taipei, ROC. [Online]. Available: http://rocling.iis.sinica.edu.tw/ROCLING/MAT/TCC-300brief.htm

19. Hsiao-Chuan Wang, "Speech Corpora and ASR Assessment in Taiwan", In Proc. of Oriental COCOSDA Workshop, Beijing, China, Oct. 16, 2000.

20. Chih-Chung Chang and Chih-Jen Lin, "LIBSVM : a library for support vector machines", Software available at

http://www.csie.ntu.edu.tw/~cjlin/libsvm

21 C. L. Blake, C. J. Merz, ”Repository of machine learning databases”, University of California, Irvine, Dept. of Information and Computer Sciences, 1998.
URL: http://www.ics.uci.edu/~mlearn/MLRepository.html