

國立雲林科技大學  
電子工程系碩士班

碩士論文

語音辨識與對話策略

A Study on Speech Recognition and Dialogue Policies

研究生：陳增安

指導教授：黃永廣博士

中華民國 九十三年六月

# 語音辨識與對話策略

A Study on Speech Recognition and Dialog Policy

研究 生：陳 增 安  
指導 教授：黃 永 廣 博 士

國立雲林科技大學

電子工程碩士班

碩士論文

A Thesis  
Submitted to  
Institute of Industrial Design  
National Yunlin University of Science & Technology  
in Partial Fulfillment of the Requirements  
for the Degree of  
Master of Science  
in  
Electronic and Information Engineering

July 2004  
Douliu, Yunlin, Taiwan, Republic of China

中華民國 九十二 年 六 月

# 語音辨識與對話策略

學生：陳增安

指導教授：黃永廣

國立雲林科技大學電子工程系碩士班

## 摘要

在本論文中，我們將語音辨識與對話策略結合在一起。在辨識系統中，我們利用梅爾頻譜分析（MFCC）取得特徵值，然後再使用向量量化（VQ）方式取得碼書，再用隱藏式馬可夫模型（HMM）訓練出每個聲音的模型；辨識方面是利用維特比（Viterbi）找出最接近的樣本聲音。我們利用語音辨識器來辨識關鍵字後，當作對話系統的輸入。在對話系統的對話策略方面，我們運用強化學習（RL theory）並以在咖啡店裡點咖啡為對話範例，讓代理人扮演店員的角色來訓練其對話策略，使得店員知道何時問什麼問題較為合理。

# **A Study on Speech Recognition and Dialogue Policies**

Student: Zhen-An Chen      Advisor: Dr. Wing-Kwong Wong

Institute of Electronic Engineering  
National Yunlin University of Science & Technology

## **ABSTRACT**

This thesis studies two aspects of a dialog agent that communicates with human users in natural language. The first aspect is speech recognition, which recognizes voice signals as textual sentences. The second is dialogue policy, which determines how the agent responds to a user in various conditions. The agent can improve its dialogue policy with a reinforcement learning algorithm.

## 誌謝

在校這兩年，首先感謝指導教授黃永廣老師在研究、課業上細心的教導，讓能順利完成這論文。感謝已畢業的學長們、勝程學長、晰勛學長和勝楷、錦堂、勇全和人工智慧實驗室中所有成員，包括助理律均、欣恬，大學部的子平，所有朋友們，無論是在課業上、生活中給予我支持與幫助。特別感謝國科會計劃 NSC 92-2520-S-224-001、國科會數位學習國家型科技研究計劃 NSC 93-2524-S-224-002 與志祥科技股份有限公司。另外感謝我的父母親與家人們給予我最大的支持與鼓勵，由衷的感謝。

## 目 錄

中文摘要.....	i
英文摘要.....	ii
誌謝.....	iii
目錄.....	iv
表目錄.....	vii
圖目錄.....	viii

### 第一章 緒論

1.1 研究動機.....	1
1.2 研究方法.....	2
1.3 論文架構 .....	2

### 第二章 文獻探討

2.1 語音辨識的技術.....	4
2.2 對話策略探討.....	6

### 第三章 語音辨識

3.1 研究方法.....	11
3.2 聲音特徵值之計算.....	14
3.2.1 高頻訊號之補強.....	14
3.2.2 漢明視窗.....	14
3.2.3 離散傅立葉轉換.....	15
3.2.4 濾波器處理.....	16
3.2.5 對數能量運算及梅爾倒頻譜計算.....	17
3.3 訓練語音模型.....	17
3.3.1 HMM 設計之理論基礎.....	20
3.3.2 HMM 機率計算.....	22
3.3.2.1 正算程式 (Forward procedure).....	22
3.3.2.2 逆算程式 (Backward Procedure).....	23
3.3.2.3 Forward-Backward Recursion.....	\.24
3.3.3 訓練 HMM 模型.....	24
3.4 辨識方式.....	26
3.4.1 維特比演算法.....	26
3.4.2 測試辨識器.....	28
3.4.2.1 實驗內容設定.....	28
3.4.2.2 實驗結果.....	29

3.3.2.3 討論.....	31
3.5 本對話策略的對話內容詞彙辨識.....	31

## 第四章 強化學習運用於對話策略理論基礎

4.1 強化學習理論介紹.....	33
4.2 對話策略訓練.....	33
4.3 實驗和討論.....	35
4.3.1 最短路徑學習.....	35
4.3.1.1 店員模型.....	35
4.3.1.2 決定動作方法.....	36
4.3.1.3 即時回饋的給定.....	36
4.3.1.4 顧客(環境模型).....	37
4.3.1.5 實驗過程.....	38
4.3.1.6 實驗結論.....	48
4.3.2 學習確認.....	49
4.3.2.1 決定動作方法.....	50
4.3.2.2 回饋的給定.....	50
4.3.2.3 實驗過程.....	51
4.3.2.4 實驗結論.....	52

## 第五章 總結

5.1 組合語音辨識和對話策略.....	55
5.2 討論.....	56
5.2.1 語音的問題.....	56
5.2.2 對話策略.....	57
5.3 未來和展望.....	58
參考文獻.....	59

## 圖目錄

圖 1.1 系統方塊圖.....	2
圖 2.1 語音辨識與語言模型：語音訊號處理之流程.....	5
圖 2.2 NJFun system 的對話範例.....	7
圖 2.3 不做確認動作的對話範例.....	8
圖 3.1 語音辨識系統方塊圖.....	11
圖 3.2 辨識結果給對話策略.....	11
圖 3.3 LPCC 技術與 MFCC 技術的比較.....	13
圖 3.4 訊號處理.....	14
圖 3.5 交叉切割.....	15
圖 3.6 傅立葉轉換.....	16
圖 3.7 特徵向量兌換觀測符號示意圖.....	18
圖 3.8 Voronoi regions 示意圖.....	19
圖 3.9 本研究 HMM 之設計圖：使用 4 個狀態.....	21
圖 3.10 正算程式示意圖.....	23
圖 3.11 逆算程式示意圖.....	24
圖 3.12 $x_t(i, j)$ 示意圖.....	25
圖 4.1 Sarsa 演算法.....	34
圖 4.2 狀態轉移圖.....	36
圖 4.3 固定主動程度=0 的結果.....	40
圖 4.4 固定主動程度=0.25 的結果.....	41
圖 4.5 固定主動程度=0.5 的結果.....	42
圖 4.6 固定主動程度=0.75 的結果.....	43
圖 4.7 固定主動程度=1 的結果.....	44
圖 4.8 固定反覆程度=0 的結果.....	45
圖 4.9 固定反覆程度=0.25 的結果.....	46
圖 4.10 固定反覆程度=0.5 的結果.....	47
圖 4.11 固定反覆程度=0.75 的結果.....	48
圖 4.12 「學習確認」極端例子的回饋 ( $\epsilon = 0.1$ ).....	53
圖 4.13 「最短路徑學習」極端例子的回饋 ( $\epsilon = 1/\text{回合}$ ) .....	54
圖 5.1 詳細的系統方塊圖.....	55
圖 5.2 執行結果.....	56

## 表目錄

表 2.1 語音辨識與語言模型：各種文法比較之結果.....	4
表 2.2 NJFun 系統狀態的 7 個屬性.....	9
表 2.3 NJFun 系統所有狀態與對話動作的對應表.....	10
表 3.1 辨識測試樣本.....	28
表 3.2 一個詞訓練一個模型.....	29
表 3.3 五個相同的詞訓練一個 Model.....	30
表 3.4 全部相同詞（20 個）訓練一個 Model.....	30
表 3.5 實驗數據整理.....	31
表 3.6 對話策略的對話內容詞彙辨識結果.....	32
表 4.1 系統的所有狀態.....	35
表 4.2 顧客類型的組合.....	38
表 4.3 實驗的顧客類型.....	38
表 4.4 學習(3000 回合)後期回饋平均值表.....	49
表 4.5 顧客類型的組.....	49
表 5.1 辨識率比較.....	57

# 第一章緒論

## 1.1 研究動機

在人工智慧的應用上，有很多理念都由於電腦硬體逐漸成熟，而慢慢的被實用化，語音辨識的技術就是一例。也有很多人工智慧的理念來自於科幻小說，像電腦以語音方式與人類交談、電腦會學習、電腦有感情、...。事實上，這些也是人工智慧領域努力的目標之一。如果將來語音辨識的辨識率夠高的話，能夠帶來很多便利性。例如：對於一般非專業人士，尤其對高齡使用者、小朋友或是不方便使用鍵盤滑鼠等輸入裝置的使用者來說，用說話的方式來與電腦溝通無疑是大大增加人與機器之間的友善度，減少一般使用者在使用電腦上所產生的障礙。對於聽障者來說，藉由電腦語音辨識，將別人的話轉換成文字，便能夠幫助聽障者瞭解別人的想法。

人工智慧中一個子領域「機器學習」的應用也非常廣泛。在遊戲中，電腦可以扮演對立的角色，例如：下棋。在電腦輔助教學方面，電腦可以做數學定理的證明、邏輯的推演。讓「機器學習」的方法有很多，例如：「強化學習」、「類神經網路」...等等。

本論文的目的有兩個，一個是要讓電腦可以聽懂人的聲音，另外是電腦能自己學得對話策略。透過語音辨識元件，將人的聲音轉成電腦能理解的文字，再用電腦自己所學得的對話策略做適當的回應。

電腦在以前我們的語音辨識元件（陳智聰，民國 91 年，語音辨識與語言模型，國立雲林科技大學電子與資訊工程研究所碩士論文）是線性預測編碼加上隱藏式馬克夫模型（Hidden Markov Models, HMMs）(Rabiner, 1989)的計算方法。線性預測編碼是用來作為訊號處理的問題；而隱藏式馬克夫模型是用來決定測試語音與訓練樣本的相似度。但是我們發現在辨識過程中有蠻多問題的，在本論文第三章 3.1 節展示一個實驗，其結果說明梅爾頻譜分析（Mel Frequency Cepstrum Coefficients, MFCCs）(Becchetti & Ricotti, 1999) 能夠解決以往的一些訊號處理的問題，因此我們這次用梅爾頻譜分析加上隱藏式馬克夫模型去做語音辨識。

除了語音辨識之外，本論文也會介紹如何訓練電腦，使它由經驗中知道哪種對話策略是比較好的。我們的對話場景是顧客到咖啡廳點咖啡，電腦扮演店員的角色。我們會使用「強化學習」( Sutton & Barot, 1998 ) 預先訓練店員，然後再

用訓練後的店員跟使用者對話。

本研究主要分成兩個部分，一個是語音辨識，另外一個是對話策略。在語音辨識方面，系統先用訊號處理的方法去獲得語音特徵值。然後以隱藏式馬可夫模型來決定語音特性的相似度。在對話策略的學習問題上，系統是以強化學習中的 Sarsa 演算法為核心。圖 1.1 是系統方塊圖。

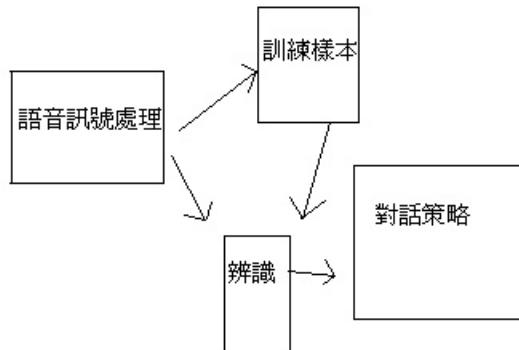


圖 1.1 系統方塊圖

## 1.2 研究方法

在語音辨識中，我們先研究聲波的特徵，而在之前我們用線性預測編碼遇到了兩個主要問題：語音切割、雜訊問題。然後我們發覺，解決這些問題方法中以梅爾頻譜分析的效果比較好。

在對話策略的研究中，強化學習一直是我們的研究方向，然而我們曾猶豫是要用哪一種獎勵方式才較為合理。在 NJFun ( Singh, Litman, Kearns & Walker, 2002 ) 對話系統中，它的獎勵方式是由使用者在對話結束後才給定的。它的訓練方式是將 NJFun 對話系統放置在網路上，讓大家來訓練它。但是我們採用店員每講一句話，就給一個獎勵的方式。而對話策略訓練是用模擬的方式去進行，就是令電腦再扮演顧客的角色，如此一來訓練速度會加快。

## 1.3 論文架構

本論文的第一章是緒論。在這一章我們闡述研究目的與動機，簡述研究的方式。第二章是文獻探討。闡述過去語音的辨識方法與運用強化學習去訓練對話策略的例子。第三章是語音辨識。先介紹語音訊號辨識架構與理論基礎，比較並說明線性預測編碼與梅爾頻譜分析的優劣，再介紹用梅爾頻譜分析取得聲音特徵參數的方法、向量量化 ( Vector Quantization, VQ ) ( Lai, and Lue, 1996 ) 壓縮訊號 HMM 聲音模型的建立、辨識方式，和實驗結果。第四章是強化學習運用於對話

策略理論基礎。在這一章，我們說明如何將強化學習運用於對話策略。我們會以賣咖啡的店員為例子，詳細的說明電腦如何學習。其中包括強化學習概論、Sarsa演算法、對話策略訓練，和實驗結果。第五章是結論，說明如何將語音辨識與對話策略結合並且說明未來展望。



## 第二章 文獻探討

### 2.1 語音辨識的技術

語音辨識是一個很迷人的計算問題。從以前就有很多人從事這方面的研究，一直到這幾年，也有一些大公司（例如，IBM、Microsoft）推出商業化的產品了。雖然是已經商業化了，但是還是有不足的地方。例如，有些人發音較為不標準以至於電腦會誤判。而且人們使用這些產品時，其態度都要正經八百才能辨識正確。我們希望語音辨識的技術能夠做到像人們平常在聊天那樣子，輕輕鬆鬆就能讓電腦知道使用者的意思。

陳智聰的碩士論文「語音辨識與語言模型」(2002)的目標是要用語言模型來提高語音辨識的辨識率。理由是：「因為語音辨識只靠聲音訊號與聲音參考樣本做比對這樣是不夠的，除了做聲音訊號的辨識外，還加入了語言模型來幫助辨識，當輸入連續字音時，沒有語言模型的輔助，所辨識出的句子可能會不合語法，使用語言模型可以組合出更加合理的句子」（節錄於該論文）。

我們不可否認上述的想法，是非常好的做法。該論文也舉了三個語言模型：context-free grammar、bigram model、HMM。並且也展示四個實驗：分別是：一、沒有任何文法輔助，完全由辨識系統判斷。二、context-free grammar。三、bigram model。四、HMM。如下表：

表 2.1 語音辨識與語言模型：各種文法比較之結果

訓練指令數/ 測試指令數	實驗一 沒有任何文 法輔助	實驗二 CFG	實驗三 Bigram	實驗四 HMM
150/250	68%	96%	92.4%	88%

很顯然的，沒有任何文法輔助，其辨識率遠小於有語言模型輔助。該論文分成兩個部分：第一、語音辨識，第二、語言模型。由上面的結論得知語音辨識若沒有語言模型輔助的話，那麼辨識率就很差。也就是說，語音辨識還有改善的空間。

我們檢視了該論文的「語音辨識」的部分，其中語音訊號的分割、和取得特徵值這兩個部分做得不理想。因為我們發現該論文在處理語音訊號時，並沒有濾除雜訊。在真實的對話環境中充滿著噪音，但人類可以容忍這些噪音，並且將它

排除。不過以電腦的觀點來看，這些噪音會使得電腦非常敏感。

該論文在訊號處理流程如下：

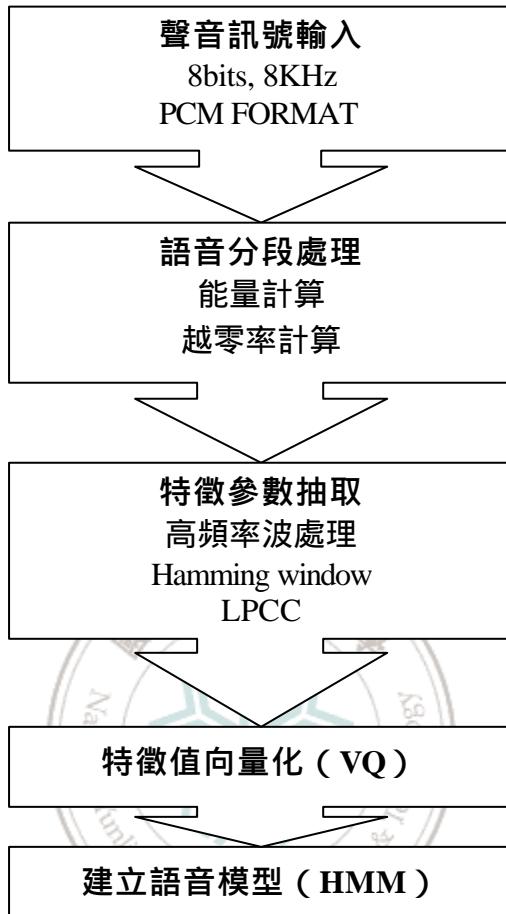


圖 2.1 語音辨識與語言模型：語音訊號處理之流程

「語音分段處理」的工作是用短時距能量測量 (Short-time Energy) 和越零率 (Zero-Crossing Rate) 來判斷分割點，用處理線性預測係數 (Linear Predictive Cepstrum Coefficients, LPCC)來取得語音特徵。

在短時距能量測量的技術中，因為這個方法要判斷某一短時距內能量變化的情形，作為區分有聲語音與無聲語音之依據，它用此方法的原因是因為無聲語音的能量值將遠小於有聲語音之能量值。但這裡的問題是我們無法斷定「短時距」的時間是多少，還有無法明確定義「無聲語音能量值」的大小以及「有聲語音能量值」的大小與比例。

假設我們使用計算越零率來觀測在一個短暫時間內訊號密集程度，用此方式來判斷是否為無聲語音，若越零率高則表示此段語音為無聲語音。反之，越零率低表示有聲語音。我們並無法明確的指出越零率的大小標準來決定是否為無聲語

音或有聲語音。

一段語音如果沒有濾除雜訊（噪音），則電腦會將雜訊的部分也考慮在內，而做出錯誤的計算。對於電腦來說，雜訊也是一種訊息。舉例來說，如果有一個人在很安靜的房間跟另外一人對話，而房間內有台冷氣。當冷氣沒開時進行對話，兩人都會聽的很清楚對方在所表達。當冷氣開了時進行對話，雖然有噪音，但兩人還是都會聽懂對方在所表達。現在將人和人的對話，改成人和電腦的對話，那麼當冷氣沒開時，則不會電腦計算它所聽到的資訊，就可以正確計算。但是如果開了冷氣，則電腦聽到的資訊也包括了冷氣的噪音，就會計算錯誤。

在陳智聰的論文裡面，其中語音訊號的分割、和取得特徵值這兩個部分是做得不好。它的做法是用短時距能量測量 (Short-time Energy)和越零率 (Zero-Crossing Rate) 來判斷分割點，並用線性預測係數(Linear Predictive Coefficient Cepstra, LPCC) (Rabiner & Schafer 1978) 來取得語音特徵。在短時距能量測量的技術中，因為這個方法要判斷某一短時距內能量變化的情形，作為區分有聲語音與無聲語音之依據。它用此方法的原因是因為無聲語音的能量值將遠小於有聲語音之能量值。但這裡的問題是我們無法斷定「短時距」的時間是多少，還有無法明確定義「無聲語音能量值」的大小以及「有聲語音能量值」的大小與比例。

針對上述的問題我們找到了一個比較好的方法。在訊號處理的領域內，以時間軸為基底的訊號是可以轉換至以頻率軸為基底的訊號，我們將訊號轉換至頻率軸的話，就可以濾除掉不想要的訊號。我們只要設定頻率的範圍就可以解決上述的問題，我們運用梅爾頻譜分析的語音特徵擷取技術取代陳智聰的論文的語音特徵處理的部分。它是一種使用傅立葉轉換和濾波器的特徵擷取技術。

## 2.2 對話策略探討

人與人之間的對話，都有一條脈絡可循，這一條脈絡可視為對話策略( Dialog policy) 無論是敘述事情、針對一件事情爭吵或是討論。我們運用強化學習，讓電腦學習對話策略，用其經驗來與人對話。這是一個應用很廣泛的研究。例如：旅遊的諮詢對話系統、點餐對話系統、購物系統、...等等。而語音辨識所扮演的角色是一個輸入裝置，而這個輸入裝置是帶給人方便，因此本研究才將語音辨識與對話策略結合在一起的。

我們先介紹一個運用強化學習理論的對話系統-- The NJFun System ( Singh, Litman, Kearns & Walker, 2002 )。NJ Fun 是一個電話語音旅遊的諮詢對話系統，它主要的服務是要讓打電話進來的人，詢問紐澤西有什麼好玩的地方。它詢問使用者一些條件，然後根據條件來搜尋適合的活動給使用者做參考，根據使用者的回饋來調整它的對話策略。然而，因為它是用電話作為溝通工具，所以它與人的溝通是透過語音辨識來將使用者的語音轉譯成文字，以及透過語音合成元件來將電腦資訊告訴使用者。其目標是經過很多人使用這套系統後，它會被調整成大部分的人都滿意的對話策略。

這套旅遊諮詢對話系統，是以活動、地點，時間做為查詢資料庫的依據。對話系統是透過語音辨識，將使用者的語音轉譯成文字告訴電腦。但辨識的結果有時會不精確，因此系統會有一個確認的機制，這個確認的機制會影響其對話策略。

NJFun system 會用語音合成技術詢問使用者，包括：(一) 使用者想要做活動形式 ( activity type )，例如：室內、室外。(二) 使用者想去的地點 ( location )，例如：博物館、酒場。(三) 使用者休閒的時間 ( time of day )：早上、下午、晚上。圖 2.2 是一則對話範例。

S: Welcome to NJFun. How may I help you?  
U: I'd like to find um winetasting in Lambertville in the morning.  
(ASR: I'd like to find out wineries the in the Lambertville in the morning.)  
S: Did you say you are interested in Lambertville?  
U: Yes.  
S: Did you say you want to go in the morning?  
U: Yes.  
S: I found a winery near Lambertville that is open in the morning.  
It is the Poor Richard's Winery in Lambertville.  
Thank you for using the system.  
Please give me feedback by saying 'good' 'so-so' or 'bad'.  
U: Good.  
S: Goodbye and have a nice day!  
**U:使用者      S: NJFun system**

圖 2.2 NJFun system 的對話範例

當系統跟使用者打過招呼後，使用者表達他希望去早上去 Lambertville 的一個品酒的地方。但是電腦把 'winetasting' 聽成 'wineries'。這時候電腦的自動語

音辨識系統（Automatic Speech Recognition, ASR）會告訴電腦辨識的可靠度在標準底下，電腦便會根據對話策略做一些確認的動作。最後，電腦查詢資料庫後，告訴使用者一些資訊後，也會問使用者對於該系統的對話策略評比：好、馬馬虎虎或不好，以作為修正對話策略的參考。

NJFun 這套系統可以學習對話策略。它主要的學習目的是它詢問使用者的方式。例如：它可以每一次都對於它聽到的關鍵詞（時間：早上、中午。地方：Lambertville。活動名稱：winetasting。）做一些確認的動作，也可以不用對每個項目都做確認。而決定要跟使用者確認與否就要看使用者的回饋情況。說的更明白，如果大部分的使用者都不喜歡確認的動作，那麼圖 2.2 的對話就會變成圖 2.3 的對話。原因是因為使用者若跟該系統對話，如果該系統產生圖 2.2 的對話策略，那使用者到最後會給 bad 的回饋，迫使系統修正對話策略，經過幾次了之後，系統就會將對話策略修改成圖 2.3。

```
S: Welcome to NJFun. How may I help you?  
U: I'd like to find um winetasting in Lambertville in the morning.  
(ASR: I'd like to find out wineries the in the Lambertville in the morning.)  
S: I found a winery near Lambertville that is open in the morning.  
It is the Poor Richard's Winery in Lambertville.  
...
```

圖 2.3 不做確認動作的對話範例

NJFun 可以選擇以下三種的任一方式去跟使用者對話：系統主導，使用者主導以及混合前面兩種方式。系統主導是系統告訴使用者它的服務項目，讓使用者選擇。例如：'Did you say you are interested in going to <activity>?'、'Welcome to NJFun. Please say an activity name or say a list of activities for a list of activities I know about.'使用者主導的方式，是系統不需要告訴使用者它的服務項目。像 'Welcome to NJFun. How may I help you?'。混合方式是系統主導和使用者主導的混合。也就是說，系統告訴使用者它的服務項目。使用者可以回答該問題，也可以不回答。例如：'Please tell me the activity type. You also can tell me the location and time.'

NJFun 系統狀態（Choice-State）由 7 個屬性組成，而 7 個屬性標示系統與使用者目前的對話情況，例如：打過招呼。底下表 2.2 則是 7 個屬性的列表。  
'Greet'：系統是否已對使用者打過招呼(0 no, 1 yes)。'Attribute'：系統正在問什麼(1=activity, 2=location, 3=time, 4=完成)。'Confidence/ Confirmed'：ASR 的信心度

0=low, 1=medium, 2=high , 3=要 ASR 再聽一次使用者說什麼 , 4=不用再問了(0~2 是一組; 3,4 是一組)。‘value’：是否得到了使用者給的條件資訊，例如：時間是早上，地點是 Lambertville ，...，等等(0 no, 1 yes)。‘Tries’：使用者給的條件資訊已經問了幾次(0 ~2)。'Grammar': 用哪一種 Grammar(限定或非限定)。限定指的是系統所問的問題是明確的（有選項的），使用者要針對該問題回答。

例如：

S: Did you say you want to go in the morning?

U: Yes.

非限定指的是系統所問的問題比較廣。例如：

S: Welcome to NJFun. How may I help you?

U: I'd like to find um winetasting in Lambertville in the morning.

'History': 之前的條件資訊是否有些是得不到的。有時候自動語音辨識器和使用者無法配合，以至於重複問兩次都失敗，因此系統就會在搜尋資料庫的時候，把所有的條件通通找出來。所以資料量會比較多。

表 2.2 NJFun 系統狀態的 7 個屬性

Feature	Values	Explanation
Greet (G)	0,1	Whether the system has greeted the user
Attribute (A)	1,2,3,4	Which attribute is being worked on
Confidence/Confirmed (C)	0,1,2,3,4	0,1,2 for low, medium, and high ASR confidence. 3,4 for explicitly confirmed, and disconfirmed
Value (V)	0,1	Whether value has been obtained for current attribute
Tries (T)	0,1,2	How many times current attribute has been asked
Grammar (M)	0,1	Whether non-restrictive or restrictive grammar was used
History (H)	0,1	Whether there was trouble on any previous attribute

我們 NJFun 系統在每個狀態都會有兩個對話動作 ( Choice-Action )。對話動作是決定該狀態要說什麼話。表 2.3 是該系統的所有狀態與對應對話動作的對應表中，由表 2.3 中我們可以發現系統主導的詢問方式有比較多的確認動作，而使用者主導時都比較少確認動作。NJFun 的對話策略就是要從很多使用者身上學習，將這些對話動作融合，而調整成為一個比較符合大眾的對話策略。為了達到這個目標，這些對話動作都是不會違反常理。例如：打過招呼了又打一次招呼。

本論文訓練對話策略的目的，是要從一些合理的與不合理的選擇中選出合理的。因為有的時候，我們的對話動作有很多，而我們還要一個一個的對狀態作分類，有的時候很多狀態，那麼就要考慮很多。所以我們希望用「強化學習」讓電腦學到避免重複的對話。

表 2.3 NJFun 系統所有狀態與對話動作的對應表

Choice-States							Action Choices	Action	Prompt
G	A	C	V	T	M	H			
0	1	0	0	0	0	0	GreetS,GreetU		Welcome to NJFun. Please say an activity name or say 'list activities' for a list of activities I know about.
1	1	0	0	1	0	0	ReAsk1S,ReAsk1M		Welcome to NJFun. How may I help you?
1	1	0	1	0	0	0	NoConf,ExpConf1		
1	1	0	1	0	1	0	NoConf,ExpConf1		
1	1	1	1	0	0	0	NoConf,ExpConf1		
1	1	1	1	0	1	0	NoConf,ExpConf1		
1	1	2	1	0	0	0	NoConf,ExpConf1		
1	1	2	1	0	1	0	NoConf,ExpConf1		
1	1	4	0	0	0	0	ReAsk1S,ReAsk1M		
1	1	4	0	1	0	0	ReAsk1S,ReAsk1M		
1	2	0	0	0	0	0	Ask2S,Ask2U		I know about amusement parks, aquariums, cruises, historic sites, museums, parks, theaters, wineries and zoos. Please say an activity name from this list.
1	2	0	0	0	0	1	Ask2S,Ask2U		
1	2	0	0	1	0	0	ReAsk2S,ReAsk2M		
1	2	0	0	1	0	1	ReAsk2S,ReAsk2M		Please tell me the activity type. You can also tell me the location and time.
1	2	0	1	0	0	0	NoConf,ExpConf2		
1	2	0	1	0	0	1	NoConf,ExpConf2		
1	2	0	1	0	1	0	NoConf,ExpConf2		
1	2	0	1	0	1	1	NoConf,ExpConf2		
1	2	1	1	0	0	0	NoConf,ExpConf2		
1	2	1	1	0	0	1	NoConf,ExpConf2		
1	2	1	1	0	1	0	NoConf,ExpConf2		
1	2	1	1	0	1	1	NoConf,ExpConf2		
1	2	2	1	0	0	0	NoConf,ExpConf2		
1	2	2	1	0	0	1	NoConf,ExpConf2		
1	2	2	1	0	1	0	NoConf,ExpConf2		
1	2	2	1	0	1	1	NoConf,ExpConf2		
1	2	2	1	0	1	1	NoConf,ExpConf2		
1	2	4	0	0	0	0	ReAsk2S,ReAsk2M		
1	2	4	0	0	0	1	ReAsk2S,ReAsk2M		
1	2	4	0	1	0	0	ReAsk2S,ReAsk2M		
1	2	4	0	1	0	1	ReAsk2S,ReAsk2M		
1	3	0	1	0	0	0	NoConf,ExpConf3		Did you say you are interested in going to < activity >?
1	3	0	1	0	0	1	NoConf,ExpConf3		-
1	3	0	1	0	1	0	NoConf,ExpConf3		
1	3	0	1	0	1	1	NoConf,ExpConf3		
1	3	1	1	0	0	0	NoConf,ExpConf3		Did you say you are interested in < location > ?
1	3	1	1	0	0	1	NoConf,ExpConf3		-
1	3	1	1	0	1	0	NoConf,ExpConf3		
1	3	1	1	0	1	1	NoConf,ExpConf3		
1	3	2	1	0	0	0	NoConf,ExpConf3		
1	3	2	1	0	0	1	NoConf,ExpConf3		
1	3	2	1	0	1	0	NoConf,ExpConf3		
1	3	2	1	0	1	1	NoConf,ExpConf3		

(a)

(b)

## 第三章 語音辨識

### 3.1 研究方法

在語音訊號分析辨認程式中，先以數位訊號處理技術將訊號樣本做必要的切割，再進行樣本的訓練或辨認的工作，如圖 3.1。在這裡的語音辨識，是以詞的辨識為主，因為在完成語音辨識後，我們將其辨識結果交給對話策略，如圖 3.2，而對話策略的輸入是以關鍵詞為元素。

由於語音的資料量太大了，我們不可能將原始語音全部儲存下來當作標準的語音參考樣本，而必須針對聲音訊號的特性，來取得適當的特徵參數來進行聲紋比對辨識。特徵參數若取得好、穩定，則有助於辨認的效果，所以擷取特徵參數在一個辨識系統中是相當重要的一環。



圖 3.1 語音辨識系統方塊圖

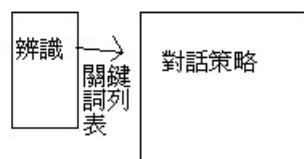


圖 3.2 辨識結果給對話策略

在之前「語音辨識語言模型」，我們主要運用線性預測係數 (Linear Predictive Cepstral Coefficients, LPCC)來抓取語音訊號的特徵值。由於 LPCC 是一種語音訊號的壓縮技術，並不適合當作語音訊號的特徵值擷取技術。我們找到一個更符合

我們需要的擷取特徵值技術 MFCC ( Mel-Frequency Cepstral Coefficients ) 圖 3.3 是我們對於 LPCC 和 MFCC 的比較實驗。實驗內容是 LPCC 和 MFCC 對於一樣的聲音波形所轉換的特徵值。音框長度是 256 個取樣值。我們分別用 LPCC 和 MFCC 轉換出來的特徵參數顯示於圖 3.3(a) 及圖 3.3(b) 底下的數據。很明顯的，在 LPCC ( 圖 3.3(a) ) 方面，因為原始聲音訊號中有少許的雜訊，所以會將其雜訊也做編碼的動作，因此無法觀測真正的聲音是在那一段波形。用 MFCC ( 圖 3.3(b) ) 所轉換出來的特徵參數可以分出那一段是真正的聲音，那一段是雜訊。

因此，實驗結果告訴我們：LPCC 的抗雜訊能力很低，幾乎連環境的雜訊都被誤認為是使用者發出的聲音。當雜訊被認為是使用者的聲音時，會影響我們的對語音的切割。在這裡做語音切割的目的是要將雜訊屏除。當我們將語音切割錯誤時，我們就會送錯誤的特徵值傳給辨識器。雖然辨識器有容錯的機制，但是如果錯誤率太高的話，也會影響其辨識結果，所以雜訊也會影響我們的辨識結果。因此我們必須將不必要的雜訊捨棄。

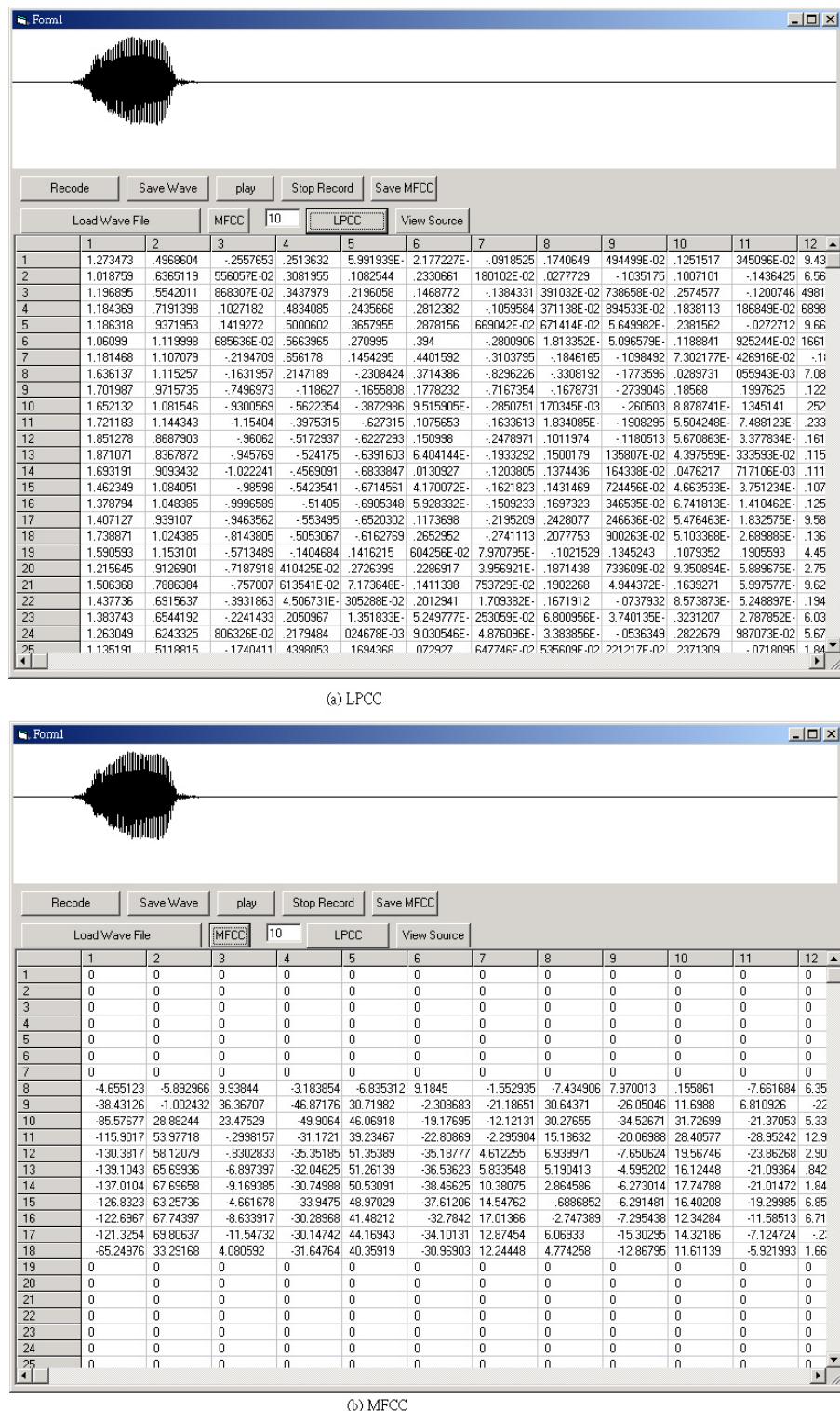
如果用 LPCC 技術來擷取特徵值，必須要以使用較高級、抗雜訊的錄音設備來擷取聲音，還要必須配合語音分段處理 ( 短時距能量計算、越零率計算 )，才能得到我們想要的特徵值。MFCC 的抗雜訊能力比 LPCC 高，並且有效的解決 LPCC 會將環境的雜訊被誤認為是使用者發出的聲音的問題，也不需用短時距能量計算、越零率計算來處理語音分段。因為 MFCC 可以處理這些問題，所以我們採用 MFCC 將環境的雜訊濾除。

擷取語音特徵參數值後，我們使用隱藏式馬可夫模型(Hidden Markov Model, HMM)，辨識語音訊號特徵的相似度。隱藏式馬可夫模型是加了機率處理的有限狀態機，使用了雙重隨機程式，一個是狀態轉移機率，另一是狀態觀測機率。隱藏式馬可夫模型依據語音音素的特性，將一連串音框的特徵向量分割成 n 個狀態，使具有類似性質的音框相結合，也就是說每個音都會用一個含有 n 個狀態的轉移網路 的模型來代表。

在我們使用 HMM 之前，必須將每個音框的特徵向量視為一個符號，但是要用什們樣的機制來將如何將特徵向量兌換成一個符號？我們希望它能夠被當成訓練和辨識的特徵向量基本樣本，如此一來無論在訓練或辨識的時候，都能公平的對應到一個觀測符號，簡單來說，我們需要一個能夠將特徵向量分類的機制。因此我們使用向量量化 ( Vector Quantification ) 來作為觀測符號產生的機制。

我們將聲音訓練樣本經由訊號處理，轉成語音特徵，再經由 VQ 轉換成觀測符號的序列，然後將每個觀測符號序列變成一個 HMM 模型，所以每一個聲音都有一個 HMM 模型。我們辨識聲音的做法是將語音測試樣本轉成語音特徵值，然

後再經由 VQ 的編碼簿轉換成觀測符號序列，再計算各聲音模型產生觀測符號序列的機率，選出最接近的模型。計算一個聲音模型產生觀測符號序列的機率方法，一般都用維特比（Viterbi）演算法。



(a) LPCC

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	-4.655123	-5.892966	9.93844	-3.183854	-6.835312	9.1845	-1.552935	-.7.434906	9.790013	.155861	-.7.661684	6.35
9	-38.43126	-1.002432	36.36707	-46.87176	30.71982	-2.308683	-21.18651	30.64371	-26.05046	11.65988	6.810926	-.22
10	-85.57677	28.88244	23.47529	-49.9064	46.06918	-19.17695	-12.12131	30.27655	-34.52671	31.72639	-21.37053	5.33
11	-115.9017	53.97718	-.2998157	-31.1721	39.23467	-22.80869	-.2.395904	15.18632	-20.06988	28.40577	-28.95242	12.9
12	-130.3817	58.12079	-.8302833	-35.35185	51.35389	-35.18777	4.612255	6.939971	7.650624	19.56746	-23.86268	2.90
13	-139.1043	65.69936	-6.897397	-32.04625	51.26139	-36.53623	5.833548	5.190413	-4.595202	16.12448	-21.09364	.842
14	-137.0104	67.69568	-9.169385	-30.74988	50.53039	-38.46562	10.38075	2.664569	-6.273014	17.74788	-21.01472	1.84
15	-126.8323	63.25736	-4.6561678	-33.9475	48.97029	-37.61208	14.54762	-6.6886852	-6.291481	16.40208	-19.29985	6.85
16	-122.6967	67.74397	-8.633917	-30.28968	41.48212	-32.7842	17.01366	-2.747389	-7.295438	12.34284	-11.58513	6.71
17	-121.3254	69.80637	-11.54732	-44.16943	-34.10131	12.87454	6.06933	-15.30295	14.32186	-7.124724	-.2	
18	-65.24976	33.29188	4.080592	-31.64764	40.35919	30.96903	12.24448	4.774258	-12.86795	11.61139	-5.921993	1.66
19	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0

(b) MFCC

圖 3.3 LPCC 技術與 MFCC 技術的比較

### 3.2 聲音特徵值之計算

我們由前面的 LPCC 與 MFCC 實驗結果可知，LPCC 的抗雜訊能力比 MFCC 還低，因此我們以 MFCC 取代原本的 LPCC 語音特徵值計算。

整個 MFCC 的訊號特徵計算的流程如圖 3.4

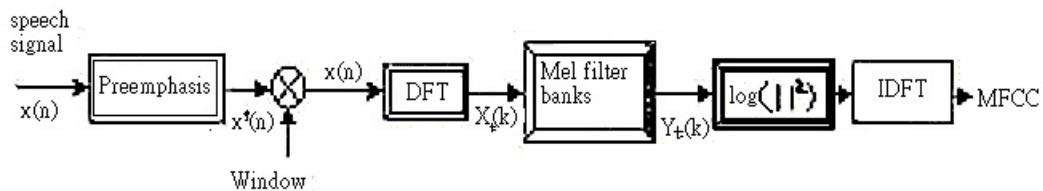


圖 3.4 訊號處理

#### 3.2.1 高頻訊號之補強

聲音信號進來了之後，要先做高通濾波，因為聲音從嘴唇發出來的時候，會有高頻損失，所以我們必須還原高頻訊號。由訊號處理得知，如果要將某個頻帶的訊號凸顯出來，我們以用濾波器處理該訊號。

但為了聲音音量一致性，避免因為聲音的音量影響辨識率，我們在聲音樣本(包含訓練和測試)進入濾波器之前，要先將樣本調高至最大聲量而不失真。

高通濾波器的作法是：

$$x'(n) = x(n) - 0.95x(n-1) \quad .[\text{式 3. 1}]$$

$x'(n)$  是濾波器的輸出。

$x(n)$  是濾波器的輸入。

#### 3.2.2 漢明視窗

在語音辨識中，常用漢明視窗（Hamming Window）來作為音框處理，以增加音框左端和右端的連續性。因為如果我們直接切割作為音框處理的話，音框之後就會不連續。

漢明視窗做法如下：

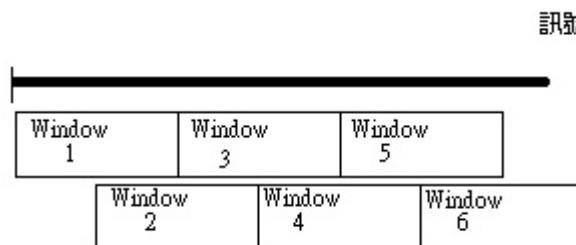
$$W(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), n = 0 \dots N-1 \quad [式 3. 2]$$

N 是音框長度。

本研究是將 N 設定為 256。我們的原始資料序列  $x'(n)$  要乘上漢明視窗成為  $X_h(i)$  如下：

$$X_h(i) = x'(i) * W(i), i = 0, \dots, N-1 \quad [式 3. 3]$$

我們為了保留語音訊號的相關性，所以用交叉切割的方式來切割出漢明視窗。然後送 FFT 做頻普分析。如圖 3.5：



一個音框長度是 256 個資料點，重疊部分是 128 個資料點。

### 3.2.3 離散傅立葉轉換

我們的目的是要擷取語音的特徵值，然而若將一段波形直接拿來做分析的話，是非常困難的事情。或許用眼睛能看出一點規則，但是這並不是解決問題的根本之道。因此我們才求助於另外一種方法—波形成分分析--，也就是說一段波形是由哪些基礎波形組成的。而最常用的方法是傅立葉轉換。傅立葉轉換是將一段波形從時間域轉至頻率域，以正弦波為基波，如圖 3.6。原始波形圖 (a) 經由傅立葉轉換得到圖(b)。圖(b)的自變數是正弦波的頻率，圖(a)的是時間。圖(b)的應變數是正弦波的振幅大小，圖(a)的是該時間點的振幅大小。

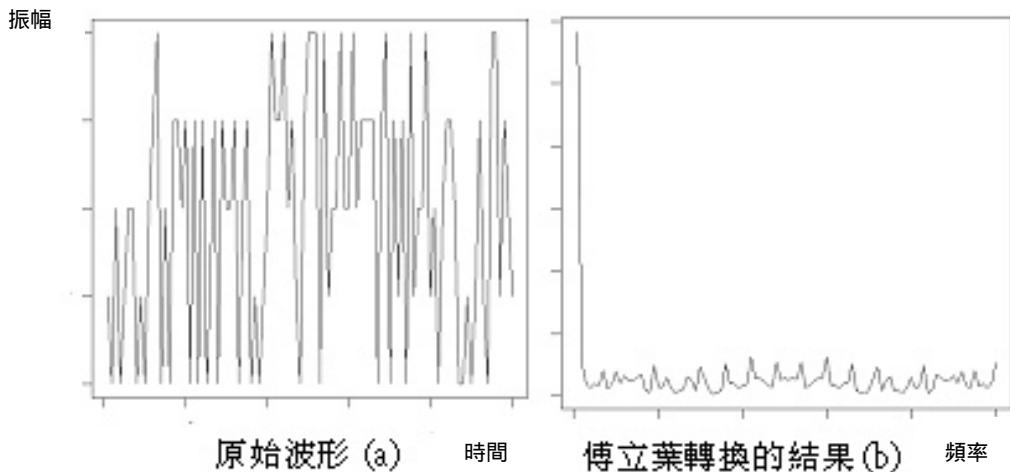


圖 3.6 傳立葉轉換

但是由於我們的語音是以離散的形式儲存於電腦，所以就使用離散傅立葉轉換(Discrete Fourier Transform; DFT)，離散傅立葉轉換的定義如下：

$$X_t(k) = \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi kn}{N}} \quad [\text{式 3. 4}]$$

N 是序列長度，x(n)是序列的內容。

爾後，將經由漢明視窗處理過的音框  $X_h(i)$  視為離散傅立葉的原始資料 x(n) 代入。

### 3.2.4 濾波器處理

傅立葉轉換分析幫我們把聲音所有頻率的特徵分析出來。但是有些特徵我們需要、有些卻不用，我們所要的特徵是人耳所聽得到的特徵，所以我們需要一組濾波器，來幫我們找出人耳所聽得到的特徵。梅爾濾波器組 (Mel filter bank) 正是我們需要的濾波器，因為它的頻率響應跟人耳是類似的。這種濾波器的主要特性是，在頻率為 1kHz 以下時，其增益/頻率比是線性的；在頻率為 1kHz 以上是對數型。它的計算方法如下：

$$Y_t(m) = \sum_{k=b_m-\Delta_m}^{b_m+\Delta_m} X_t(k) U_{\Delta_m}(k + b_m) \quad [\text{式 3. 5}]$$

$$b_m = b_{m-1} + \Delta_m \quad [\text{式 3. 6}]$$

k 是傅立葉轉換的頻率；  $X_t(k)$  傳立葉轉換結果；  $b_m$  是中心頻率；

$U_{\Delta_m}(k + b_m)$  是 triangular window :

$$U_{\Delta_m}(k) = \begin{cases} |k| < \Delta_m \rightarrow 1 - k / \Delta_m \\ |k| \geq \Delta_m \rightarrow 0 \end{cases} \dots \dots \dots \text{[式 3. 7]}$$

$\Delta_m$  是一個頻帶寬度。在 1kHz 以下， $\Delta_m = 1\text{kHz}/10$ ，在 1kHz 以上， $\Delta_m = 1.2 \times \Delta_{m-1}$ 。

### 3.2.5 對數能量運算及梅爾倒頻譜計算

前面的訊號處理已經將一段語音訊號轉換成耳朵所聽到得的特徵。至於 MFCC 的計算，我們是用反傅立葉轉換求得。而在這裡我們將梅爾濾波器組的輸出  $Y_t(m)$  取對數，因為取對數之後，計算上比較簡單。為了避免複雜的計算，我們用離散餘弦轉換 (Discrete Cosine Transform (DCT))：

$$y_t(k) = \sum_{m=1}^M \log\{|Y_t(m)|} \cos\left(k\left(m - \frac{1}{2}\right)\frac{p}{M}\right), k = 0, \dots, 12 \dots \dots \dots \text{[式 3. 8]}$$

M 是濾波器的個數。

k 是特徵向量的階數。

### 3.3 訓練語音模型

當語音訊號被分音框與被取特徵值 (MFCC) 之後，每個音框的特徵向量被視為一個符號，因此每一段語音訊號皆可看成一連串的符號序列。將每個音框的特徵向量視為一個觀測符號，那麼就要有一個共通的編碼簿來將特徵向量兌換成一個符號，如圖 3.7。

語音處理通常使用了向量量化(VQ)的技術處理語音特徵參數。VQ 是於 1984 年由 Gray 提出並應用於資料壓縮。VQ 可以降低傳輸位元率，以便增加通訊通道之容量和儲存記憶體，是一種有損失的區塊編碼壓縮法，經過壓縮編碼後無法完整恢復到原始資料。除了壓縮資訊外，VQ 亦可做到分類的效果，所以我們用它來分類 MFCC 的語音參數。

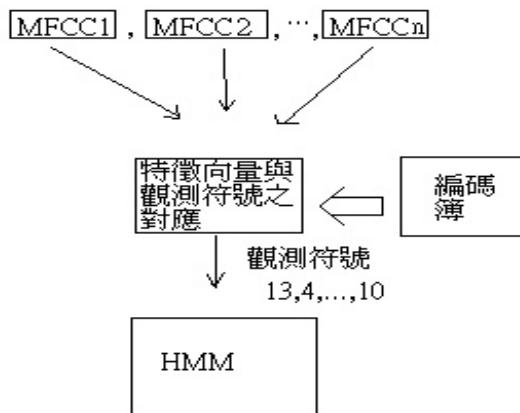


圖 3.7 特徵向量兌換觀測符號示意圖

VQ 是將在  $k$  維的向量空間  $R^k$  中，分成有限個區域 (Voronoi regions) 如下圖 3.8，每一個區域以一個向量  $y_i$  來代表屬於此區域的所有向量，簡單的說，VQ 是從多個來源的向量樣本中找出幾個代表向量。這些代表向量分別代表區域，以後若有測試樣本落在某代表區域的話，就用該代表向量來表示該測試樣本。這也就是將向量分類，每一個  $y_i$  稱為代碼 (code vector 或是 codeword)，而在向量空間  $R^k$  中所有的 codeword 的集合  $Y = \{y_i | i = 1, 2, \dots, N\}$  稱為編碼簿(codebook)， $N$  表示編碼簿大小 (codebook size)，Voronoi regions 定義為：

$$V_i = \{x \in R^k : \|x - y_i\| \leq \|x - y_j\|, \text{for all } j \neq i\} \dots \dots \dots \text{[式 3. 9]}$$

$$\bigcup_{i=1}^N V_i = R^k \dots \dots \dots \text{[式 3. 10]}$$

$$V_i \bigcap V_j = \emptyset \dots \dots \dots \text{[式 3. 11]}$$

$$\forall i \neq j$$

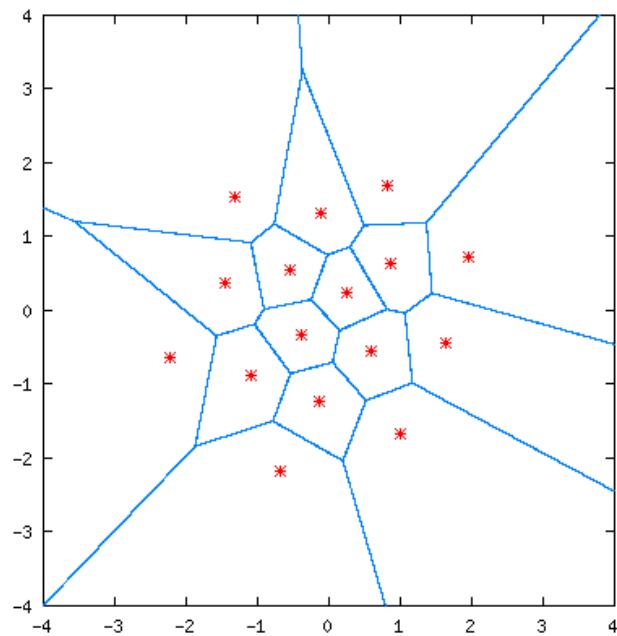


圖 3.8 Voronoi regions 示意圖

在 VQ 中，編碼簿生成的方式是根據 最小失真率 ( minimum distortion ) 的原則，決定如何做適當的分割以及如何在分割區域中找出一個最適合的代碼，即是求一個最小的平均失真。

本研究使用 Linde-Buzo-Gray(LBG)演算法(Lai & Lue, 1996)來求取編碼簿(編碼簿大小為 128)，演算法如下：

**Step 1:**初始編碼簿，求取中心向量。

$$c_1^* = \frac{\sum_{m=1}^M x_m}{M} \dots \text{[式 3.12]}$$

Codebook 內只有一個代碼:  $c_1^*$

計算所有來源向量  $x_m$  和中心點  $c_1^*$  平均距離

$$D_{one}^* = \frac{\sum_{m=1}^M \|x_m - c_1^*\|^2}{Mk} \dots \text{[式 3.13]}$$

M=來源向量個數，k=比重(Weight)， $D_{one}^*$ =距離平均， $x_m$ =來源向量

**Step 2 分裂 (Splitting) 編碼簿**

$$c_j^{(0)} = (1 + \epsilon) c_j^* \dots \dots \dots \text{[式 3. 14]}$$

$$c_{N+j}^{(0)} = (1 - \epsilon) c_j^* \dots \dots \dots \text{[式 3. 15]}$$

$N$  = 目前的編碼簿大小(會變動的),  $\epsilon$  = 一個很小的誤差值,  $1 \leq j \leq N$   
做完這個動作,  $N=N*2$  (擴大目前的編碼簿)。

**Step 3 重複(i th iteration, 迴圈第 i 次):** 這個部分是要修正編碼簿內的每個向量  
( $\bar{c}$ ). 使得每個  $\bar{c}$  在所對應 Position 內的所有向量為最短距離.

(i) 對於  $m=1,2,\dots,M$ , 用編碼簿找出最短來源向量距離  $\|x_m - c_n^{(i)}\|^2$ ,

$n=1,2,\dots,N$ 。依照最短距離, 將  $x_m$  對應到最短的  $c_n^{(i)}$ , 使得

$Q(x_m) = c_{n*}^{(i)}$ ,  $n*$  是  $x_m$  在編碼簿向量中最短距離的索引。

(ii) 對於  $n=1,2,\dots,N$ , 更新編碼簿:

$$c_n^{(i)} = \frac{\sum_{Q(x_m)=c_n^{(i)}} x_m}{\sum_{Q(x_m)=c_n^{(i)}} 1} \dots \dots \dots \text{[式 3. 16]}$$

(iii)  $i=i+1$

(iv) 計算那一類( $c_n^{(i)}$ )的平均距離。

$$D_{avg}^{(i)} = \frac{\sum_{m=1}^M \|x_m - Q(x_m)\|^2}{Mk} \dots \dots \dots \text{[式 3. 17]}$$

(v) 如果  $(D_{avg}^{(i-1)} - D_{avg}^{(i)}) / D_{avg}^{(i-1)} > \epsilon$  回到步驟(i), 否則到步驟(vi)。

(vi)  $D_{avg}^{(*)} = D_{avg}^{(i)}$ ,  $c_n^* = c_n^{(i)}$ ,  $n=1,2,\dots,N$ 。

重複 Step 3 and 4 直到編碼簿的大小被擴增到所要的數目。

### 3.3.1 HMM 設計之理論基礎

我們將欲訓練的語音樣本輸入經過重新估算 HMM 上的機率值，此即為訓練的程式。對每一個單字或單詞建立一個隱藏式馬可夫模型 I，此為每個單字語音

的參考模型，若有  $N$  個語音觀測序列即建立  $N$  個 HMM： $I_1, I_2, \dots, I_N$ 。

HMM 的主要元素有如圖 3.9 所示：

1.  $N$ ：模型狀態數目
2.  $M$ ：每一個狀態中之觀測符號個數，即 symbol 數目
3.  $p$ ：初始狀態機率分佈：

$p = \{p_i\}$  = probability that initial state is  $S_i$

$$p_i = P(q_1 = S_i), 1 \leq i \leq N \dots \dots \dots \text{[式 3. 18]}$$

$$\sum_i p_i = 1$$

$S_i$ ：第  $i$  個狀態

$q_t$ ：時間  $t$  的狀態

4. 狀態轉移機率矩陣 (State transition probabilities)：

$$A = \{a_{ij}\}$$

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq N \dots \dots \dots \text{[式 3. 19]}$$

5. 狀態  $j$  的觀測符號機率矩陣 (Observation symbol probability)：

$$B = \{b_j(k)\}$$

$$b_j(k) = P(v_k \text{ at time } t | q_t = S_j), 1 \leq j \leq N, 1 \leq k \leq M \dots \dots \dots \text{[式 3. 20]}$$

第  $k$  個觀測符號

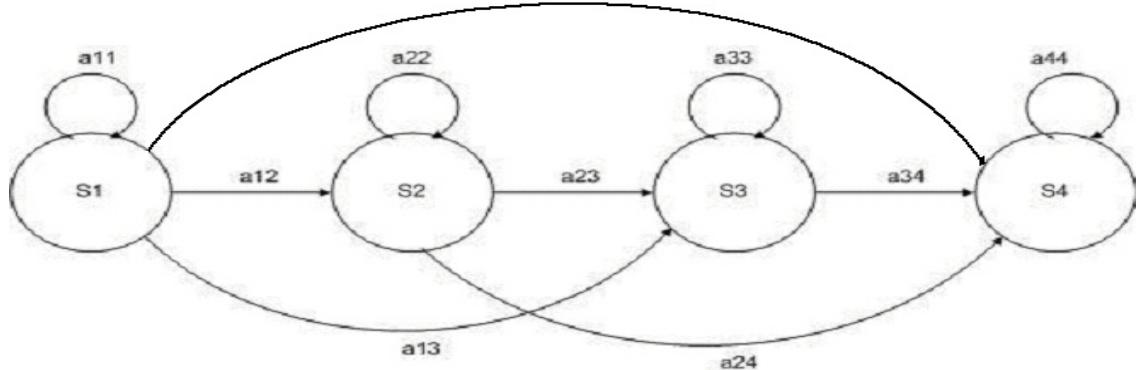


圖 3.9 本研究 HMM 之設計圖：使用 4 個狀態

在給定適當的  $N$ 、 $A$ 、 $B$  和  $p$  值，HMM 可視為觀測序列  $O = (o_1 o_2 \dots o_T)$  長度為  $T$  的產生器。因此，可計算出給定  $O$  後， $O$  的機率值， $P(O|?)$ 。應用在語音辨識時， $O$  是一序列的語音特徵向量 (codeword)，以下將說明如何有效計算  $P(O|?)$ ，如何調整模型參數  $?=(A, B, p)$  使得  $P(O|?)$  有最大值。

### 3.3.2 HMM 機率計算

給定一個隱藏式馬可夫模型，欲計算一個得到觀測序列  $O = O_1 O_2 \dots O_T$  的機率，即  $P(O|?)$ ，最直接的方式就是計算所有可能狀態序列的機率總和。考慮一個固定的狀態序列  $Q = q_1 q_2 \dots q_r$ ， $q_1$  是初始狀態。

其中，

$$P(O | Q, \mathbf{I} ) = \prod_{t=1}^T P(O_t | q_t, \mathbf{I} ) \\ = b_{q_1}(O_1) b_{q_2}(O_2) \cdots b_{q_T}(O_T) \quad \dots \dots \dots \text{[式 3. 22]}$$

$$P(O | I) = \sum_{q_1, q_2, \dots, q_T} b_{q_1}(O_1) b_{q_2}(O_2) \cdots b_{q_T}(O_T) p_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T} \\ = \sum_{q_1, q_2, \dots, q_T} p_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T) \quad \dots \text{式 3. 24}$$

若隱藏式馬可夫模型有  $N$  個狀態，且允許由某一狀態轉移到其他狀態，假設觀測序列中有  $T$  個觀測值，則所有狀態轉換序列數目為  $N^T$ ，需要  $O(2TN^T)$  的運算，當  $T$  值太大時，會使得機率運算太過耗時。因此，一般都使用正算-逆算程式 (Forward-Backward procedure) ( Rabiner, 1989 ) 來計算 HMM 的機率。

### 3.3.2.1 正算程式 (Forward procedure)

給定一個模型  $\pi$ , 定義正算變數 (forward variable)  $a_t(i)$  為部分觀測序列  $O_1, O_2, \dots, O_t$  在時間為  $t$  時, 系統狀態停留在狀態  $S_i$  的機率：

**演算法：**

### i. Initialization :

$$\mathbf{a}_1(i) = \mathbf{p}_i b_i(O_1), \quad 1 \leq i \leq N \dots \dots \dots \text{[式 3. 26]}$$

ii. Induction :

$$\mathbf{a}_{t+1}(j) = \left[ \sum_{i=1}^N \mathbf{a}_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1; \quad 1 \leq j \leq N \dots \text{[式 3. 27]}$$

iii. Termination :

$$P(O | I) = \sum_{i=1}^N \mathbf{a}_T(i) \dots \dots \dots \text{[式 3. 28]}$$

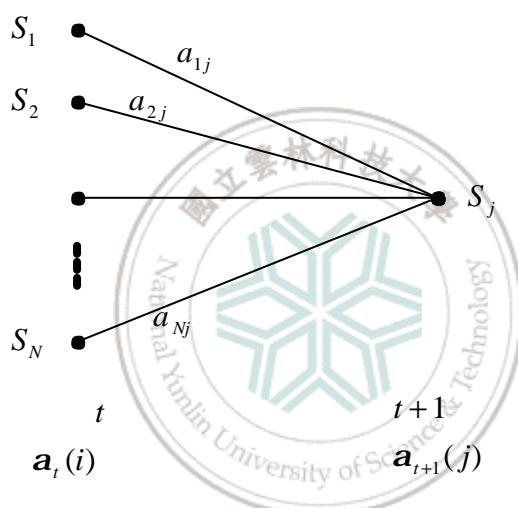


圖 3.10 正算程示意圖

### 3.3.2.2 逆算程式 (Backward Procedure)

逆算程式與正算程式類似，定義逆算變數  $b_t(i)$ ，給定模型？在時間  $t$  時狀態停留在狀態  $S_i$  時候，部分觀測序列從時間  $t+1$  到最後結束的機率值：

$$\mathbf{b}_t(i) = P(O_{t+1} O_{t+2} \cdots O_T | q_t = S_i, I) \dots \dots \dots \text{[式 3. 29]}$$

演算法：

i. Initialization

$$\mathbf{b}_T(i) = 1, \quad 1 \leq i \leq N \dots \dots \dots \text{[式 3. 30]}$$

### ii. Induction

$$\mathbf{b}_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \mathbf{b}_{t+1}(j), \quad t = T-1, T-2, \dots, 1, \quad 1 \leq i \leq N \dots [式 3.31]$$

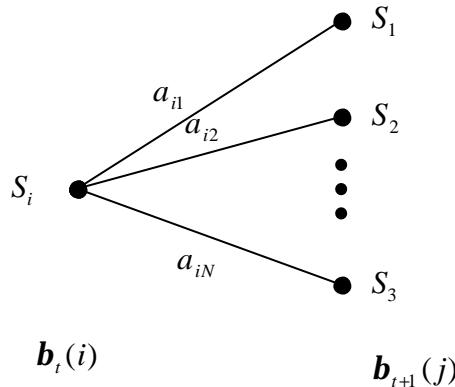


圖 3.11 逆算程式示意圖

### 3.3.2.3 Forward-Backward Recursion

由前面正算與逆算程式可知：

$$P(O | I) = \sum_{i=1}^N \mathbf{a}_T(i) \dots [式 3.32]$$

$$= \sum_{i=1}^N \mathbf{a}_t(i) \mathbf{b}_t(i) \dots [式 3.33]$$

$$= \sum_{i=1}^N \sum_{j=1}^N \mathbf{a}_t(i) a_{ij} b_j(O_{t+1}) \mathbf{b}_{t+1}(j) \dots [式 3.34]$$

這樣的演算法只需要  $O(N^2T)$  的計算時間，顯然比原來的方式有效率多了。

### 3.3.3 訓練 HMM 模型

給定一個模型  $\mathbf{I} = (A, B, \mathbf{p})$  之後，必須去修正 HMM 上的機率，使得屬於這個模型觀測序列其機率值為最大 (maximize  $P(O|?)$ )，即在語音辨識中，輸入參考樣本去修正每一個語音的 HMM，因為到目前並無任何解析方式去找出最佳的參數，只好使用疊代程式 (Iterative Procedure) 或是升降技術 (Gradient Techniques)。比較常用的是波氏 (Baum-Welch) 演算法。在給定一個模型  $\mathbf{I}$  與觀測序列  $O$ ，定義  $x_t(i, j)$  為從時間  $t$  到  $t+1$  時狀態由  $S_i$  轉到  $S_j$  的機率。

$$\begin{aligned} x_t(i, j) &= P(q_t = S_i, q_{t+1} = S_j | O, \mathbf{I}) \\ &= \frac{P(q_t = S_i, q_{t+1} = S_j, O | \mathbf{I})}{P(O | \mathbf{I})} \end{aligned} \quad [\text{式 3. 35}]$$

將上述 Forward-Backward procedure 中的  $\alpha$ ,  $\beta$  代入展開得到

$$\begin{aligned} x_t(i, j) &= \frac{\mathbf{a}_i a_{ij} b_j(O_{t+1}) \mathbf{b}_{t+1}(j)}{P(O | \mathbf{I})} \\ &= \frac{\mathbf{a}_i a_{ij} b_j(O_{t+1}) \mathbf{b}_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \mathbf{a}_i(i) a_{ij} b_j(O_{t+1}) \mathbf{b}_{t+1}(j)} \end{aligned} \quad [\text{式 3. 36}]$$

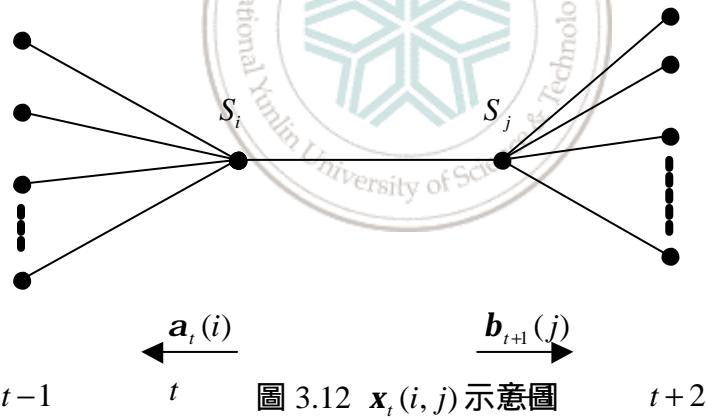


圖 3.12  $x_t(i, j)$  示意圖

定義  $g_t(i)$  為在  $t$  時間下，在狀態  $i$  的機率值。

$$g_t(i) = \sum_{j=1}^N x_t(i, j) = \frac{\mathbf{a}_t(i) \mathbf{b}_t(i)}{P(O | \mathbf{I})} \quad [\text{式 3. 37}]$$

若將時間  $t$  上所有  $g_t(i)$  加起來，可視為狀態  $i$  被拜訪過的期望次數，亦可解釋為狀態  $i$  轉移出去的期望次數。而針對  $x_t(i, j)$  來說，可看成狀態  $i$  轉移到狀態  $j$  的期望次數。

因此，我們可以根據目前參數  $\alpha$ 、 $A$ 、 $B$ ，由波氏演算法得知：

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \mathbf{x}_t(i, j)}{\sum_{t=1}^{T-1} g_t(i)} \quad \dots \quad [\text{式 3. 39}]$$

上式表示狀態轉移機率重估值  $\bar{a}_{ij}$ ，其分母為停在狀態  $i$  的期望值，其分子為狀態  $i$  轉移到狀態  $j$  的期望值。

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \mathbf{g}_t(j)}{\sum_{t=1}^T o_t = v_k} \dots \text{[式 3. 40]}$$

上式表示狀態觀測機率重估值  $\bar{b}_j(k)$ ，其分母為停留在狀態  $j$  的期望值，其分子為停留在狀態  $j$  且觀測對象輸出為  $k$  的期望值。

### 3.4 辨識方式

上節已說明我們將語音訊號變成 HMM 模型的訓練過程，訓練的結果是每一個聲音都有一個模型。我們辨識語音的做法是找出該語音訊號最接近的模型。而計算與各聲音模型產生一個觀測符號的機率方法，一般都用維特比（Viterbi）演算法。以下是維特比演算法的介紹。

### 3.4.1 維特比演算法

維特比演算法是用來找尋最佳路徑的演算法。它是動態規劃的方法，在 HMM 狀態的轉移中，演算法必須記錄轉移至各狀態最大機率值和先前的狀態，一直到

達觀測符號的結束時刻  $T$  為止。在找出  $T$  時刻具有最大機率值的狀態之後，再按照剛才的記錄，往回找出這條最佳路徑 (Optimal Path) 所歷經的狀態序列。

維特比演算法計算如下：

假設輸入觀測序列  $C_1, C_2, \dots, C_T$ ，每個觀測時間  $C_i$  都有  $w_1, w_2, \dots, w_N$   $N$  個候選音，演算法步驟如下：

設定兩個陣列 SEQSCORE(n,t) 與 BACKPTR(n,t)，SEQSCORE(n,t) 是紀錄前面  $t$  個觀測沿單一路徑在  $C_i$  時，停留在  $w_n$  時的機率最高，而 BACKPTR(n,t) 是紀錄在觀測符號  $C_i$  中每一個字往前一個時間點回溯的最佳路徑。

### 1. 初始化步驟

For  $i=1$  to  $N$  do

$$\text{SEQSCORE}(i,1)=\text{PROB}(w_1 | w_0)$$

$$\text{BACKPTR}(i,1)=0$$

其中  $\text{PROB}(w_1 | w_0)$  是指  $w_1$  這個字當作一個語音的開始機率。

### 2. 遷迴運算步驟

For  $t=2$  to  $T$

For  $i=1$  to  $N$

$$\text{SEQSCORE}(i,t)=\text{MAX}_{j=1}^N (\text{SEQSCORE}(j,t-1) * \text{PROB}(w_i | w_j))$$

BACKPTR(i,t)=使得 SEQSCORE(i,t) 最大化之  $j$  值

### 3. 最佳序列取得步驟

$C(T)=i$  使得  $\text{SEQSCORE}(i,T)$  最大值

FOR  $i=T-1$  to do

$C(i)=\text{BACKPTR}(C(i+1), i+1)$

\

### 3.4.2 測試辨識器

#### 3.4.2.1 實驗內容設定

樣本來源是中央研究院收集的 MAT-2000-Edu 語料庫。我們隨機抽取語料。只有固定說者的性別，例如：詞語是「大學」，性別：女，就是「大學」這個詞訓練樣本及測試樣本的說者都是女性，而教育程度、年齡及其他因素都是隨機的。本詞語及性別如表 3.1。

表 3.1 辨識測試樣本

詞語	性別	訓練樣本個數	測試樣本個數
「大學」	女	20	18
「專科」	女	20	24
「閩南語」	女	20	28
「男性」	男	20	19
「國中」	女	20	29
「不是」	女	20	20
「研究所」	女	20	11
「國語」	女	20	21
「女性」	女	20	20
「沒有」	女	20	21

訓練方式：分別有一個詞訓練一個模型，有五個相同的詞訓練一個模型，全部相同詞（20 個）訓練一個模型，HMM 有四個 State。

辨識率：要計算辨識率，我們要先定義什麼叫做「接近正確辨識」。一個語言辨識器的「接近正確辨識」有很多計算方式。因為我們不單只有辨識器而已，後面還有語言模型與對話策略可以幫助提高辨識率。所以對於辨識器的「接近正確辨識」就下了這樣的定義：「如果一個辨識器辨識某段聲音，其辨識結果（數個候選字詞，例如 5 個）包含一個或以上正確聲音的話，稱為「接近正確辨識」」這種辨識率的定義是比較寬鬆的，我們只計算該詞是否在候選字詞內，例如：我們讓辨識器辨識一個「大學」的測試樣本，如果候選字詞含有「大學」，不管排在第幾個，也不管候選字詞內有多少個「大學」都是「接近正確辨識」。採用這樣定義的理由是我們辨識器的輸出是接到對話策略的輸入。當該辨識器將候選字給了對話策略之後，對話策略應該會做有效的判斷。這樣的話，其候選詞的順序

就顯得不是那麼重要了。相反的「接近正確辨識率」就很重要了，如果辨識結果不包括正確的語言，就沒有辦法給對話策略使無法產生正確的結果。因此辨識率就等於：

$$\frac{\text{接近正確辨識的測試樣本數}}{\text{全部的測試樣本數}} \dots \dots \dots \text{[式 3. 41]}$$

全部的樣本數是指全部給辨識器辨識的樣本數。接近正確辨識的測試樣本數是指辨識器的辨識結果出現「接近正確辨識」的樣本數。

「候選字詞數」在本實驗是 5 個。

### 3.4.2.2 實驗結果

表 3.2 一個詞訓練一個模型

詞語	測試樣本個數	辨識結果位於五個候選字詞內的測試樣本數	百分比(%)
「大學」	18	15	83
「專科」	24	15	79
「閩南語」	28	23	82
「男性」	19	16	84
「國中」	29	23	79
「不是」	20	17	85
「研究所」	11	8	73
「國語」	21	16	76
「女性」	20	12	60
「沒有」	21	12	57
全部	211	157	74

$$\text{平均辨識率} = \frac{\text{接近正確辨識的測試樣本數}}{\text{全部的測試樣本數}} = \frac{157}{211} = 74\%$$

表 3.3 五個相同的詞訓練一個 Model

詞語	測試樣本個數	辨識結果位於五個候選字詞 內的測試樣本數	百分比(%)
「大學」	18	8	44
「專科」	24	14	58
「閩南語」	28	17	61
「男性」	19	13	68
「國中」	29	24	83
「不是」	20	12	60
「研究所」	11	10	91
「國語」	21	16	76
「女性」	20	13	65
「沒有」	21	14	67
總和	211	141	67

$$\text{平均辨識率} = \frac{\text{接近正確辨識的測試樣本數}}{\text{全部的測試樣本數}} = \frac{141}{211} = 67\%$$

表 3.4 全部相同詞 (20 個) 訓練一個 Model

詞語	測試樣本個數	辨識結果位於五個候選字詞 內的測試樣本數	百分比(%)
「大學」	18	15	83
「專科」	24	9	38
「閩南語」	28	18	64
「男性」	19	3	16
「國中」	29	19	66
「不是」	20	19	95
「研究所」	11	8	73
「國語」	21	14	67
「女性」	20	13	65
「沒有」	21	14	67
總和	211	132	63

$$\text{平均辨識率} = \frac{\text{接近正確辨識的測試樣本數}}{\text{全部的測試樣本數}} = \frac{132}{211} = 63\%$$

### 3.3.2.3 討論

我們將上面的實驗結果，再加以整理，將「詞語」、「測試樣本個數」欄合併。整理後的表 3.5：

表 3.5 實驗數據整理

詞語	測試樣本 個數	一個詞訓 練一個模 型	五個相同的 詞訓練一個 模型	全部相同詞訓 練一個模型
「大學」	18	15	8	15
「專科」	24	15	14	9
「閩南語」	28	23	17	18
「男性」	19	16	13	3
「國中」	29	23	24	19
「不是」	20	17	12	19
「研究所」	11	8	10	8
「國語」	21	16	16	14
「女性」	20	12	13	13
「沒有」	21	12	14	14
總和	211	157	141	132
平均辨識率		74%	67%	63%

很顯然的在訓練時，用一個聲音樣本去訓練一個 HMM 模型的效果是比較好的。其原因或許是因為 HMM 模型，會將該模型的參數調到對於該觀測符號產生最大的機率值。如果給該模型另外一組觀測符號的話，那麼原先所找到的模型參數或許會被改變，然後導致該模型對於原先的觀測符號不會產生最大的機率值。以至於該模型遇到類似於原先所訓練的觀測符號的測試樣本，其機率值也下降，所以辨識率就下降。

## 3.5 本對話策略的對話內容詞彙辨識

在這裡我們要做本對話策略的對話內容詞彙辨識的實驗。樣本方面：因為我們在中央研究院的 MAT-2000-Edu 語料庫中找不到我們想要的詞彙，因此就用吾人（性別：男）的聲音樣本。每個詞語的訓練樣本都是 10 個，而且用一個聲音樣本去訓練一個 HMM 模型。「候選字詞」在本實驗是 3 個。測試結果如下：

表 3 6 對話策略的對話內容詞彙辨識結果

詞語	測試樣本個數	辨識結果位於三個候選字詞內的 測試樣本數	百分比(%)
「需要冰塊」	8	4	50
「需要奶精」	9	8	89
「不要冰塊」	16	12	75
「不要奶精」	14	12	86
「不要糖」	9	8	89
「需要糖」	9	7	78
總和	65	51	78

$$\text{平均辨識率} = \frac{\text{接近正確辨識的測試樣本數}}{\text{全部的測試樣本數}} = \frac{51}{65} = 78\%$$



## 第四章 強化學習運用於對話策略理論基礎

### 4.1 強化學習理論介紹

強化學習是人工智慧中策略學習的一種，基於預期最大利益原則。強化學習是一種從很多條可供選擇的路中，挑選一條比較好的路，但是在一般情況下，我們並不可能在短時間內把一個問題的所有可能條件通通找出來，而且更不可能將所有的條件一個一個的去嘗試，我們只能由知道一小部份的條件，由代理人慢慢的找出一條較好的路，最後根據這些條件中找到最佳的路徑。

強化學習的任務是使用環境對行為的獎勵或懲罰去學得行為的決策函數。這是一個困難的問題，因為代理人不被直接告知什麼一定是對的行為，也不知道決定最終的輸贏的關鍵行為是哪一種。

「一個競局代理人可能除了一次失誤之外，全都進行的毫無瑕疵，但在競局結束後，得到一個單一的強化訊號，被告知「你輸了」，代理人就要想辦法找出那一次失誤。」( Russell & Norvig, 2003 )

強化學習的基本要素是：策略 (policy) 定義代理人的行為。它是被用來代理人某個狀態決定下一個可能的動作之參考依據。

“ At each time step, the agent implements a mapping from states to probabilities of selecting each possible action. This mapping is called agent's policy ...”(Sutton & Barto ,1998)

對話來說，就是代理人在某個狀態底下，從很多可能的可以選擇的對話中，挑出某一句話來說，而決定選擇哪一句話去回應的對應就叫「對話策略」。回饋 (reward) 幫助代理人去到達目標。以對話來說，就是代理人用了某種溝通策略之後，它從使用者得到的強化訊號（例如：NJFun 是使用者給的'Good'，'so-so'，'bad'）期望值 (value) 表示代理人的下一步期望值。以對話來說，就是代理人在某個狀態決定說哪一句話的決定因數。環境模型 (model of environment) 是一個模擬環境，以對話情境來說，就是使用者。

### 4.2 對話策略訓練

我們以店員和顧客為主題來設計訓練店員對話策略的系統，其對話內容是假設一個顧客來到一家咖啡店要點咖啡。顧客向店員點了咖啡，店員還會問他，這杯咖啡是要加什麼配料，配料有冰塊、有奶精、也有糖。然後在店員的對話中有很多問題是有時合理、有時不合理。例如：顧客已經表示過要糖了，但是，店員

這時又要問顧客，咖啡要加糖嗎？原則上這是不合理的，因為顧客已經表示過。因此顧客會以較為不滿的言語，告訴店員。店員就會自動修正對話策略。讓他跟顧客對話從中學到合乎邏輯的對話策略。

原則上，我們是用強化學習的方式要求來作為店員學習方法的核心。為了以即時回饋的方式，來修正其對話策略，我們要求演算法能夠每走一步都可以給予回饋。在強化學習的理論裡，有幾種學習方法來修正其對話策略。我們選用 Sarsa 演算法(Sutton & Barto, 1998)作為學習演算法。如圖 4.1：

Initialize  $Q(s,a)$  arbitrarily

Repeat (for each episode):

    Initialize  $s$

    Choose  $a$  from  $s$  using policy derived from  $Q$  (e.g.  $\epsilon$ -greedy)

    Repeat (for each step of episode):

        Take action  $a$ , observe  $r, s'$

        Choose  $a'$  from  $s'$  using policy derived from  $Q$  (e.g.  $\epsilon$ -greedy)

$Q(s,a) \leftarrow Q(s,a) + \alpha[r + Q(s',a') - Q(s,a)]$

$s \leftarrow s'; a \leftarrow a'$ ;

    until  $s$  is terminal



$s$ : 狀態。以這個系統來說，代表現在的屬性狀態。就是顧客的咖啡配料需求情形。

表 4.1 是這個系統的所有狀態。

$a$ : action。在這裡是指店員所說的話。

$Q(s, a)$ : 該狀態選擇  $a$  的期望值。

$\epsilon$  探索機率：這裡是用  $\epsilon = \frac{1}{\# episodes}$ ，因為我們希望學習前期盡量探索；越到最後越不需要探索。

$\alpha$  : 學習速率，我們用  $\alpha = 0.1$ 。

## 4.3 實驗和討論

### 4.3.1 最短路徑學習

在套用演算法時，將演算法的成員（在這裡是 Sarsa 演算法的成員有  $s$ ,  $a$ ,  $Q(s, a), \dots$ ）對應到問題。

表 4.1 系統的所有狀態

狀態表：店員目前所得到顧客的資訊 (? : 顧客還沒說 Y: 顧客要 N: 顧客不要)		
要冰塊	要奶精	要糖
?	?	?
?	?	Y
?	?	N
?	Y	?
?	Y	Y
?	Y	N
?	N	?
?	N	Y
?	N	N
Y	?	?
Y	?	Y
Y	?	N
Y	Y	?
Y	Y	Y

#### 4.3.1.1 店員模型

使用 Sarsa 演算法為學習核心，狀態數有 27 個。如表 3.1。

每一個狀態有三個屬性 (Attributes)，分別是冰塊、奶精、糖。每一個屬性有三個值分別是：0=顧客還沒說、1=要、2=不要。

每一個狀態有八個動作 (Actions)，分別表示店員可以問的八個問題：

「需要冰塊嗎？」

「需要奶精嗎？」

「需要糖嗎？」

- 「需要奶精和冰塊嗎？」
- 「需要冰塊和糖嗎？」
- 「需要奶精和糖嗎？」
- 「需要冰塊,奶精和糖嗎？」
- 「你的咖啡,[配料值],確定好了嗎?」(確認)

#### 4.3.1.2 決定動作方法

店員在一般狀態中，都只有前七種動作可以選擇。當店員的狀態轉移到顧客咖啡的配料都被設定的狀態時，店員才可以選第八種動作。店員狀態的轉移是靠店員選定問題（action），對顧客發問。顧客給回應後，店員的新狀態便確定下來，完成狀態的轉移，如圖 4.2。而動作的選擇是用最大期望值或是探索的方式，完全是由  $e$  決定。例如：假設  $e$  為 0.1，系統產生一個 0 與 1 之間的亂數，如果亂數大於 0.1，便選用目前狀態下評估最佳的動作，否則便隨機選取一個動作。

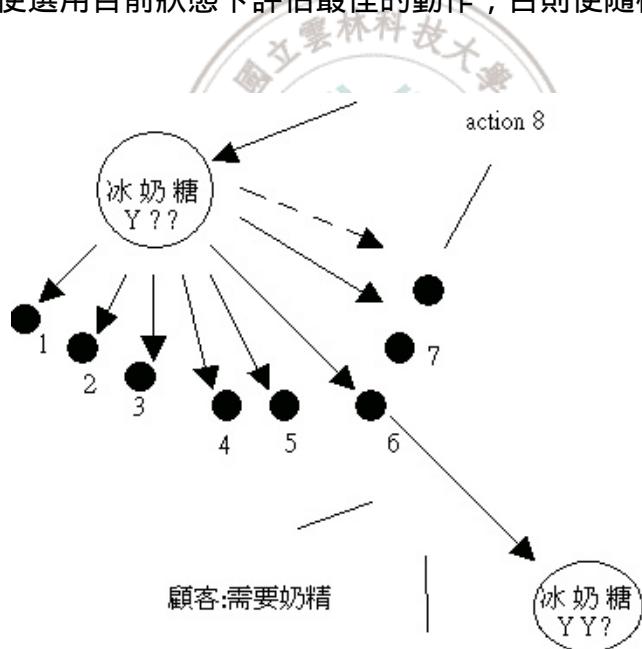


圖 4.2 狀態轉移圖

#### 4.3.1.3 即時回饋的給定

原則上，店員每問一個問題時，回饋分數給-1分。當顧客覺得店員的問題不合邏輯時，就再度被扣1分。在這裡我們只有考慮一個不合邏輯的狀況：當顧客

告訴了店員配料值，但是還是詢問顧客時。當店員重複問幾個配料值，加扣幾分。

#### 4.3.1.4 顧客(環境模型)

顧客回答方式有 29 種，這 29 種如下：

- (要,不要)冰塊 [2 種]
- (要,不要)奶精 [2 種]
- (要,不要)糖 [2 種]
- (要,不要)奶精和(要,不要)冰塊 [4 種]
- (要,不要)冰塊和(要,不要)糖 [4 種]
- (要,不要)奶精和(要,不要)糖 [4 種]
- (要,不要)冰塊, (要,不要)奶精和(要,不要)糖 [8 種]
- 是(意指:顧客決定不改變任何屬性) [1 種]
- 還沒(意指:顧客還要改變屬性) [1 種]
- 你很煩(意指:顧客生氣了)(選擇的) [1 種]

顧客類型使用兩個參數：決定 pb1=主動型的程度、pb2=反覆的程度，因此顧客在回答店員的問題之前要先決定下列事項：

1. 如果店員重複問的話，客戶便跟店員說：「你很煩！」
2. 決定是否配合回答問題
3. 決定是否改變主意
4. if (決定要改變主意) 變更顧客想要的配料
5. If (店員要顧客確認)
  - (a) If (顧客要改變主意)
    - i. If (要配合回答) 只說改變的部分
    - ii. else 全部說出
  - (b) else 跟店員說決定不改變
6. else
  - (a) 檢查在此狀態店員有沒有重複問，藉此顧客可以表現出他的情緒反應。
  - (b) if (要配合回答) 依照店員的問題回答
    - else 以每個配料的要或不要機率為 0.5，來決定是否說出要不要該配料。(跟店員的問題沒關係)

顧客類型有四個極端的組合。如表 4.2 所示。(一)配合回答型：店員問什麼，回答什麼。(二)反覆但配合回答：若有機會，會改變屬性值。(三)主動型而不反覆：完全不理會店員的疑問，說出自己的需求。(四)主動型和反覆：完全不理會店員的疑問，說出自己的需求。且若有機會，會改變屬性值。

表 4.2 顧客類型的組合

pb1 \ pb2	0 不反覆	1 反覆
0 被動型	配合回答型	反覆但配合回答
1 主動型	主動型而不反覆	主動型和反覆

原則上，本系統將顧客分成 25 種，如下 4.3 表。但是有 5 個是反覆程度為 1 的，是不會收斂的，因此我們不考慮，所以實際上，我們只考慮了 20 種。

表 4.3 實驗的顧客類型

Pb1 \ Pb2	0(不反覆)	0.25	0.5	0.75	1 ( 反覆 )
0(被動回答型)	V	V	V	V	X
0.25	V	V	V	V	X
0.5	V	V	V	V	X
0.75	V	V	V	V	X
1 ( 主動型 )	V	V	V	V	X

#### 4.3.1.5 實驗過程

我們訓練店員的對話策略的方式是讓每一種類型的顧客與店員對話 10000 回合。一回合是指店員從問第 1 個問題到店員問：「顧客，確定好了嗎？」顧客答：「是」在此設定實驗中我們  $e = 1/\text{回合數}$ 。店員剛開始會隨便亂問，有的時候會問同樣的問題。底下的對話是店員的訓練過程，顧客的參數  $pb1$  ( 主動 ) = 0、 $pb2$  ( 反覆 ) = 0 ( 也就是配合回答型的顧客 )。

回合:1

店員:需要奶精和冰塊嗎?

顧客:需要冰塊,需要奶精[-1]

店員:需要奶精嗎?

顧客:你很煩! [-1]

(因為前面顧客有說要奶精，這裡又問要奶精嗎？因此顧客會心裡覺得不舒服，所以會給店員不好的回饋。)

顧客:需要奶精 [-1]

店員:需要糖嗎?

顧客:需要糖 [-1]

店員:你的咖啡,加冰塊,加奶精,加糖,確定好了嗎?

顧客:是 [-1] 回饋總和 : [-5]

回合:2

店員:需要冰塊嗎?

顧客:需要冰塊 [-1]

店員:需要奶精和冰塊嗎?

顧客:你很煩! [-1]

顧客:需要冰塊,需要奶精 [-1]

店員:需要奶精和冰塊嗎?

顧客:你很煩! [-2] ( 兩個屬性都重複問 )

顧客:需要冰塊,需要奶精 [-1]

店員:需要冰塊,奶精和糖嗎?

顧客:你很煩! [-2] ( 兩個屬性都重複問 )

顧客:需要冰塊,需要奶精,不要糖 [-1]

店員:你的咖啡,加冰塊,加奶精,不加糖,確定了嗎?

顧客:是 [-1] 回饋總和 : [-10]

到這裡為止，店員還是會問重複的問題。因為該狀態中，問重複問題的對話動作期望值比較高。所以代理人會選擇那個比較高的對話動作。

回合:9999

店員:需要奶精和糖嗎?

顧客:不要奶精,需要糖 [-1]

店員:需要冰塊嗎?

顧客:不要冰塊 [-1]

店員:你的咖啡,不加冰塊,不加奶精,加糖,確定了嗎?

顧客:是 [-1] 回饋總和 : [-3]

回合:10000

店員:需要冰塊嗎?

顧客:不要冰塊 [-1]

店員:需要奶精嗎?

顧客:需要奶精 [-1]

店員:需要糖嗎?

顧客:需要糖 [-1]

店員:你的咖啡,不加冰塊,加奶精,加糖,確定了嗎?

顧客:是 [-1] 回饋總和 : [-3]

當對話訓練結束，我們看看最後的幾個回合的對話，店員不會問重複的問題。因為該狀態中，問重複問題的對話動作期望值已被修正得很低。

下面是店員與 20 種顧客類型對話的回合數 (x 軸) — 回饋 (y 軸) 圖的表格 (圖 4.3-4.11)。考慮圖 4.3，固定主動程度為 0。我們可以看到在前面 2500~3000 回合，店員是在學習如何對答，所以回饋才會如此低。當反覆機率=0 時，回饋的震盪是最小，而隨著反覆機率變高，其震盪是也越大。我們以為反覆機率越高，越快收斂，表示學習效果越好。當回饋收斂時，其回饋的震盪幅度會隨著反覆機率升高而增加。

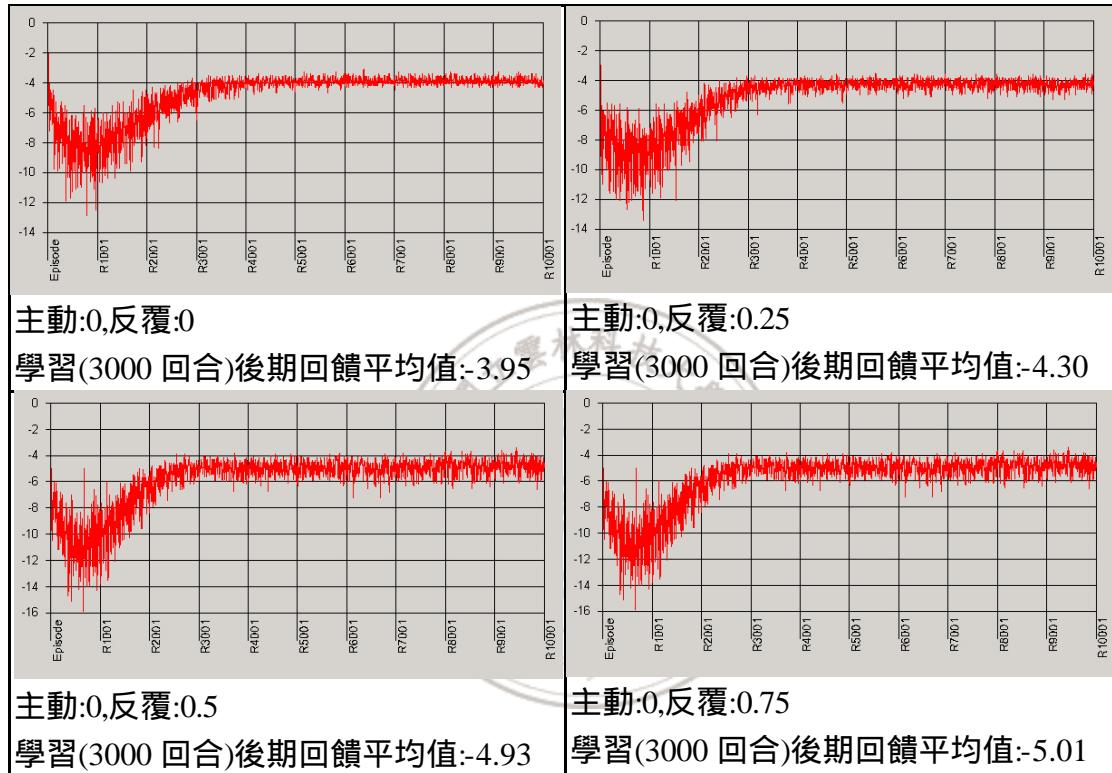


圖 4.3 固定主動程度=0 的結果

考慮圖 4.4，固定主動程度為 0.25。如果反覆機率增加的話，在學習完 3000 個回合後，回饋總和還是會震盪的，其震盪的原因是客戶不斷的改變主意。整體而言，其反覆機率越高，回饋越低。則表示顧客反覆、改變主意的情形越嚴重，店員較難應付。

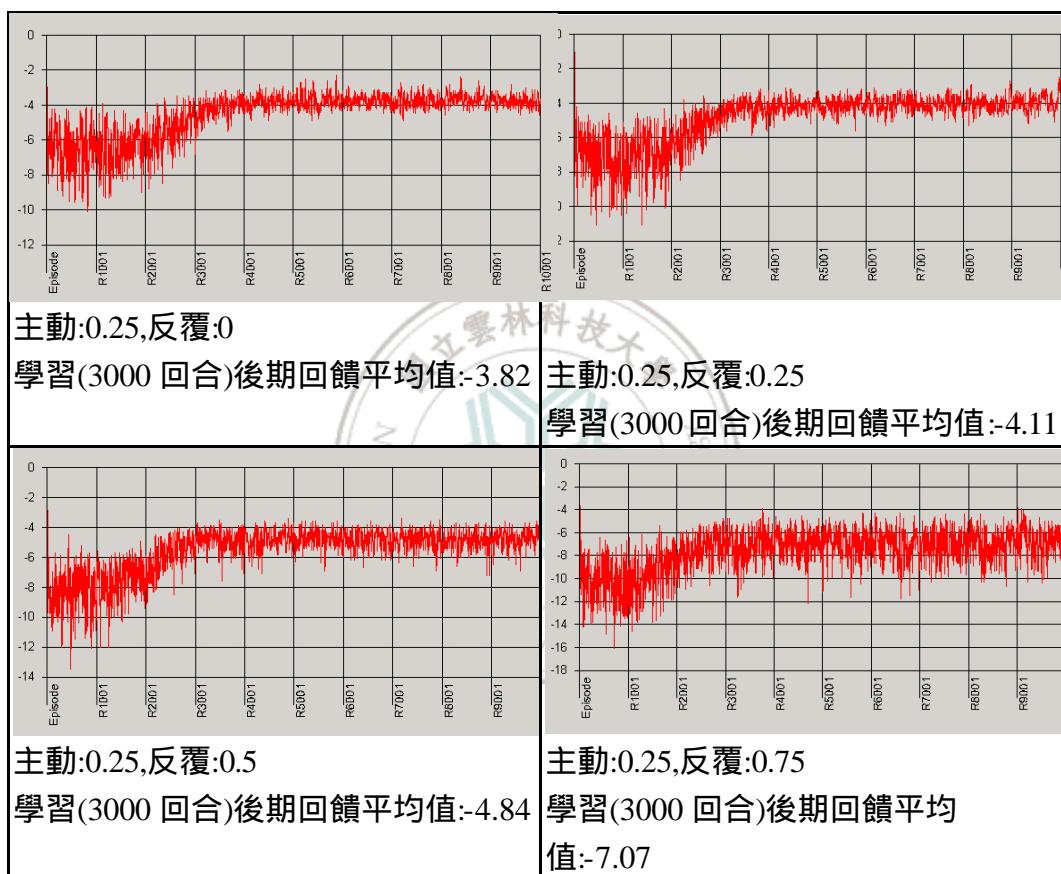


圖 4.4 固定主動程度=0.25 的結果

考慮圖 4.5，固定主動程度為 0.5。如果反覆機率增加的話，就會越難看出學習的部分。但若主動型機率比反覆機率大時，收斂後回饋震盪的幅度會變小。

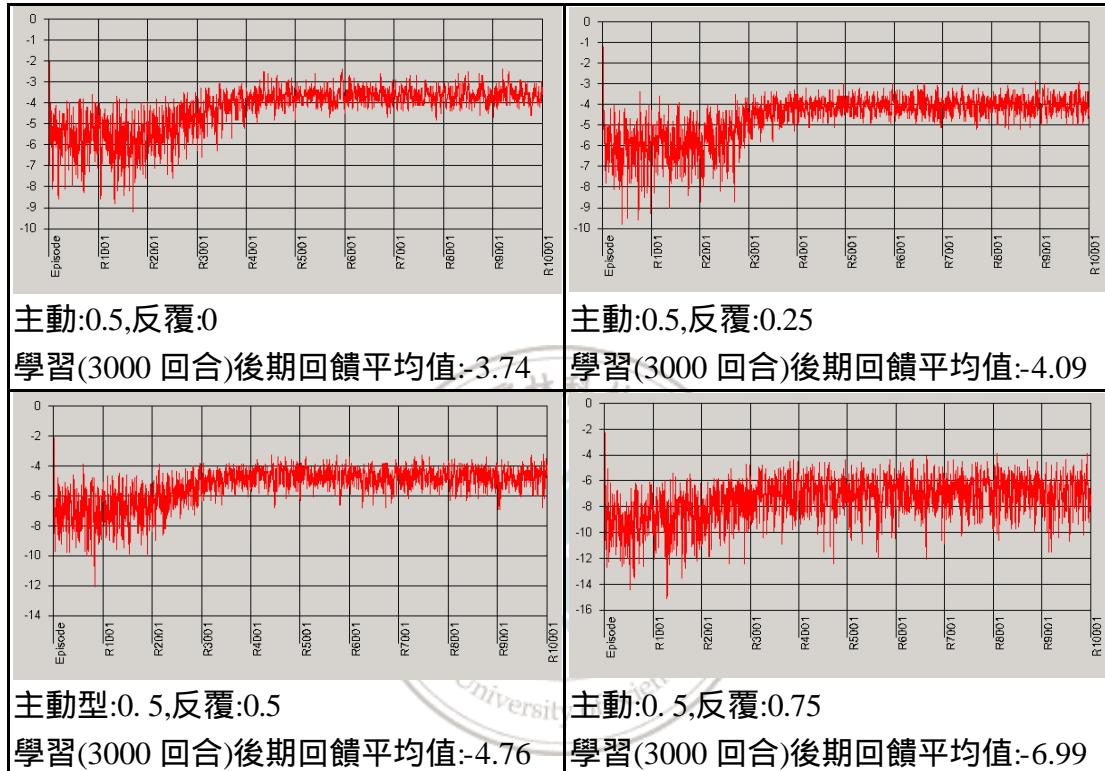


圖 4.5 固定主動程度=0.5 的結果

考慮圖 4.6，固定主動程度為 0.75。這裡就很難分出學習時期的曲線，因為學習前 3000 回合的時候，回饋值震盪較大；在學習 3000 回合後之學習過程中，回饋值震盪與前期是差不多的。主動型機率比反覆機率大時，回饋會增加。

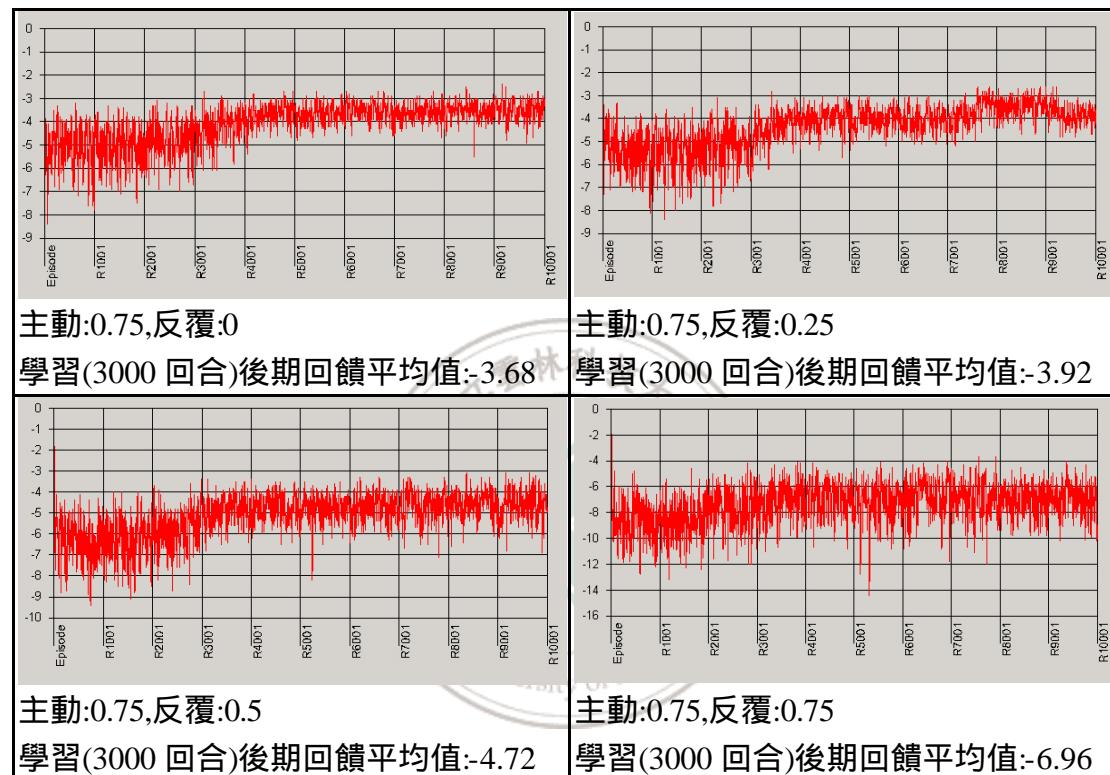


圖 4.6 固定主動程度=0.75 的結果

考慮圖 4.7，固定主動程度為 1。看不出有收斂的情形，因為店員與自主性強的顧客對話的時候，顧客所回答的答案跟店員的問題是答非所問，同樣的問題重複好幾遍以至於回饋變得很低。

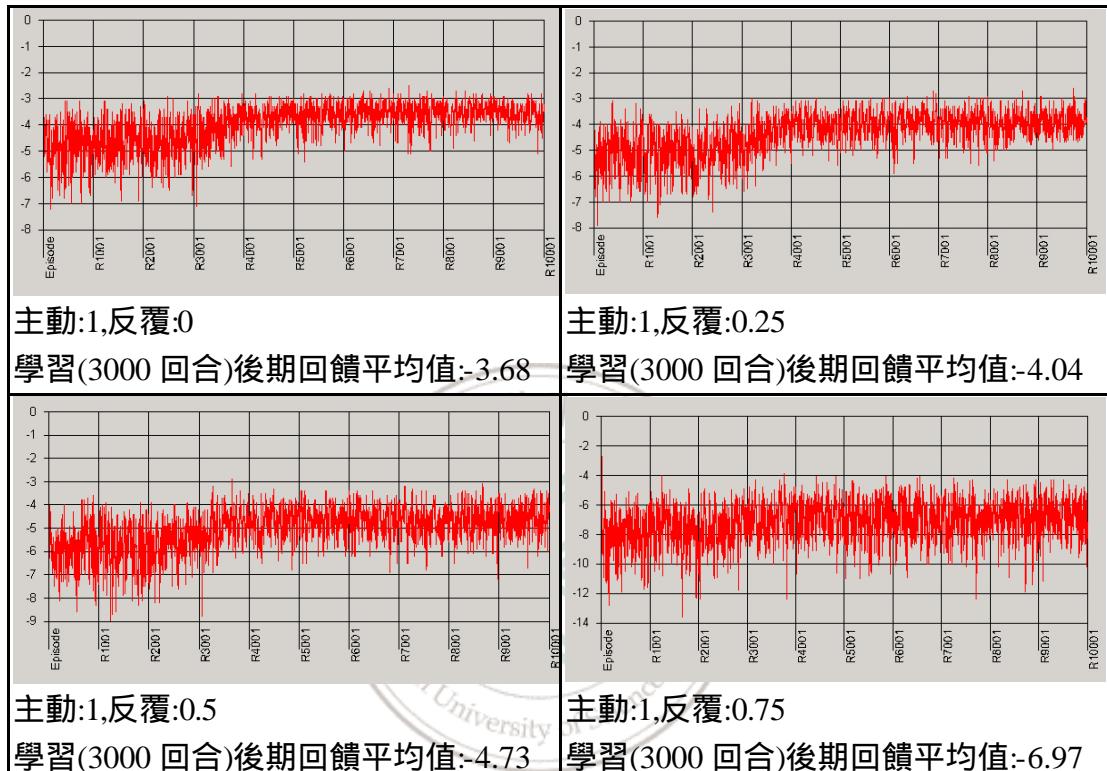


圖 4.7 固定主動程度=1 的結果

從圖 4.8 至 4.11，每張圖將反覆程度固定，再考慮主動程度從 0 遷增至 1。先參考圖 4.8，固定反覆程度為 0。主動型機率會影響學習時回饋的穩定度，主動型機率越高的話，回饋越高，但是回饋震盪越大。

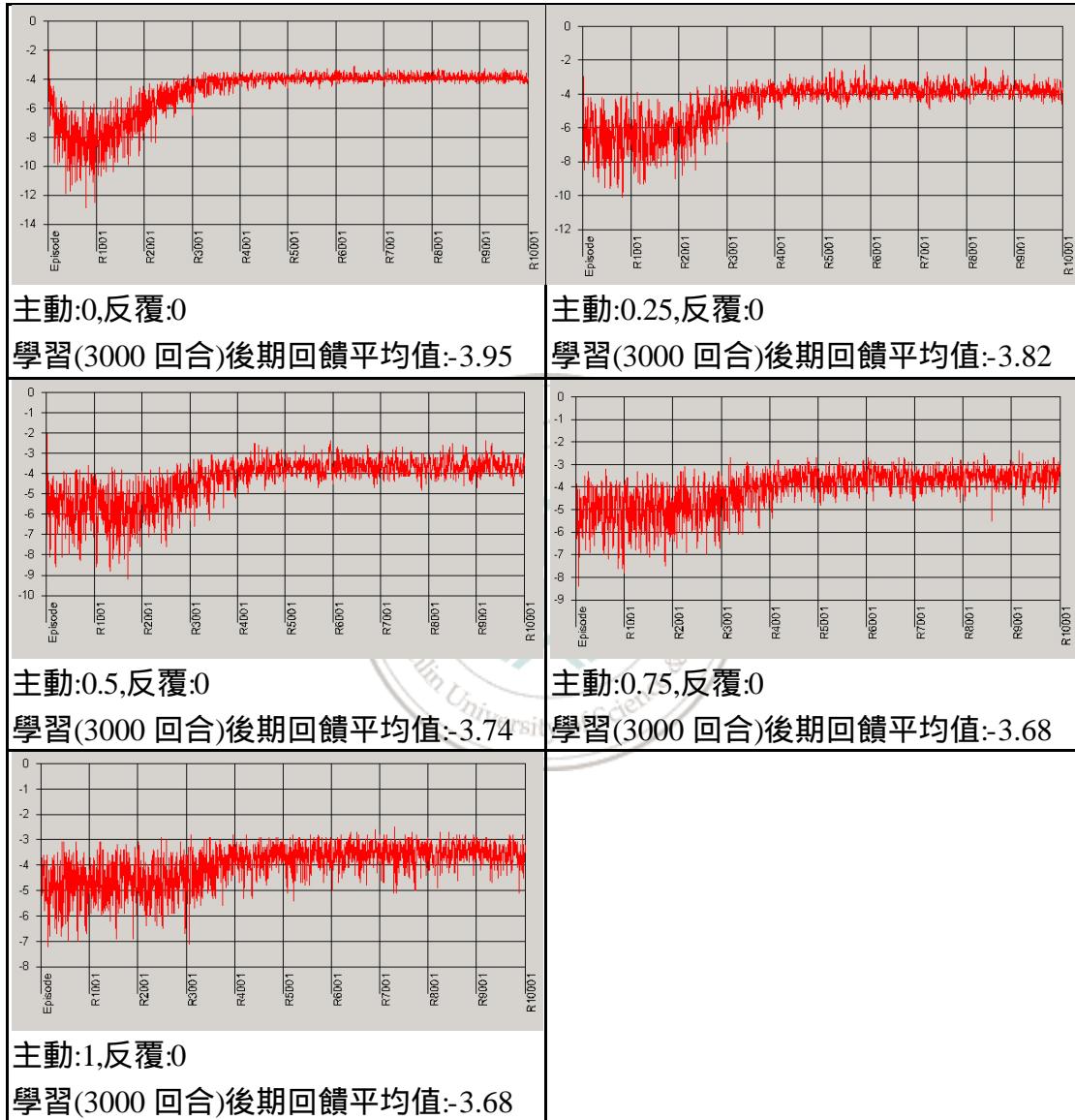


圖 4.8 固定反覆程度=0 的結果

參考圖 4.9 固定反覆程度為 0.25。當主動機率較高時候，整個回饋就會提高，因為主動程度越高，就越會忽略店員所問的問題。

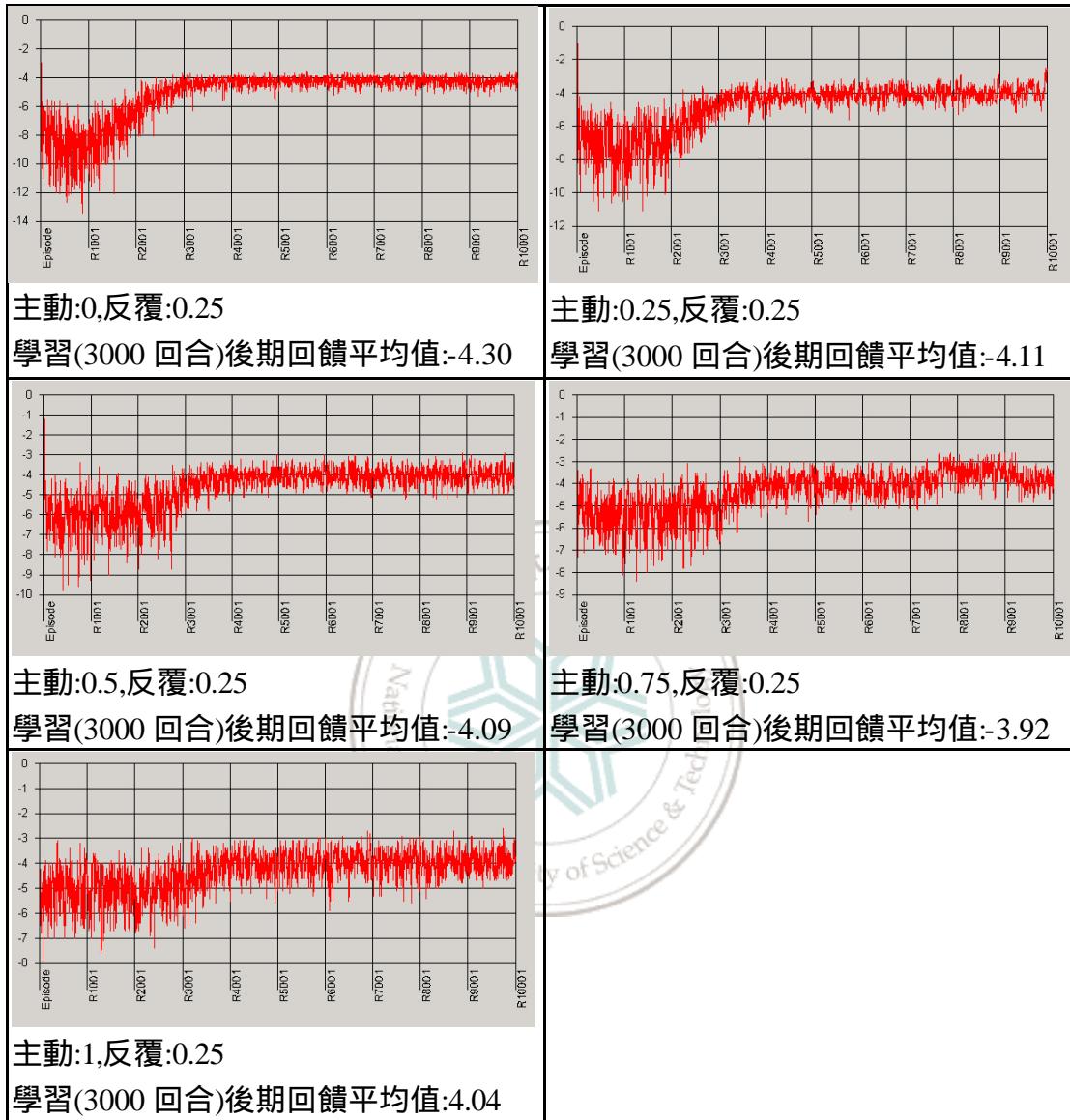


圖 4.9 固定反覆程度=0.25 的結果

參考圖 4.10，固定反覆程度為 0.5。當主動型機率提高，有某種程度的幫助回饋增加，整體回饋會比較高。比起反覆程度為 0.25 的情況，反覆機率:0.5 就要多問一次。

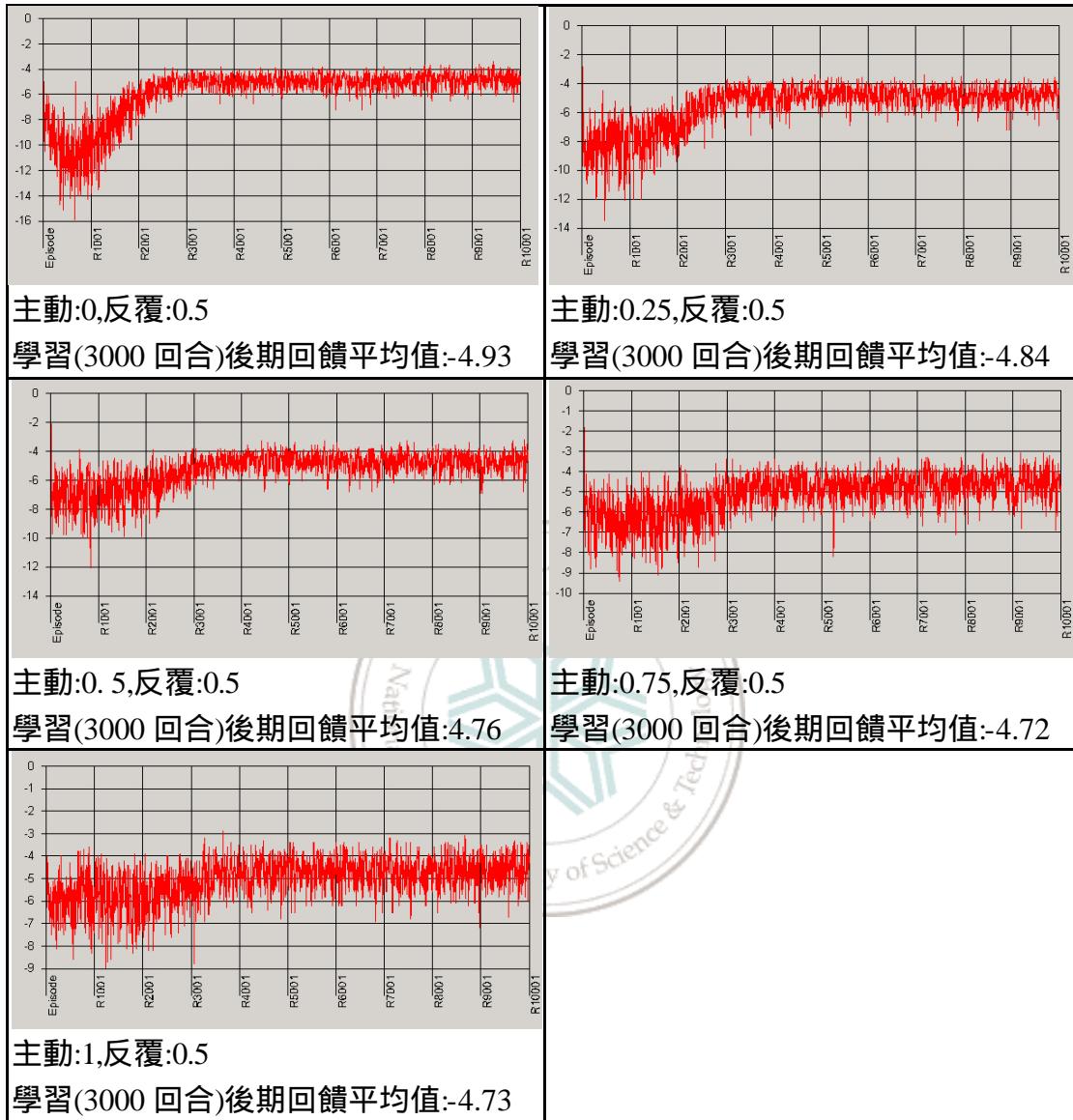


圖 4.10 固定反覆程度=0.5 的結果

參考圖 4.11，固定反覆程度為 0.75。在反覆程度為 0.75，主動為 0 的情況下，就比在反覆程度為 0，主動為 0 平均要多問 1 次。因為反覆程度為 0.75，主動為 0 的學習後期平均值是 -5.01。而反覆程度為 0，主動為 0 的學習後期平均值是 -3.95。根據觀察，主動性越高，則會讓回饋越震盪，因此主動和反覆會互相作用。

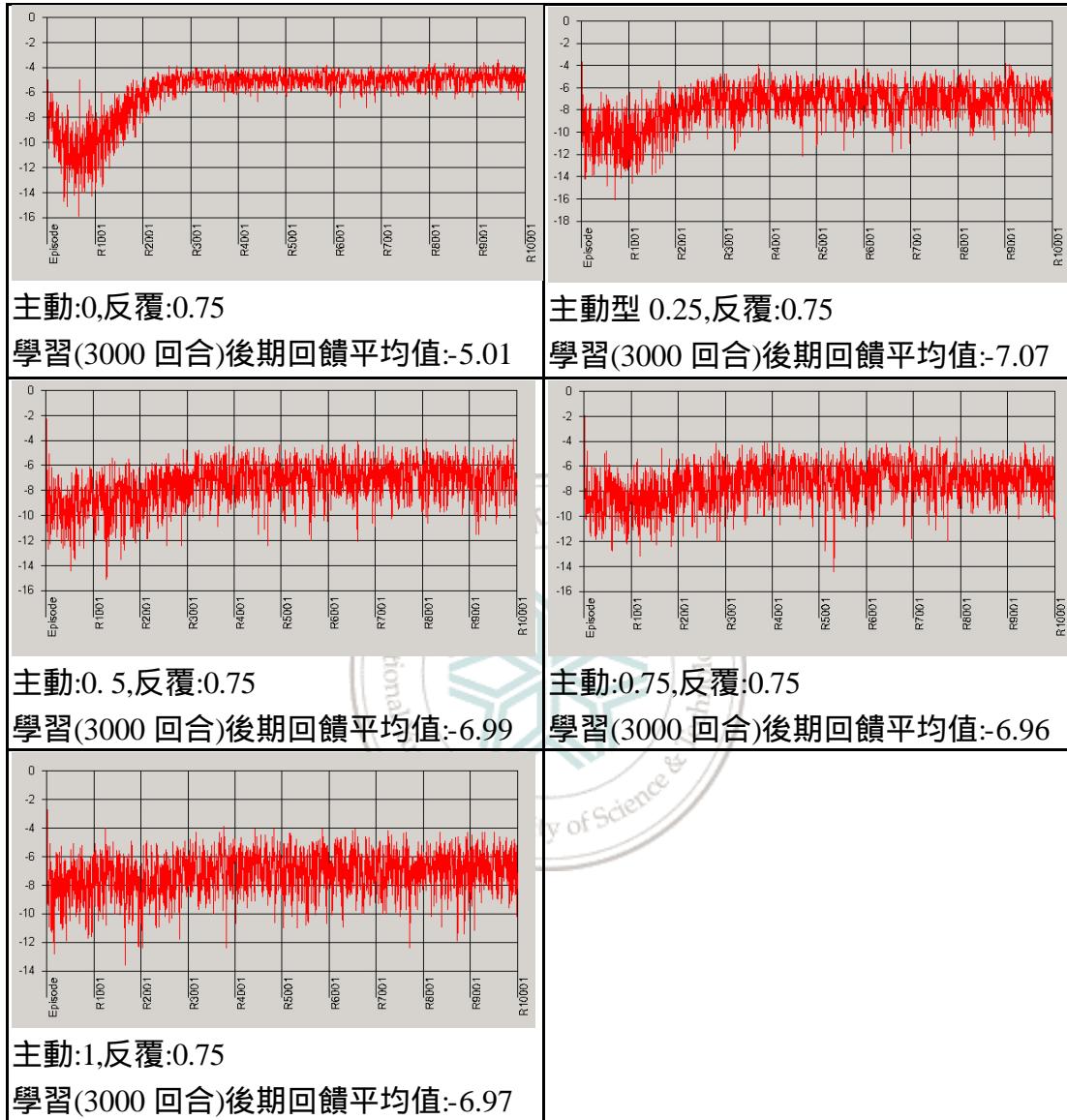


圖 4.11 固定反覆程度=0.75 的結果

#### 4.3.1.6 實驗結論

實驗結果告訴我們，強化學習方法能夠讓電腦學到合理的策略。店員剛開始會重複問相同的問題，而到後來就知道在什麼情況下該問什麼問題。雖然有的圖

很亂，其原因並非店員學不到合理的策略，而是顧客的反覆性、主動性的關係。

將圖 4.3-圖 4.11 的學習(3000 回合)後期回饋平均值建表如下：

表 4.4 學習(3000 回合)後期回饋平均值表

主動 反覆	0	0.25	0.5	0.75	1	平均值
0	-3.95	-3.82	-3.74	-3.68	-3.68	-3.774
0.25	-4.3	-4.11	-4.09	-3.92	-4.04	-4.092
0.5	-4.93	-4.84	-4.76	-4.72	-4.73	-4.796
0.75	-5.01	-7.01	-6.99	-6.96	-6.97	-6.588
平均值	-4.5475	-4.945	-4.895	-4.82	-4.855	

由此表 4.4 可見，觀察橫列得知，當店員反覆參數為 0 的時候，主動型的後期回饋平均值為 -3.74。表示店員平均要問 3-4 次才能知道顧客的需求。依此類推，當店員反覆參數為 0.25 的時候。表示店員平均要問 4 次才能知道顧客的需求。當店員反覆參數為 0.5 的時候。店員平均要問 4-5 次才能知道顧客的需求。當店員反覆參數為 0.75 的時候。店員平均要問 6-7 次才能知道顧客的需求。但是觀察縱行得知，其平均值均為 -4~ -5 之間。店員平均要問 4-5 次才能知道顧客的需求。

原先顧客類型有四個極端的組合。將其學習後期回饋平均值建表 4.5 如下：

表 4.5 顧客類型的組合

pb1 pb2	0 不反覆	0.75 反覆
0 被動型	配合回答型(-3.95)	反覆但配合回答 (-5.01)
1 主動型	主動型而不反覆(-3.68)	主動型和反覆 (-6.97)

我們發現配合回答型的回饋約為 -4，而主動型而不反覆的回饋約為 -3.68，因為主動型的顧客能夠主動表達自己內心的需求。當與反覆但配合回答的顧客對話時，回饋約為 -5，但與主動型和反覆的顧客對話時，回饋約為 -7。因為反覆但配合回答顧客的反悔原因是點錯了配料，會修正一、二次後，就會表達自己內心的需求。主動型和反覆是顧客會反悔的情況，但他自己內心真正的需求卻舉棋不定，所以會讓店員多問幾次。

### 4.3.2 學習確認

店員狀態和回饋值跟前面「最短路徑學習」的假設大致上都是一樣。只是這

個實驗的目標是要讓店員能夠學得「什麼時候發出確認的訊息」(第八種動作)。在「最短路徑學習」的實驗中，其確認（「你的咖啡,[配料值],確定好了嗎？」）的機制是固定的（當店員的狀態轉移到顧客咖啡的配料都被設定的狀態時，店員才可以選第八種動作）。然而，在這裡先將這個機制（判斷配料是否都被設定）拿掉，看他是否能同樣用 Sarsa 演算法學習到適時發出確認的機制。

#### 4.3.2.1 決定動作方法

在前面「最短路徑學習」的實驗中，店員在一般狀態底下，都只有前七種動作可以選擇。當店員的狀態轉移到顧客咖啡的配料都被設定的狀態時，店員才可以選第八種動作。而在這裡，我們讓店員在所有狀態都有機會選擇那八個動作之中的其中一種。而其他的假設也都一樣。

#### 4.3.2.2 回饋的給定

原則上，在這裡的回饋跟「最短路徑學習」實驗假設方式是一樣。但為了讓店員學習到「在某些狀態下不能選擇第八種動作」，所以當店員選擇第八種動作，顧客就要檢查顧客自己配料值是否全部說出，如果尚未全部說出，就給予「拜託！我都還沒講完！」的回應並且加扣 1 分。因此一共有 30 種回應方式。而「顧客在回答店員的問題之前要先決定下列事項」要修改如下：

```
if(店員選擇第八種動作)
    if(顧客自己配料值未全部說出) 客戶便跟店員說：「拜託！我都還沒講完！」
    else
        如果店員重複問的話，客戶便跟店員說：「你很煩！」
        決定是否配合回答問題
        決定是否改變主意
        if (決定要改變主意) 改變顧客想要的配料
        If (店員的這次是要讓顧客確認)
            If (決定要改變主意)
                If (顧客主動) 全部說出自己的需求
                else 只說改變的部分
            else 跟店員說決定不改變
        else
            檢查在此狀態有沒有重複問，藉此顧客可以表現出他的情緒反應。
```

if (顧客主動) 全部說出自己的需求 (跟店員的問題沒關係)  
else 依照店員的問題回答。

#### 4.3.2.3 實驗過程

底下的對話是店員的訓練過程 ( $e = 0.1$ )，顧客的參數  $pb1$  (主動) =0、 $pb2$  (反覆) =0。這裡要特別聲明，這次實驗特別將  $e$  設成 0.1。與之前的實驗將  $e$  設成 (1/回合) 不同，即是訓練前期回合較小， $e$  較大，後期則  $e$  漸漸減小。這樣設定是因為想要看看學習效果有沒有因此而改變。

##### 回合:1

店員:需要糖嗎? [-1]

顧客:需要糖

店員:需要冰塊和糖嗎? [-1]

顧客:你很煩! [-1]

顧客:不要冰塊,需要糖

店員:你的咖啡,不加冰塊, 加糖,確定好了嗎? [-1]

顧客:拜託!我都還沒講完! [-1]

(因為顧客尚未表達完屬性值，所以會給店員扣一分。)

店員:需要冰塊,奶精和糖嗎? [-1]

顧客:你很煩! [-2]

顧客:不要冰塊,不要奶精,需要糖

店員:需要糖嗎? [-1]

顧客:你很煩! [-1]

顧客:需要糖

店員:需要糖嗎? [-1]

顧客:你很煩! [-1]

顧客:需要糖

店員:需要糖嗎? [-1]

顧客:你很煩! [-1]

顧客:需要糖

店員:你的咖啡,不加冰塊,不加奶精,加糖,確定好了嗎? [-1]

顧客:是

回饋總和 : -15

##### 回合:2

店員:需要奶精和糖嗎? [-1]

顧客:不要奶精,需要糖

店員:需要糖嗎? [-1]

顧客:你很煩! [-1]  
顧客:需要糖  
店員:需要糖嗎? [-1]  
顧客:你很煩! [-1]  
顧客:需要糖  
店員:需要糖嗎? [-1]  
顧客:你很煩! [-1]  
顧客:需要糖  
店員:需要糖嗎? [-1]  
顧客:你很煩! [-1]  
顧客:需要糖  
店員:你的咖啡, ,不加奶精,加糖,確定了嗎? [-1]  
顧客:拜託!我都還沒講完! [-1]  
店員:需要奶精嗎? [-1]  
顧客:你很煩! [-1]  
顧客:不要奶精  
店員:需要奶精和冰塊嗎? [-1]  
顧客:你很煩! [-2]  
顧客:不要冰塊,不要奶精  
店員:你的咖啡,不加冰塊,不加奶精,加糖,確定了嗎? [-1]  
顧客:是

回饋總和 : -17

到這裡為止，店員還是會問重複的問題，而且還會顧客尚未表達完屬性之前，就會作確認。

#### 回合:10000

店員:需要冰塊,奶精和糖嗎? [-1]  
顧客:不要冰塊,不要奶精,需要糖  
店員:你的咖啡,不加冰塊,不加奶精,加糖,確定了嗎? [-1]  
顧客:是

回饋總和 : -2

當對話訓練結束，我們看看最後的幾個回合的對話，店員不但不會問重複的問題，而且還會等顧客表達完屬性之後，才會作確認。

#### 4.3.2.4 實驗結論

下面我們討論它的回饋-回合曲線圖。我們將圖 4.11 和圖 4.12 做比較，在反

覆=0 的部分，我們發現「學習確認」回饋的學習(3000 回合)後期回饋平均值:(主動=1 時，回饋=-2.55；主動=0 時，回饋=-2.29)比「最短路徑學習」(主動=1 時，回饋=-3.95；主動=0 時，回饋=3.68)好，反覆=1 的部分則是「最短路徑學習」(主動=1 時，回饋=-5.01；主動=0 時，回饋=-6.97)比「學習確認」(主動=1 時，回饋=-9.37；主動=0 時，回饋=主動=1 時，回饋=-6.25)好，而反覆機率越大的話，代表環境變動越大值。因此當環境變動較小值選擇  $e = 0.1$ 的結果會比較好，當環境變動較大值選擇  $e = 1/回合$  的結果會比較好。

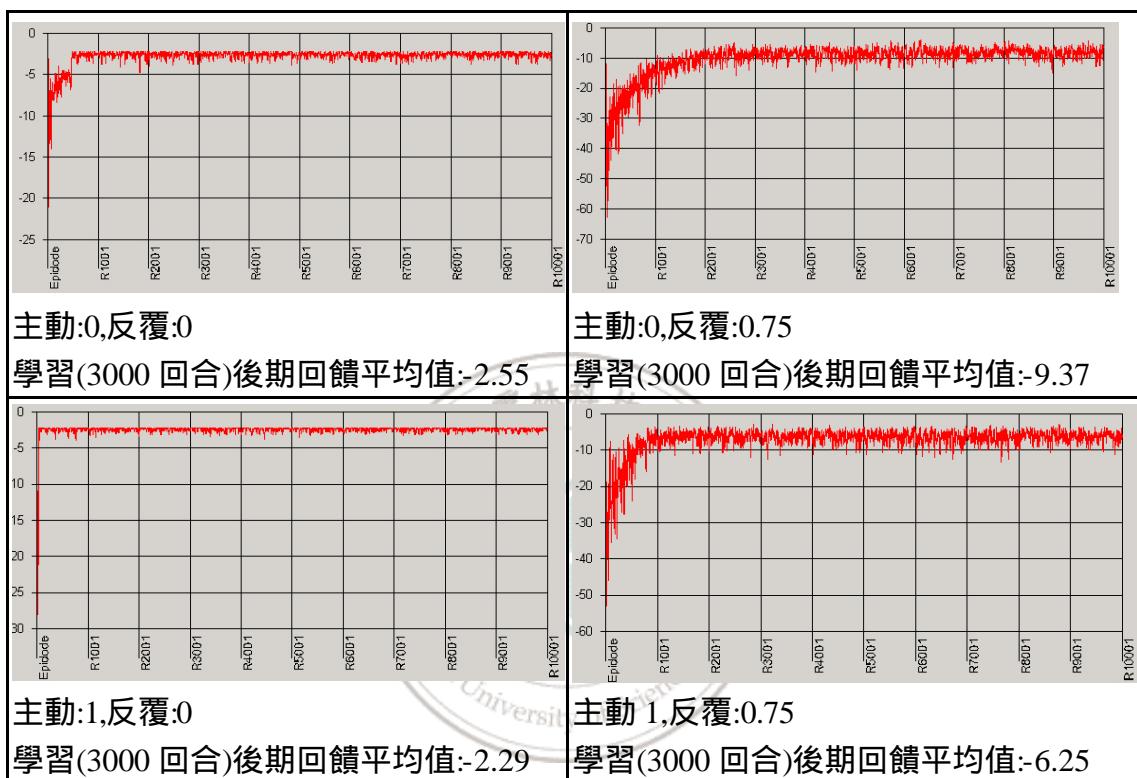


圖 4.12 「學習確認」極端例子的回饋 ( $e = 0.1$ )

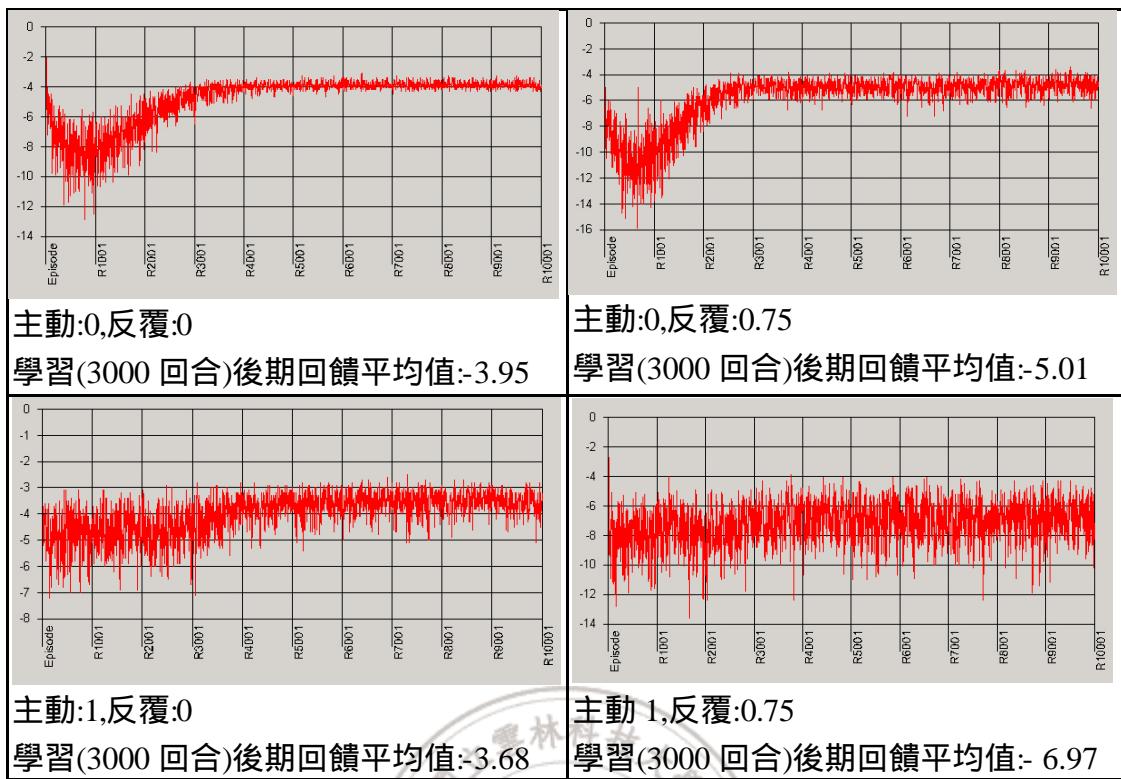


圖 4.13 「最短路徑學習」極端例子的回饋 (  $\epsilon = 1/\text{回合}$  )

## 第五章 總結

### 5.1 組合語音辨識和對話策略

本論文已經在前兩章介紹過語音辨識和對話策略。語音辨識方面介紹了語音訊號處理和語音模型的建立以及辨識的方法。對話策略方面介紹如何讓咖啡店的店員從會問不合理的問題，訓練到只會問合理的問題。接下來就是要將他們結合成為個系統。圖 5.1 是我們的詳細的系統架構，它比圖 1.1 更詳細。

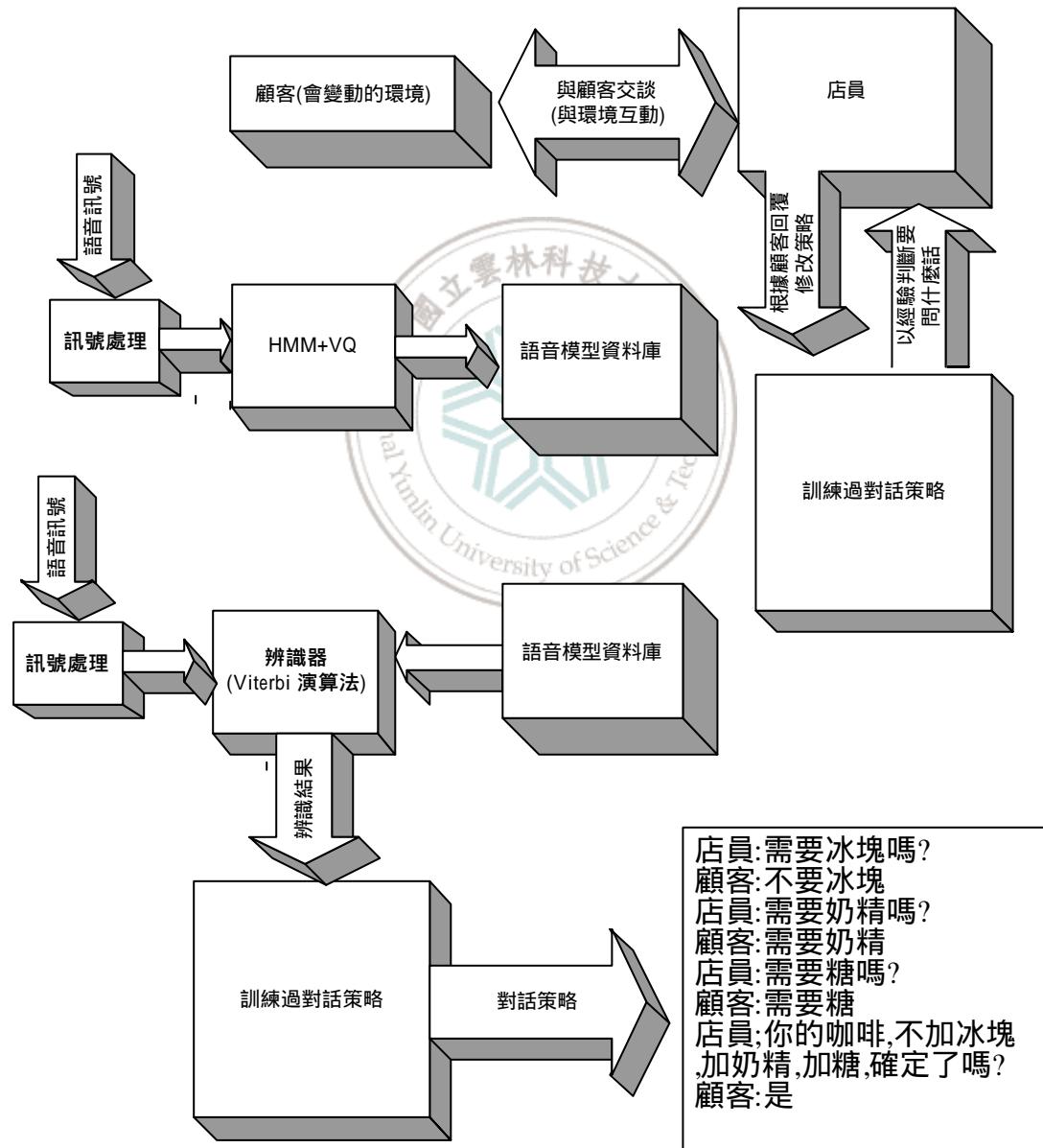


圖 5.1 詳細的系統方塊圖

訓練對話策略的程式是用 R (類似 S-plus) 開發，語音辨識程式的部分，是

用 VC 及 VB 開發的。我們目前的系統是將語音訓練和語音辨識和對話策略整合在一起，除了訓練對話策略的功能以外，下面是使用流程：

- 1 建立：要先把對話策略訓練好並且建立使用者語音模型資料庫。對話策略是由訓練對話策略的程式，產生經驗檔（較佳的對話策略）。語音模型資料庫是要用本系統來產生。使用者要先錄音，再訓練語音模型，如此一來，使用者就建立了使用者語音模型資料庫。
2. 辨識：本系統把對話策略的經驗檔載入進來。使用者就可以扮演顧客的角色以語音的輸入和電腦（店員）對話。如下圖是執行結果。

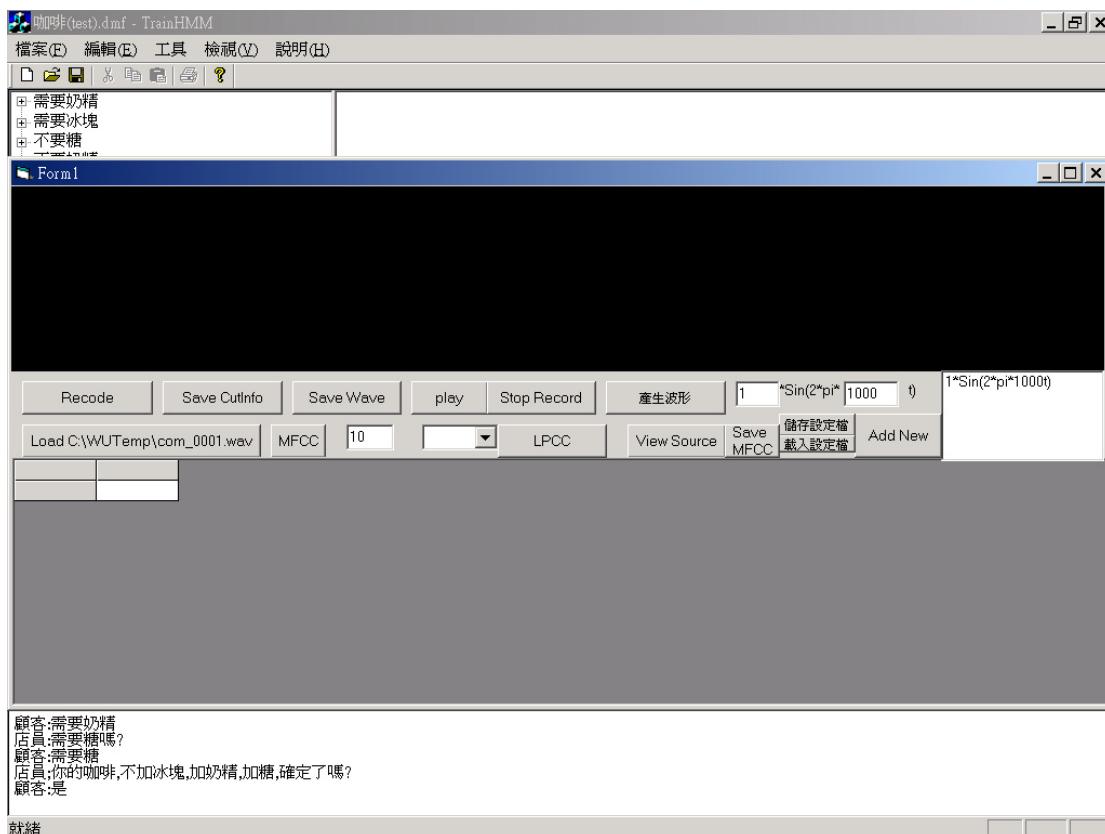


圖 5.2 執行結果

## 5.2 討論

### 5.2.1 語音的問題

在第三章我們運用了梅爾頻譜分析的語音特徵來取代「語音辨識與語言模型」的「語音分段處理」以及「特徵參數抽取」的部分。很明顯的，從前面的實驗數據中我們得到了明顯改善的證明，如表 5.1。這張表是論文「語音辨識與語言模型」的辨識器（使用 LPCC）與本論文（使用 MFCC）的辨識器的辨識率比較。樣本來源：「中央研究院的 MAT-2000-Edu 語料庫」是要說明不同人之間的辨識率。樣本來源：「吾人」是要說明相同人之間的辨識率。MFCC 在相同人之

間的辨識率與不同人之間的辨識率都有比 LPCC 好表現。

表 5.1 辨識率比較

語音辨識與語言模型： 辨識器直接判斷，使用 LPCC	本論文：使用 MFCC	
68%	樣本來源；中央研究院的 MAT-2000-Edu 語料庫	樣本來源；吾人
	74%	78%

在前面「文獻探討」的部分，我們提到「語音辨識與語言模型」的語音辨識部分並沒有做好濾除雜訊的工作，在梅爾頻譜分析中，我們利用傅立葉轉換，然後經由濾波器組，達到濾除雜訊的目的，其辨識率已經改善。

### 5.2.2 對話策略

NJFun system 是以對於一個問題會採系統主導或使用者主導的方式來變化出很多種類的對話策略，讓使用者評分。當很多使用者使用完畢該系統之後，其對話策略，就會變成大眾化的對話策略。NJFun 能夠應付系統主導型、使用者主導型、混合型的對話，而產生令大部分使用者都滿意的對話策略。本論文訓練對話策略的目的，是要讓電腦習得避免一些不合宜的對話。這兩個目的是截然不同的。一個要收集對話，用人工去區分出不合宜的對話。另外一個是要收集對話，讓電腦學習到在某個狀態中，哪些對話是不合宜的，當電腦遇到這個狀態，要避免這些對話。

我們從前面所述的實驗得到下列的結論：這個對話系統會由於顧客給的回饋而學得避免掉重複詢問的對話，而且也可以學習到什麼時候是確認的時機。我們顧客用的主動和反覆的機率值來模擬顧客的行為，事實上這是包括大部分的顧客類型。對於配合回答型的顧客而言，該對話方式就會變系統主導型的對話方式。因為顧客都有針對店員的問題回答。對於主動型的顧客而言，該對話方式就會變使用者主導型（在這裡應該說，顧客主導型。因為與店員對話的角色是顧客）的對話方式。因為顧客有時候會針對店員的問題回答，但有的時候說出自己想要的。對於反覆的顧客會對話方式就會變成混合型（系統主導型和顧客主導型），因為我們的對話系統最後會給顧客反悔機會，顧客反悔的時候，對於被動回答型的顧客而言，就必須說出自己想要的。這是系統主導型的對話方式；至於主動型顧客會說出全部配料，這是顧客主導型的對話方式。

### 5.3 未來和展望

這個系統可以有很多地方可以加強的。語音辨識的部分有底下規劃：

1. 在語音模型的產生的方式是收集語音樣本、再做訓練的工作。然而這樣子實在很麻煩。所以希望能夠以自動切割的方式，來訓練語音模型。針對此點吾人想到的是既然可以濾除雜訊，那麼以濾除的雜訊位置來做切割，這是比較簡單了方法。
2. 雖然我們已經可以濾除雜訊，但是現在我們的元件只能辨識詞語而已，我們希望最終是能夠將母音、子音分離出來、甚至將背景聲音去除。針對此點吾人想到的是如果第一個問題可以解決的話，那麼第二個只是只是短時間的語音辨識而已。我們將切割好的語音訊號放入做分析找出相同頻率的，因為母音會比較多重複的地方。

此外，有關對話策略的部分我們希望將原本的對話內容能夠再擴大，讓店員會講更多話。負責更多點餐的菜色。如此一來，就會比較像真的店員。我們也希望將本對話系統與 NJFun 做法結合，本對話系統將收集的對話做分類，避免重覆的對話，然後再循著 NJFun 做法給使用者評分，學習到大眾化的對話策略。



## 參考文獻

- [1] Sutton, R. S. and Barot, A. G. (1998). Reinforcement Learning: An Introduction. MIT Press.
- [2] Singh, S., Litman, D., Kearns, M., Walker, M. (2002). Optimizing dialogue management with reinforcement learning: Experiments with NJFun system. Journal of Artificial Intelligence Research, 16, 105-133.
- [3] Claudio Becchetti and Lucio Prina Ricotti (1999). Speech Recognition Theory and C++ Implementation. JOHN WILEY & SONS, LTD Inc. New York USA.
- [4] 陳智聰，民國 91 年，語音辨識與語言模型，國立雲林科技大學電子與資訊工程研究所碩士論文。
- [5] A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, L. R. Rabiner, *Proc. IEEE*, Vol. 77, No. 2, pp. 257-286, February 1989
- [6] Lai, Jim Z.C. and Lue, C.C., Fast Search Algorithms for VQ Codebook Generation JVCIR(7), No. 2, PP163-168, 1996.
- [7] Qiang Huo and Chorkin Chan, Contextual Vector Quantization for Speech Recognition with Discrete Hidden Markov Model. Speech, Image Processing and Neural Networks, Proceedings of ISSIPNN 1994, Page(s): 698 -701 vol.2.
- [8] L. R. Rabiner / R. W. Schafer Digital Processing of Speech Signals Prentice-Hall, Inc., Englewood Cliffs, New Jersey 076322
- [9] Stuart Russell and Peter Norvig Artificial Intelligence A Modern Approach (2e), Pearson Education International, 2003