

# 國立交通大學

## 資訊工程系

### 碩士論文

利用多觀察值型隱馬可夫模型進行人體動作辨識



Human Motion Classification by Hidden Markov Model  
with Multiple Feature Streams

研究生：顧正偉

指導教授：陳 稔 博士

中華民國九十四年六月

利用多觀察值型隱馬可夫模型進行人體動作辨識

Human Motion Classification by Hidden Markov

Model with Multiple Feature Streams

研究生：顧正偉

Student : Cheng-Wei Ku

指導教授：陳 稔 博士

Advisor : Dr. Zen Chen

國立交通大學

資訊工程系

碩士論文

A Thesis

Submitted to Department of Computer Science and Information Engineering

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

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

# 利用多觀察值型隱馬可夫模型進行人體動作辨識

學生： 顧正偉

指導教授： 陳稔 博士

國立交通大學資訊工程學系

## 摘要

本論文的主要目的，在於對人體的動作進行辨識，以期望電腦能以較為有意義的方式，來描述動作。

本論文試著利用「隱馬可夫模型」(Hidden Markov Model)的一種變形—「多觀察值型隱馬可夫模型」(Hidden Markov Model with Multiple Feature Streams)來對於動作進行辨識。利用多觀察值型隱馬可夫模型，我們對於動作的分析，可以同時考慮到不同、獨立的特徵，而不像傳統的隱馬可夫模型僅能考慮單一特徵；因此，HMM/MFS可以處理更為複雜的資料。此外，對於因為雜訊而使理想模型時無法辨識的資料，本論文也提出了三種不同方法來解決這些問題。

在實驗的部份，本論文使用實際的二維資料和模擬的三維資料來進行實驗。在二維影像的部份，是使用單一固定視角的二維影像序列作為主要的資料，總共有八種不同的動作；這一部分是由影像中擷取出人體形狀的水平、垂直投影以及長寬比例來當作三個特徵，在對這三組特徵進行分群(分群數目由 3~20，共取四組)轉換為符號後，先利用 HMM/MFS 訓練並建立模型，然後再拿所建立的模型來進行辨識。而三維的模擬資料實驗中，本論文試著模擬由人體動作的三維資訊來切割基礎動作，並以此作為辨識的依據；實際上拿來測試的動作有兩種，於理想的動作序列加入雜訊後，再拿來利用 HMM/MFS 進行訓練、辨識。

# Human Motion Classification by Hidden Markov Model with Multiple Feature Streams

Student: Cheng-Wei Ku

Advisor: Dr. Zen Chen

Department of Computer Science and Information Engineering  
National Chiao Tung University

## Abstract

The main purpose of this thesis is to develop a method of analyzing and classifying a set of primitive human motions from image sequences using a computer. Hopefully the computer can be made more effective to visually percept and interpret the human motion.

In this thesis we use an extended Hidden Markov Model (HMM), called an HMM with multiple feature streams (HMM/MFS) to classify the human motions. The traditional HMM only allows a single observation stream, while in our HMM/MFS multiple observation streams are allowed. With these multiple observation streams we can analyze the complex human motions. More importantly, we modify the HMM/MFS to deal with the human motion data corrupted by noise or the imperfect human motion data, which cannot be handled by the standard model. We propose three different ways for handling the corrupted or imperfect cases.

In the experiments the proposed HMM/MFS is applied for the classification of 2D and 3D human motion data. The 2D motion image data are obtained by shooting a person performing some motion by a camera mounting on a still tripod. There are eight different types of these motions. Three observation streams are extracted from the motion image data. Each observation stream consists of 3 to 20 possible labels which describe the characteristics of the selected feature value. The HMM/MFS is first trained and, then, used to classify these types of human motions. The 3D motion human data are in the form of symbolic motion label sequence, each label categorizing a basic motion unit based on some 3D motion data clustering technique. There are two kinds of motion label sequences, each corresponding to a motion at a higher semantic level than the basic motion unit. Again, the HMM/MFS is trained and, then, used to classify these different kinds of motions. The considered label sequences may contain noise and have different sequence lengths.

# 誌謝

首先感謝 陳稔教授兩年來的指導，使我能學到良好的研究方法以及獨立思考、研究的能力；而在研究遇到瓶頸時，老師總會給予幫助，和我一起討論、思考，並尋求解決方法。這篇論文能夠順利完成，在此必須向老師獻上最誠摯謝意。

此外，也要感謝 PAIS 的全體成員。方彥翔學長在學習與研究的過程中，常常給予各種意見與協助；而紀文原同學、吳昶浩學弟和陳柏璵學弟在論文的實驗以及理論的推導上，也都給予了非常大的協助。

最後，要感謝父母親，由於他們無怨無悔的付出，我才能在不需顧慮到物質生活上的問題的情形下，全心全力的投入在學術上；同時，他們也是使我努力向學的最大動力。而也要感謝其其他的家人、親戚，他們支持，也都是我能順利走到這一步的助力。



當然，還有不少人對於這篇論文，也都有有形或無形的幫助，每個都是此篇論文能完成的關鍵；再此無法一一致謝，故僅以本篇論文獻給他們，表達我的感謝之意。

顧正偉 於交大

June, 2005

# 目錄

摘要 .....	i
Abstract.....	ii
誌謝 .....	iii
目錄 .....	iv
表索引 .....	vi
圖索引 .....	vii
一. 緒論 .....	1
1.1. 問題描述 .....	1
1.2. 相關研究 .....	2
1.2.1. 資料類型 .....	2
1.2.2. 人體動作辨識的方法 .....	3
1.2.3. Hidden Markov Model的沿革 .....	3
1.3. 研究內容與論文貢獻 .....	5
1.3.1. 資料來源 .....	5
1.3.2. 辨識方法 .....	5
1.3.3. 論文貢獻 .....	7
1.4. 論文架構 .....	8
二. 標準型隱馬可夫模型 .....	9
2.1. HMM的基本元素 .....	10
2.2. HMM的三個主要問題 .....	11
2.2.1. 機率計算 .....	11
2.2.2. 最佳狀態轉換序列 .....	16
2.2.3. 調整模型參數 .....	17
2.3. HMM實做的問題 .....	21
2.3.1. 起始參數估計 .....	21
2.3.2. 訓練資料 .....	21
2.3.3. Scaling Problem.....	23
三. 多觀察值型隱馬可夫模型的介紹 .....	27
3.1. HMM/MFS的修改 .....	29
3.2. 觀察值的權重 .....	31
3.2.1. 所有狀態單一權重 .....	31
3.2.2. 各狀態獨立權重 .....	32
3.2.3. 權重的計算 .....	33
四. 雜訊干擾下的資料辨識 .....	34

4.1.	理想模型無法辨識的原因 .....	34
4.2.	無法完全對應狀態的情況 .....	35
4.2.1.	解決方法一：拋棄無法對應的部份觀察值 .....	36
4.2.2.	解決方法二：拋棄權重小的部份觀察值 .....	37
4.3.	無法做狀態轉換的情況 .....	40
4.3.1.	解決方法：忽略無狀態轉移可能性的時刻 .....	41
五.	實驗與結果 .....	42
5.1.	2D實際資料試驗 .....	42
5.1.1.	實驗設計 .....	42
5.1.2.	實驗資料 .....	50
5.1.3.	實驗結果 .....	57
5.2.	3D模擬資料試驗 .....	63
5.2.1.	實驗設計、資料 .....	63
5.2.2.	實驗結果 .....	65
六.	結論與未來工作 .....	67
6.1.	結論 .....	67
6.2.	未來工作 .....	67
	參考文獻 .....	70



# 表索引

表格 1 動作數目列表 .....	50
表格 2 動作一 觀察值序列列表 .....	54
表格 3 動作二 觀察值序列列表 .....	54
表格 4 動作三 觀察值序列列表 .....	54
表格 5 動作四 觀察值序列列表 .....	55
表格 6 動作五 觀察值序列列表 .....	55
表格 7 動作六 觀察值序列列表 .....	55
表格 8 動作七 觀察值序列列表 .....	56
表格 9 動作八 觀察值序列列表 .....	56
表格 10 辨識結果整體簡表 .....	59
表格 11 訓練資料 5 筆，分群數目 3, 7, 7 .....	60
表格 12 訓練資料 5 筆，分群數目 5, 10, 10 .....	60
表格 13 訓練資料 5 筆，分群數目 7, 15, 15 .....	60
表格 14 訓練資料 5 筆，分群數目 10, 20, 20 .....	60
表格 15 訓練資料 10 筆，分群數目 3, 7, 7 .....	60
表格 16 訓練資料 10 筆，分群數目 5, 10, 10 .....	61
表格 17 訓練資料 10 筆，分群數目 7, 15, 15 .....	61
表格 18 訓練資料 10 筆，分群數目 10, 20, 20 .....	61
表格 19 訓練資料 20 筆，分群數目 3, 7, 7 .....	61
表格 20 訓練資料 20 筆，分群數目 5, 10, 10 .....	61
表格 21 訓練資料 20 筆，分群數目 7, 15, 15 .....	62
表格 22 訓練資料 20 筆，分群數目 10, 20, 20 .....	62
表格 23 3D動作模擬測試結果 .....	65

# 圖索引

圖 1-1 訓練與辨識流程圖.....	6
圖 2-1 Forward Algorithm 示意圖.....	13
圖 2-2 Forward 機率計算示意圖.....	14
圖 2-3 Backward Algorithm 示意圖.....	15
圖 2-4 Baum–Welch Algorithm 流程圖.....	18
圖 2-5 $\xi_t(i, j)$ 示意圖.....	19
圖 2-6 多訓練資料合併示意圖.....	22
圖 2-7 利用 Null State 的 A 代表 $\pi$ .....	23
圖 4-1 無狀態轉換示意圖.....	40
圖 5-1 前置處理流程圖.....	43
圖 5-2 去背景例圖.....	43
圖 5-3 擷取出的人形.....	44
圖 5-4 投影累積量例圖.....	45
圖 5-5 動作一 彎腰.....	50
圖 5-6 動作二 彎腰起身.....	51
圖 5-7 動作三 蹲下.....	51
圖 5-8 動作四 蹲下起身.....	51
圖 5-9 動作五 踢腿.....	52
圖 5-10 動作六 出拳.....	52
圖 5-11 動作七 上勾拳.....	53
圖 5-12 動作八 正步.....	53
圖 5-13 不同訓練資料數目與分群數目下的辨識正確率.....	57
圖 5-14 不同訓練資料數目與分群數目下的無法辨識機率.....	57
圖 5-15 以容錯可能性辨識的最佳結果.....	58

# 一. 緒論

## 1.1. 問題描述

動作辨識 (Action Recognition) 的主要目的，是希望能讓電腦用比較語意的方式，來告訴我們現在的動作是在做什麼；可以應用在監控系統、影像自動分析、人機介面…等方面。

在監控系統上進行動作的辨識，讓電腦能夠自動判斷出在觀察到的影像中的人的行為，並能對此做出處理或提出警告。而將動作辨識應用在影像自動分析上，可以做到影片的自動索引、加註或做摘要[7][16]。於人機介面上（如手寫、手語辨識），可以藉由動作來操作系統，不再需要使用鍵盤或滑鼠[12][17]。

要進行動作辨識，實際上就是將整個動作看成一種隨著時間變化的時序資料 (time-varying data) 來進行分析。大部分的方法中，都是把這些時序資料拿來做分類 (classification) 的動作；就是先設定好幾種已知的動作資料，而新的動作資料，則以資料對模型的可能性來決定它是哪一種動作。

## 1.2. 相關研究

### 1.2.1. 資料類型

動作辨識的主要方法，就是將時序資料拿來做分類；而時序資料的來源，主要則是分成 2D 和 3D 兩種：

#### 1. 利用 2D 資訊[7][10][11]

利用在單一視角上的人體形狀來當作主要的資訊來源。

像以論文 “Recognizing Human Action in Time-Sequential Images Using

Hidden Markov Model” [7]的方法，就是針對單一視角的網球影片，去擷取出特徵 ( Mesh feature )，並以特徵向量 ( feature vector ) 的變化來做分類的依據。

這樣的優點在於只需要一台攝影機，設備成本低、架設簡單。缺點則是大多只能處理特定視角，而且能得到的資訊實際上算是相當有限；此外，對於被遮蔽的部份，很難有良好的解決方法，所以大多只能分別差異比較大的動作。

#### 2. 利用 3D 資訊[6][8][9]

先利用動作捕捉 ( motion capture ) 的技術，將人體的各部位的 3D 位置找到，並以此為主要的資訊。

以論文 “A Model-Based Human Motion Analysis System in Multiple-Views”

[6]來說，就是先使用 Model Base 的方法先去找出人體的所有關節角度，然後以這些角度構成的向量來做為分類的資料。

這樣做的優點則是可以利用多視角的特點，降低遮蔽等問題，對於姿勢的判定，可以得到的資訊也更多。但是主要問題在於資料的取得；要取得人體各部位的 3D 位置，大多必須要使用多視角的攝影機或其他設備，不但在架設、計算上較為複雜，成本也相對提高許多。

## 1.2.2. 人體動作辨識的方法

在取得資料後，接著要做的，就是對取得的資料進行辨識的動作。目前一般對於動作的辨識，主要可以分為下面兩類[1][2]：

### 1. 樣本比對(Template matching) [11]

為每一種動作去建立他的標準範本，在要辨識時，就拿要辨識動作去和各範本比較。

論文“Real-Time Recognition of Activity Using Temporal Templates” [11]所使用的方法，就是使用他們所定義的MEI (motion energy image) 和MHI (motion history images) 來當作動作的樣本。

他的優點是計算簡單，也容易實做；缺點則是容易受到動作雜訊或時間差異干擾。同時，他對於不同視角的動作也很難做處理。

### 2. State-Space Approaches[6][7][8][9][10]

將每個靜態的姿勢定義為一個狀態 (state)，並利用各狀態之間的轉移，來描述一個連續的動作。這類的方法中，最常被拿來使用的就是以隱馬可夫模型 (Hidden Markov Model，簡稱HMM) 為基礎所衍生出的各種方法。

這類的方法的優點在於可以克服動作雜訊或時間差異的干擾；但是相對起來，他的計算量就較為複雜，而狀態數目與特徵的選取，也是很大的問題。

在此，我們選擇以第二個方法中的 HMM 為基礎來處理我們的問題。

## 1.2.3. Hidden Markov Model 的沿革

Hidden Markov Model[3][4][12][13][14][15] (HMM) 是由Markov Model衍伸而來的。

Markov Model 是一種利用狀態的轉移機率，來描述一個系統運作的隨機程序 (stochastic process)；每個狀態都對應到實際可以觀察到的事件、現象；而對於一個系統，就利用這些狀態的轉換來描述。而在狀態轉移時，則利用機率的觀念，讓他的轉移可以比較自由；而在考慮狀態轉移的機率時，每一刻的狀態只會和上一刻的狀態

有關係，和其他更早的資訊都不相關，這也就是「Markov Property」。

但是由於在 Markov Model 中，每一個狀態都是對應到特定的事件、現象，所以他的應用其實相當的限制性；因此，就衍生出 Hidden Markov Model。在 HMM 中，將現象、事件這種可以直接觀測到的資訊稱為「觀察值」(observations)；同時，也把狀態和觀察值分開，觀察值對應到現實中的現象，而將狀態視為無法觀察到的部份，形成了雙重內嵌隨機程序 (double embedded stochastic process)。而狀態無法被觀察到的這個特性，也就是 HMM 中 hidden 的觀念。HMM 利用無法觀察到的狀態轉移，以及狀態呈現的觀察值來描述一個系統；而藉著這種 hidden 的觀念，使的 HMM 的應用範圍大為提升。而由於將觀察值和狀態分開，也使的 HMM 對於可以處理的資料類型更為廣泛。



## 1.3. 研究內容與論文貢獻

### 1.3.1. 資料來源

對於 Hidden Markov Model 這一類的動作辨識方法來說，所需用以訓練、辨識的資料，只要是能代表動作地觀察值序列即可；而這種觀察值的取得，則可以由 2D 或 3D 取得，對於辨識方法本身的差異度並不大。

由於大部分的作法中，都是使用離散型的隱馬可夫模型 (Discrete Hidden Markov Model)，所以如果特徵本身就是離散的符號序列 (例如基礎動作或姿勢的名稱或代號) 的話，可以直接利用 HMM 來處理；但是如果特徵資料是連續性的數值 (如位置、角度)，則必須在進行訓練或辨識前，先將資料做分群或分類，轉換為離散的符號序列。

由於 3D 動作的資料取得較為困難，因此本論文的實際數據實驗是以 2D 的影像序列來進行；而對於 3D 的動作辨識，則是以模擬的資料來進行簡單的初步測試。

### 1.3.2. 辨識方法

而在訓練、辨識的部份，主要使用 Hidden Markov Model 的一種變形—多觀察值型隱馬可夫模型 (Hidden Markov Model with Multiple Feature Streams, 簡稱為 HMM/MFS) [15] 進行對動作的訓練與辨識。而接受的資料類型，也是使用離散 (Discrete) 的符號；因此如果是連續性的數值資料，必須要先轉換為離散的符號。

傳統的 HMM 只能夠接受一組觀察值序列，對於有多組特徵的時候，大多是把多組特徵怎合成一個特徵向量 (feature vector)，然後利用 Vector Quantization 之類的方法，將他轉換為單一的特徵序列，並以這個特徵序列當作觀察值序列供 HMM 進行訓練、辨識。但是這樣的缺點就是無法將各觀察值分開考慮，是必須全部一起考慮的。

而 HMM/MFS 和一般的 HMM 的主要差異，就在於他能同時接受多組獨立觀察值序列。因此，對於每個觀察值可以以各自獨立的方式去做計算；而如果再對觀察值加入權重後，也可以分別定義各觀察值的重要性，使的整個模型能有更好的通用性。

整個辨識的流程大致如下：

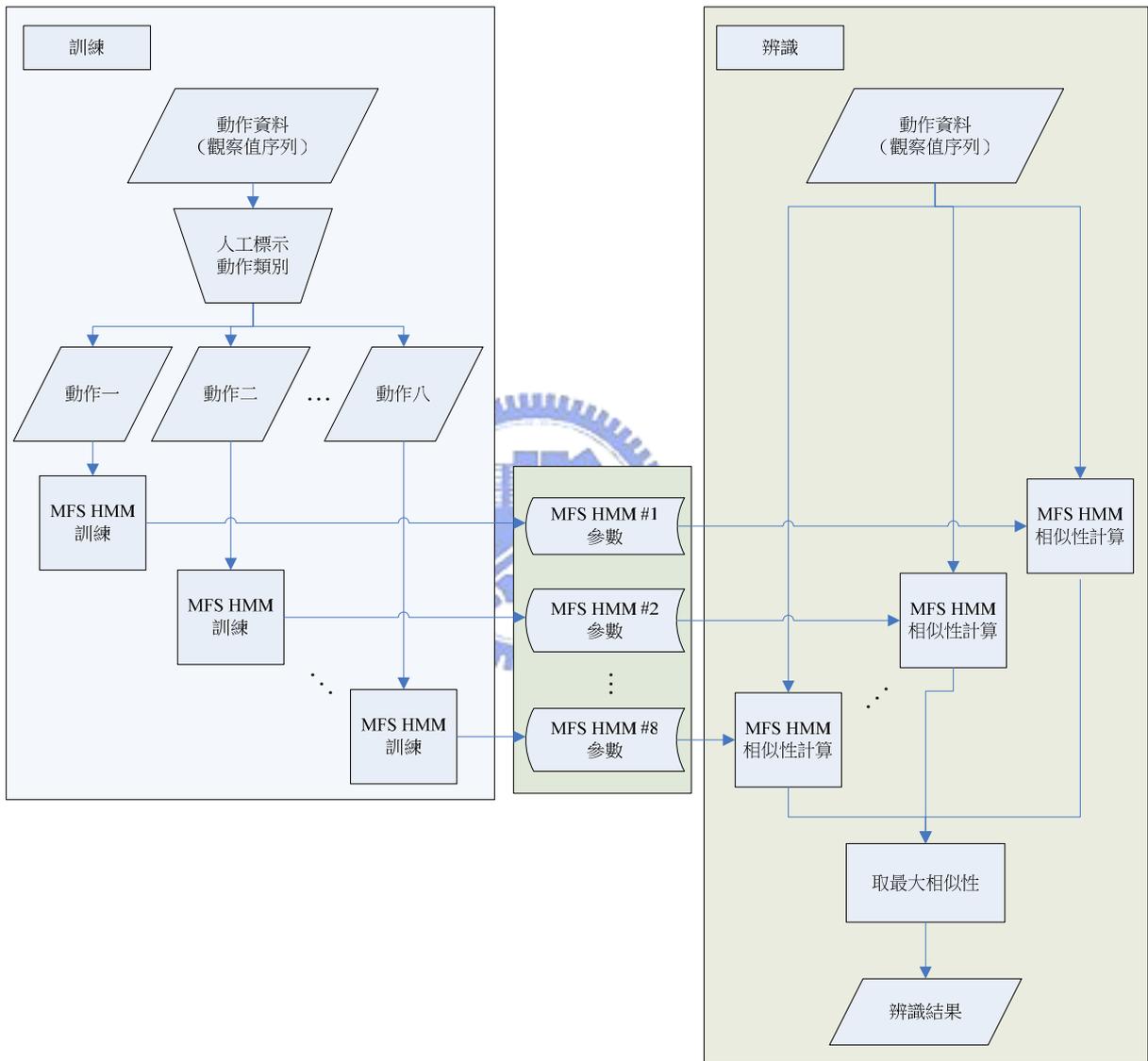


圖 1-1 訓練與辨識流程圖

這一部分為本論文的主要理論部份，將於第二章、第三章對於現有的模型的介紹。

而在實際使用多觀察值型隱馬可夫模型時，則發現不少資料會有「無法辨識」的情形；本論文也對此提出了一些解決方法，去對這些資料做特別的處理。這一部分將

於第四章做說明。

### 1.3.3. 論文貢獻

本論文主要的貢獻，在於提供了將多觀察值型隱馬可夫模型套用在動作辨識這方面的概念；我們在動作的辨識中，不再只能同時考慮整體的狀態，而能同時使用多組不同的獨立特徵來當作觀察值，並考慮不同特徵的重要性。

此外，本論文也對使用在使用多觀察值型隱馬可夫模型時，可能發生的「無法辨識」的情形，提出了幾種「容錯可能性」的計算方法，來解決這個問題、增加辨識的正確性。



## 1.4. 論文架構

本論文第一章為緒論，先大致說明本研究的主題、其他的相關研究以及本論文的內容、方法。

第二章、第三章主要在說明本論文使用的模型，多觀察值隱型馬科夫模型 (Hidden Markov Model with Multiple Feature Streams) 的基本理論。第二章先對標準型的 Hidden Markov Model 做說明，以便先了解 HMM 的基本架構與概念；然後再於第三章對多觀察值隱馬科夫模型做說明，比較它和標準型 HMM 的差異。

第四章為本論文利用「多觀察值」的特性，對於多觀察值型隱馬可夫模型主要的修改，也就是「雜訊干擾下的資料辨識」的部份。

第五章則是本論文實驗的說明，包含 2D 的實際資料和 3D 的模擬資料兩部份。

第六章提出本論文的結論，並列出可以改進以及未來可能進行的研究方向。



## 二. 標準型隱馬可夫模型

隱馬可夫模型 (Hidden Markov Model, HMM) 是一個常常被拿來做時間序列資料的分析、辨識的模型[1][4]。目前在語言、手寫文字、手勢的辨識上，都被使用的非常廣泛[4][12][15]；而在動作分析上，也常常被拿來使用[7][8]。

HMM 的基本設計，就是由內部「狀態」(states)之間的轉換以及各狀態呈現的「觀察值」(observations)來描述整個資料的模型。而實際上，所能得到的資訊只有觀察值，而無法知道內部狀態的轉換的情況；這就是 HMM 中 hidden 的觀念。利用這個 hidden 的觀念，HMM 更能夠接受觀察資料的變化、誤差，讓 HMM 本身的通用性更高。

本章主要參考“A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition” [4]的內容，來對HMM做說明。



## 2.1. HMM 的基本元素

HMM 是用多個狀態所構成的，而對於模型的描述，則是透過狀態之間的轉換以及各狀態呈現的觀察值來描述。因此，在 HMM 中擁有下列的主要元素：

1.  $S = \{S_1, S_2, \dots, S_N\}$  為狀態的集合，裡面的  $S_j$  為狀態(state)， $j = 1, 2, \dots, N$ 。
2.  $V = \{v_1, v_2, \dots, v_K\}$  為所有可能觀察到的值的集合，其中  $v_k$  為可觀察到的值， $k = 1, 2, \dots, K$ 。
3.  $q_t$  於時間  $t$  時的狀態， $t = 1, 2, \dots, T$ ， $q_t \in S$ 。
4.  $O_t$  於時間  $t$  的觀察值， $O_t \in V$ 。 $O = O_1 O_2 \dots O_T$  代表由  $t = 1$  到  $t = T$  的觀察值序列。
5.  $\pi$ ，起始狀態機率矩陣， $\pi = \{\pi_i\}$ ， $i = 1, 2, \dots, N$ ； $\pi_i = P(q_1 = S_i)$ ，代表  $t = 1$  時，狀態是  $S_i$  的機率，所以  $0 \leq \pi_i \leq 1$ ， $\sum_{i=1}^N \pi_i = 1$ 。
6.  $A$ ，狀態轉換的機率矩陣， $A = \{a_{ij}\}$ ， $i = 1, 2, \dots, N$ ， $j = 1, 2, \dots, N$ ； $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$  代表由狀態  $i$  轉變到狀態  $j$  的機率，所以  $0 \leq a_{ij} \leq 1$ ， $\sum_{j=1}^N a_{ij} = 1$ 。
7.  $B$ ，特定狀態下的觀察值分佈機率矩陣， $B = \{b_j(v_k)\}$ ， $j = 1, 2, \dots, N$ ， $k = 1, 2, \dots, K$ ； $b_j(v_k) = P(O_t = v_k | q_t = S_j)$  代表在狀態  $j$  的情形下，觀察值為  $v_k$  的機率，所以  $0 \leq b_j(v_k) \leq 1$ ， $\sum_{k=1}^K b_j(v_k) = 1$ 。

而在一般情況下，會把  $\pi$ 、 $A$ 、 $B$  這三個機率矩陣合在一起，稱為 HMM 的參數  $\lambda$ ， $\lambda = (\pi, A, B)$ ；而實際上，只要有  $\lambda$ ，就足以代表一個完整的 HMM 模型了。

## 2.2. HMM 的三個主要問題

在使用 HMM 時，會有三個主要的問題，分別是：

### 1. 機率計算

計算在給定  $\lambda = (\pi, A, B)$  的情形下， $O = O_1 O_2 \dots O_T$  這組觀察值序列發生的機率  $P(O | \lambda)$ ；也就是要去計算  $O$  這組觀察值序列對由  $\lambda$  構成的 HMM 模型的可能性。

### 1. 最佳狀態轉換序列

在已知  $\lambda = (\pi, A, B)$  和觀察值序列  $O = O_1 O_2 \dots O_T$  的情形下，去推算出一組最有可能的狀態轉移路徑  $Q = q_1 q_2 \dots q_T$ 。

### 2. 調整模型參數

根據觀察值序列  $O = O_1 O_2 \dots O_T$  調整 HMM 的參數  $\lambda = (\pi, A, B)$ ，讓  $P(O | \lambda)$  的值越大越好；也就是要讓 HMM 模型最能描述觀察值序列  $O$ 。

這也就是訓練 HMM 時所要使用的方法。



### 2.2.1. 機率計算

這個問題是要計算在給定  $\lambda = (\pi, A, B)$  的情形下， $O = O_1 O_2 \dots O_T$  這組觀察值序列發生的機率  $P(O | \lambda)$ 。

先假設觀察值序列  $O$  的狀態轉換序列為  $Q = q_1 q_2 \dots q_T$ ，如此我們先計算出下列的機率：

$$\begin{aligned} P(O | Q, \lambda) &= \prod_{i=1}^T P(O_i | q_i, \lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \dots b_{q_T}(O_T) \\ P(Q | \lambda) &= \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \\ P(O, Q | \lambda) &= P(O | Q, \lambda) P(Q | \lambda) \\ &= \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) a_{q_2 q_3} \dots a_{q_{T-1} q_T} b_{q_T}(O_T) \end{aligned} \tag{2.1}$$

而  $P(O | \lambda)$  就是考慮所有可能的  $Q$  的結果：

$$\begin{aligned}
P(O | \lambda) &= \sum_{\text{all } Q} P(O, Q | \lambda) \\
&= \sum_{\text{all } Q} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) a_{q_2 q_3} \cdots a_{q_{T-1} q_T} b_{q_T}(O_T)
\end{aligned} \tag{2.2}$$

如果直接對於所有的可能都直接各別計算的話，會有很多重複計算的部份；因此，要比較有效率的計算  $P(O | \lambda)$  的值，可以使用 Forward Algorithm 或 Backward Algorithm 這兩個演算法來達成。

## Forward Algorithm

Forward Algorithm 是透過 forward variable  $\alpha_t(i)$ ，由前往後，推算出觀察值序列  $O$  在參數  $\lambda$  下的發生機率。

$\alpha_t(i)$  定義為「給定參數  $\lambda$ ，在時間  $t$  時，狀態為  $S_i$ ，且在時間  $t$  之前的觀察值序列為  $O_1 O_2 \dots O_t$  的機率」；也就是：

$$\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = S_i | \lambda) \tag{2.3}$$

而實際在計算時的計算方式如下：

[1] Initialization :

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \tag{2.4}$$

[2] Induction :

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1, \quad 1 \leq i \leq N \tag{2.5}$$

[3] Termination :

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i) \tag{2.6}$$

在步驟 [1] 中，先計算所有狀態在時間  $t=1$  時的機率  $\alpha_1(i) = P(O_1, q_1 = S_i | \lambda)$ ，也就是  $\alpha_1(i) = \pi_i b_i(O_1)$ 。而在步驟 [2]，則是以時間  $t$  的  $\alpha_t(i)$ ，來計算下一刻，時間  $t+1$  的  $\alpha_{t+1}(j)$ ；因為對於  $\alpha_{t+1}(j) = P(O_1 O_2 \dots O_t O_{t+1}, q_{t+1} = S_j | \lambda)$  來說，並不知道時間  $t$  時的狀態 ( $q_t$ )，所以在計算時，必須要考慮所有的狀態，也就是：

$$\begin{aligned}
\alpha_{t+1}(j) &= P(O_1 O_2 \dots O_t O_{t+1}, q_{t+1} = S_j \mid \lambda) \\
&= \left[ \sum_{i=1}^N (P(O_1 O_2 \dots O_t, q_t = S_i \mid \lambda) P(q_{t+1} = j \mid q_t = i, \lambda)) \right] P(O_{t+1} \mid q_{t+1} = S_j, \lambda) \\
&= \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1})
\end{aligned} \tag{2.7}$$

也就如下圖表示：

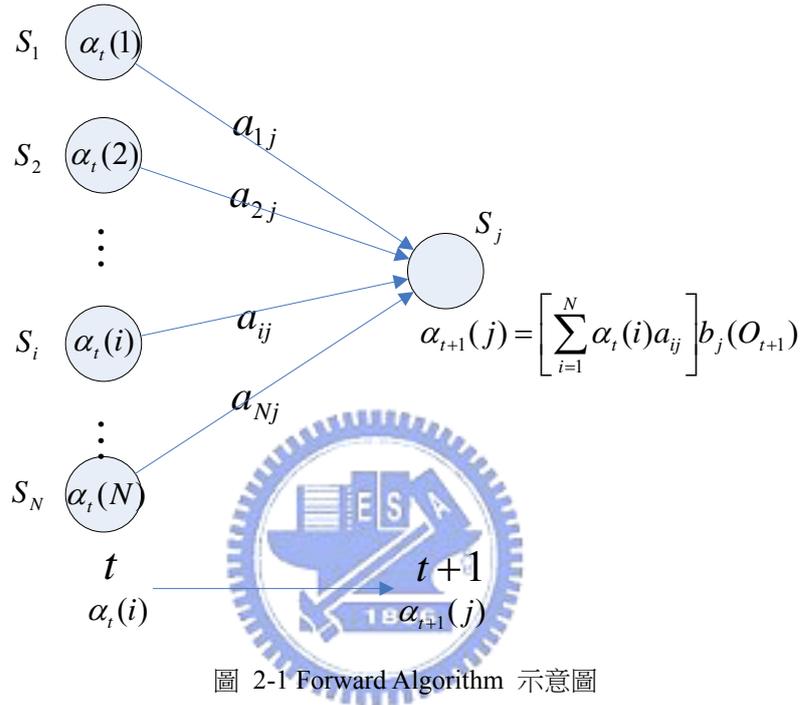


圖 2-1 Forward Algorithm 示意圖

而在步驟 [3] 則是在將所有的  $\alpha_t(i)$  計算出來後，對於所有的可能，計算出  $P(O \mid \lambda)$  的機率：

$$\begin{aligned}
P(O \mid \lambda) &= \sum_{i=1}^N P(O, q_T = S_i \mid \lambda) \\
&= \sum_{i=1}^N P(O_1 O_2 \dots O_T, q_T = S_i \mid \lambda) \\
&= \sum_{i=1}^N \alpha_T(i)
\end{aligned} \tag{2.8}$$

如下圖表示：

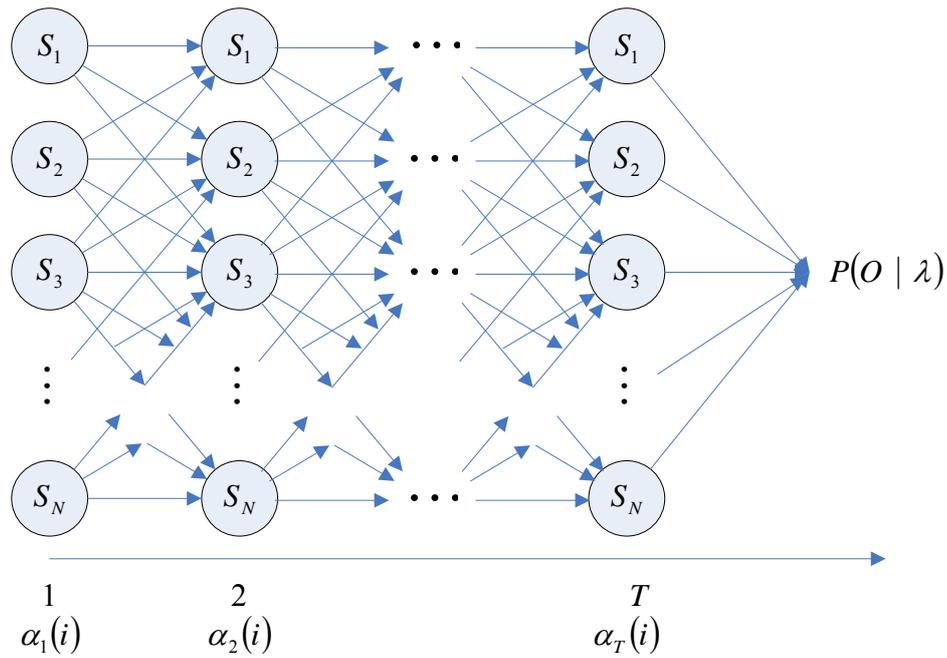


圖 2-2 Forward 機率計算示意圖

## Backward Algorithm

Backward Algorithm 則是透過 backward variable  $\beta_t(i)$ ，由後往前，推算出觀察值序列  $O$  在參數  $\lambda$  下的發生機率。 $\beta_t(i)$  的定義是「在給定參數  $\lambda$ 、時間  $t$  的狀態是  $S_i$  的情況下，在時間  $t$  之後的觀察值為  $O_{t+1}O_{t+2}\dots O_T$  的機率」；他的數學式為：

$$\beta_t(i) = P(O_{t+1}O_{t+2}\dots O_T | q_t = S_i, \lambda) \quad (2.9)$$

而實際在計算時的計算方式如下：

[1] Initialization :

$$\beta_T(i) = 1, 1 \leq i \leq N \quad (2.10)$$

[2] Induction :

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), t = T-1, T-2, \dots, 1, 1 \leq i \leq N \quad (2.11)$$

[3] Termination :

$$P(O | \lambda) = \sum_{i=1}^N \beta_1(i) \pi_i b_i(O_1) \quad (2.12)$$

由於是由後往前算，因此在步驟 [1] 中，先將所有狀態在時間  $T$  時的機率  $\beta_T(i)$  指定為 1（因為  $\beta_t(i) = P(O_{t+1}O_{t+2}\dots O_T | q_t = S_i, \lambda)$ ，是要考慮時間  $t$  之後的觀察值；所以

對於  $\beta_T(i)$  沒有需要考慮的觀察值，因此  $\beta_T(i) = 1$  )。而在步驟 [2]，則是以時間  $t+1$  的  $\beta_{t+1}(j)$ ，來計算前一刻，時間  $t$  的  $\beta_t(i)$ ；因為對於  $\beta_t(i) = P(O_{t+1}O_{t+2} \cdots O_T | q_t = S_i, \lambda)$  來說，並不知道時間  $t+1$  時的狀態 ( $q_t$ )，所以在計算時，必須要考慮所有的狀態，也就是：

$$\begin{aligned} \beta_t(i) &= P(O_{t+1}O_{t+2} \cdots O_T | q_t = S_i, \lambda) \\ &= \sum_{j=1}^N (P(q_{t+1} = j | q_t = S_i, \lambda) P(O_{t+1} | q_{t+1} = S_j, \lambda) P(O_{t+2}O_{t+3} \cdots O_T | q_{t+1} = S_j, \lambda)) \\ &= \sum_{j=1}^N (a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)) \end{aligned} \quad (2.13)$$

也就如下圖表示：

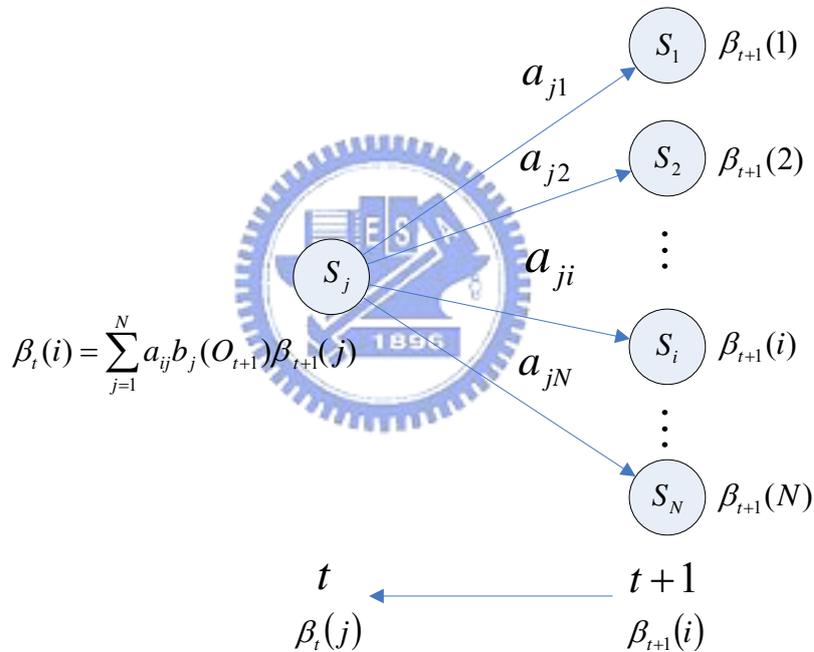


圖 2-3 Backward Algorithm 示意圖

而在步驟 [3] 則是在將所有的  $\beta_t(i)$  計算出來後，對於所有的可能，計算出  $P(O | \lambda)$  的機率：

$$\begin{aligned} P(O | \lambda) &= \sum_{i=1}^N P(O, q_1 = S_i | \lambda) \\ &= \sum_{i=1}^N P(q_1 = S_i | \lambda) P(O_1 | q_1 = S_i, \lambda) P(O_2 O_3 \cdots O_T | q_1 = S_i, \lambda) \\ &= \sum_{i=1}^N \pi_i b_i(O_1) \beta_1(i) \end{aligned} \quad (2.14)$$

## 2.2.2. 最佳狀態轉換序列

由於對於 HMM 來說，狀態的轉換是無法觀測、屬於未知的部份的，所以此問題也沒有確實的正確路徑，僅能根據所得到的觀察值，去找最有可能的狀態轉換路徑。

這邊所用的方法叫做 Viterbi Algorithm，是利用 Dynamic Programming 的方式，去從所有可能的狀態路徑中，找出最有可能的一個。也就是在給定  $O = O_1 O_2 \cdots O_T$  和  $\lambda$  的情形下，去找一組最佳的  $Q = q_1 q_2 \cdots q_T$ ，使  $P(Q | O, \lambda)$  最大；但是因為  $P(O, Q | \lambda) = P(Q | O, \lambda)P(O | \lambda)$ ， $P(O | \lambda)$  和  $Q$  的選擇無關，所以要使  $P(Q | O, \lambda)$  的值最大和使  $P(O, Q | \lambda)$  的值最大是等價的。

為此，定義一個評量用的變數

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} P(q_1 q_2 \cdots q_{t-1}, q_t = S_i, O_1 O_2 \cdots O_t | \lambda) \quad (2.15)$$

代表考慮所有的狀態路徑  $q_1 q_2 \cdots q_{t-1}$ ，在給定參數  $\lambda$  的情況下，時間  $t+1$  之前的觀察值序列為  $O_1 O_2 \cdots O_t$ 、時間  $t$  的狀態為  $S_i$  時，其最大的機率值。

而在時間  $t+1$  時，則可以表示為：

$$\delta_{t+1}(j) = \left[ \max_i \delta_t(i) a_{ij} \right] \cdot b_j(O_{t+1}) \quad (2.16)$$

而實際要計算時，則可由下列步驟完成：

[1] Initialization :

$$\begin{aligned} \delta_1(i) &= \pi_i b_i(O_1), \quad 1 \leq i \leq N \\ \psi_1(i) &= 0 \end{aligned} \quad (2.17)$$

[2] Recursion :

$$\begin{aligned} \delta_t(j) &= \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 1 \leq j \leq N, \quad 2 \leq t \leq T \\ \psi_t(j) &= \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 1 \leq j \leq N, \quad 2 \leq t \leq T \end{aligned} \quad (2.18)$$

[3] Termination :

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (2.19)$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

[4] Path(state sequence) backtracking :

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1 \quad (2.20)$$

### 2.2.3. 調整模型參數

HMM 中最重要，也是最困難的問題，就是要如何決定參數  $\lambda$ ，讓 HMM 模型能夠代表所給的觀察值序列  $O$ ，也就是要讓  $P(O | \lambda)$  的值最大；這種調整參數  $\lambda$  讓 HMM 模型更能代表觀察值序列  $O$  的動作，通常也叫做訓練 (training)。實際上這個問題並沒有確實的解答，而在這裡是使用 Baum–Welch Algorithm (部份文章直接將它叫做 EM(expectation-modification) Method)，根據在一開始所給的起始解，去找局部的最佳解 (local maximum)。

Baum–Welch Algorithm 的步驟大致如下：

1. 給定起始參數
2. E-Step(expectation step)  
估計目前的參數和訓練資料的可能性  $P(O | \lambda)$ 。
3. M-Step(modification step)  
調整目前的參數，使訓練資料對模型的可能性更大；也就是使  $P(O | \lambda)$  更大。
4. 重複 E-Step 和 M-Step，直到  $P(O | \lambda)$  不再變大

整體的流程圖如下：



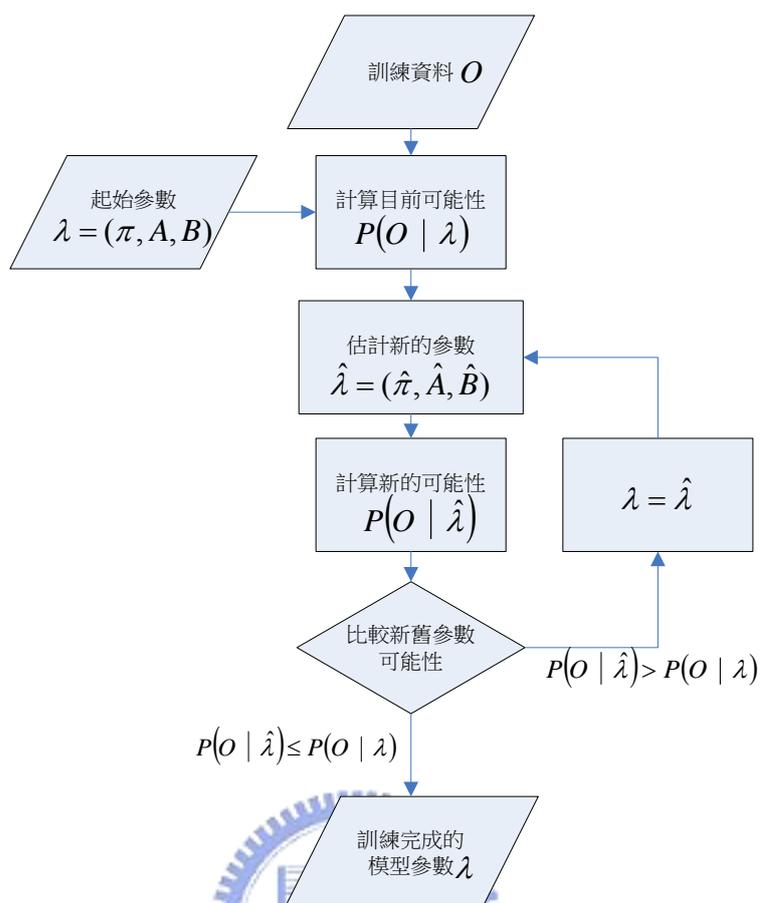


圖 2-4 Baum-Welch Algorithm 流程圖

為了幫助計算，另外再多加定義  $\xi_t(i, j)$  和  $\gamma_t(i)$  這兩個變數。 $\xi_t(i, j)$  代表在給定觀察值序列  $O$ 、參數  $\lambda$  的情形下，時間  $t$  到  $t+1$  的狀態由  $S_i$  轉換到  $S_j$  的機率；也就是：

$$\begin{aligned}
 \xi_t(i, j) &= P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \\
 &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)}, \quad 1 \leq t \leq T-1, 1 \leq i \leq N, 1 \leq j \leq N \\
 &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}
 \end{aligned} \tag{2.21}$$

以圖表示，就是

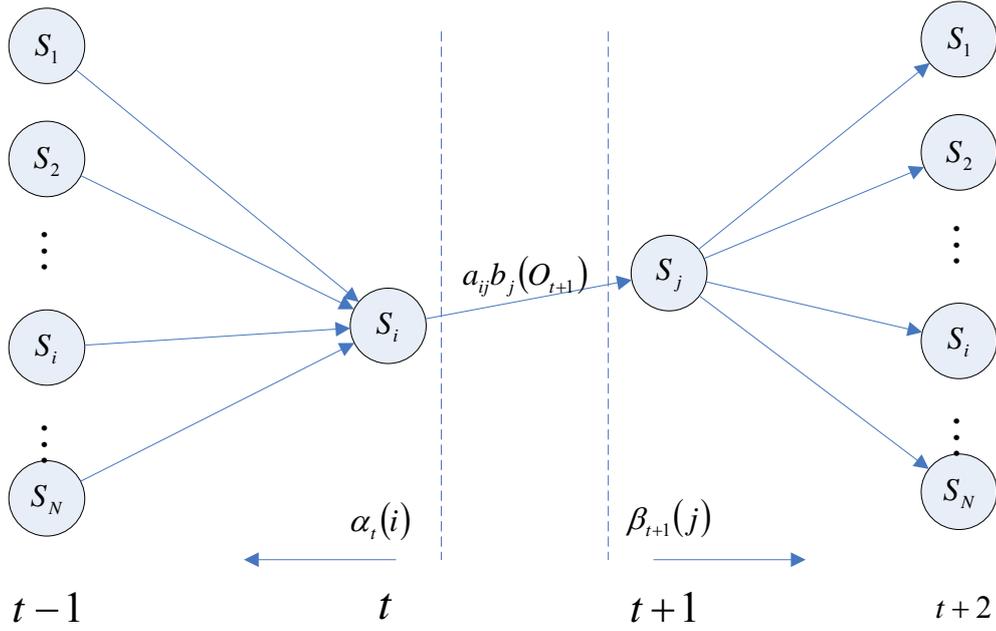


圖 2-5  $\xi_t(i, j)$  示意圖

而  $\gamma_t(i)$  代表在給定觀察值序列  $O$ 、參數  $\lambda$  的情形下，時間  $t$  到  $t+1$  的狀態由  $S_i$  轉換任何狀態的機率；也就是：

$$\gamma_t(i) = P(q_t = i \mid O, \lambda) = \sum_{j=1}^N \xi_t(i, j), \quad (2.22)$$

$$1 \leq t \leq T-1, 1 \leq i \leq N$$

所以，如果考慮時間由 1 到  $T-1$  的話，我們可以得到由狀態  $S_i$  轉換到  $S_j$  的期望值：

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{expected value of transition from } S_i \text{ to } S_j \quad (2.23)$$

以及由  $S_i$  轉換任何狀態的期望值：

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected value of transition from } S_i \quad (2.24)$$

而對於參數  $\lambda$  的調整，則是先利用計算出 Forward variable  $\alpha_t(i)$  和 backward variable  $\beta_t(i)$  後，以此計算出  $\xi_t(i, j)$  和  $\gamma_t(i)$ ；然後再以下列式子，計算出新的參數

$\hat{\lambda} = (\hat{\pi}, \hat{A}, \hat{B})$ ：

$$\hat{\pi}_i = (\text{expected frequency of in state } S_i \text{ at time } t = 1) = \frac{\gamma_1(i)}{\sum_j \gamma_1(j)}, 1 \leq i \leq N \quad (2.25)$$

$$\hat{a}_{ij} = \frac{\text{expected value of transition from } S_i \text{ to } S_j}{\text{expected value of transition from } S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad (2.26)$$

$$1 \leq i \leq N, 1 \leq j \leq N$$

$$\hat{b}_j(v_k) = \frac{\text{expected value in state } S_j \text{ with observation } v_k}{\text{expected value of transition from } S_j} = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}, \quad (2.27)$$

$$1 \leq j \leq N, v_k \in V$$

在以上述式子計算完新的  $\hat{\lambda}$  後，計算  $P(O | \hat{\lambda})$  並和原來的  $P(O | \lambda)$  做比較。如果  $P(O | \hat{\lambda}) > P(O | \lambda)$  則代表新的參數  $\hat{\lambda}$  比舊的參數  $\lambda$  更符合觀察值序列  $O$ ；如此，則可以用  $\hat{\lambda}$  來取代  $\lambda$ 。以此為原則，一直去試著以上述的方法來找到更好的  $\hat{\lambda}$ ，直到  $P(O | \hat{\lambda}) \leq P(O | \lambda)$ ；而這個時候的  $\lambda$  就是找到最符合觀察值序列  $O$  的參數，也就是最後的結果。



## 2.3. HMM 實做的問題

### 2.3.1. 起始參數估計

在 HMM 的架構中，一開始就需要決定的參數有四個：HMM 的狀態數目 ( $N$ )、 $\lambda = (A, B, \pi)$ 。其中， $\pi$  和  $A$  可以使用均勻分布(uniform probability distribution)，對於 Baum–Welch Algorithm 的最佳化不會有影響；但是如果  $B$  也使用均勻分布，會讓 HMM 將所有狀態視為相同的狀態，而無法得到有效的參數。因此，要想辦法決定 HMM 的狀態數目和  $B$  這兩個參數的起始值。

狀態數目大致上可以由訓練資料的觀察值的長度來決定；一般可以用訓練資料長度的平均值來當作觀察值長度，但如果考慮到狀態的重複性，也可以再以固定的比例來縮小[12]。



$B$  的部分則需要利用訓練資料來做預先估測，計算出適當的起始分布。不過由於要做到這件是實際上有相當的難度，所以也有幾篇論文有提到，可以用亂數分布來當作起始值[4][18]。利用亂數當起始分布的缺點就是每次計算出的結果不會相同，也會影響到訓練完後，計算出資料對模型的可能性；不過大致上也都能有一定的效果。本論文目前也是使用亂數當起始值。

### 2.3.2. 訓練資料

HMM 在訓練時，單筆訓練資料觀察值的長度需要比 HMM 的狀態數多，否則有可能會無法收斂到夠好的值。此外，上述的 Baum–Welch Algorithm 也僅能針對一組訓練資料來做最佳化，如果要套用多組訓練資料，則需要再做處理。

一般對於多訓練資料的處理方法有：

1. 直接將多組訓練資料做連接，變成單一組的訓練資料[3]

將多組訓練資料全部串連在一起，成為一組訓練資料，這樣就可以避免掉觀察值長度不夠長的問題；但是由於只剩下一筆訓練資料，所以同時也造成了  $\pi$  無法進行重新估計的問題。

2. 對多組訓練資料各別做訓練，在結束後，再將其以整合[13]

對各筆訓練資料各別去做訓練，在個別訓練結束後，再以類似加權平均的方式整合。由於是將各筆資料各自去做訓練，所以仍然無法解決資料比狀態數目少時會發生的問題。

而本論文所使用的解決方法主要是利用方法 1. 合併的概念，並再做修改以解決無法估計  $\pi$  的問題；實際上修改的結果和 “Data-Driven Design of HMM Topology For Online Handwriting Recognition” [17] 中的 「Combining Models to One Multiple Parallel-path HMM」 的方法在概念上大致上相同。

為了解決無法估計  $\pi$  的問題，除了將各筆訓練資料串在一起外，在每一筆資料的前後都加上一個虛擬的「狀態 Null」來當作起始／結束狀態，並給予此狀態一個觀察值  $v_{Null}$ ；並將狀態 Null 的  $\pi$ 、 $b$  都指定為 1，而在其他狀態則設定為 0。也就是對於狀態 Null， $\pi_{Null} = 1$ ， $b_{Null}(v_{Null}) = 1$ ；而對其他非 Null 的狀態  $i$ ， $\pi_i = 0$ ， $b_i(v_{Null}) = 0$ 。

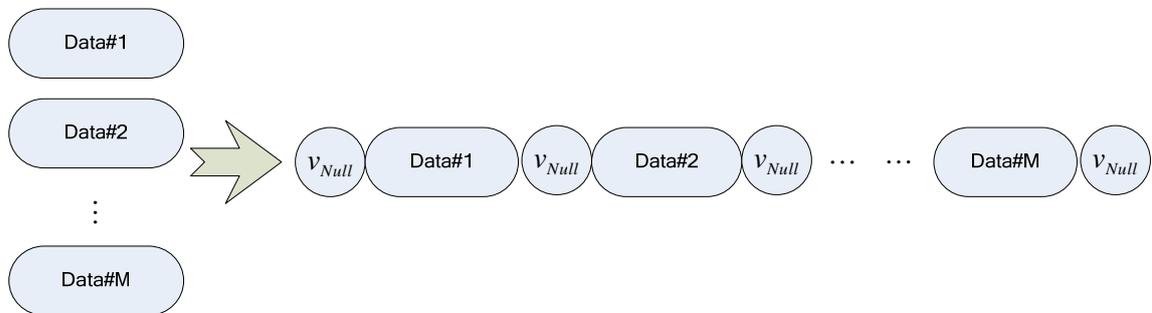


圖 2-6 多訓練資料合併示意圖

在此架構下，由狀態 Null 轉移到其他各狀態的機率  $a_{Null,i}$  就代表原來各狀態的  $\pi_i$ 。

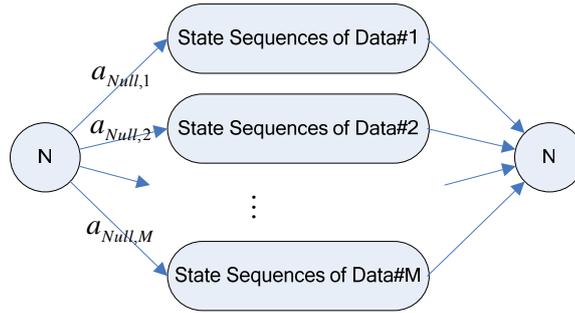


圖 2-7 利用 Null State 的 A 代表  $\pi$

N 代表虛擬的狀態

### 2.3.3. Scaling Problem

如果利用 Forward Algorithm 來計算可能性，會是  $P(O | \lambda) = \sum_{i=1}^N \alpha_T(i)$  的形式。由  $\alpha$  的定義可知，每一項  $\alpha_t(i)$  都是各種可能路徑機率 ( $\pi_{q_1} \prod_{s=1}^{t-1} a_{q_s, q_{s+1}} \prod_{s=1}^t b_{q_s}(O_s)$ , with  $q_t = i$ ) 的總和；其中每一項 ( $a$ 、 $b$ ) 都是一個機率，所以值都介於 0 和 1 之間。如此一來，當  $t$  越來越大， $\alpha_t(i)$  就會越來越小。如果當它的值小到超過電腦浮點數能處理的範圍，將會導致程式無法算出正確的值。尤其在觀察值序列本身就很長時，更容易發生這個問題；這樣將無法進行 HMM 參數的估計，也無法計算出  $P(O | \lambda)$ 。

為了避免發生這個數值計算上的問題，在做計算時必須要做一些不會影響到  $\pi$ 、 $A$ 、 $B$  這些參數的修改；解決方法是在對  $\alpha_t(i)$  做運算時，乘上一個和  $i$  獨立的係數  $c_t$ ，來使的  $\alpha_t(i)$  不會過小。

在實際運算時，由時間 1 到時間  $T$ ，對各時間  $t$  先以原來的計算方式計算出  $\tilde{\alpha}_t(i)$ ，然後再定義  $c_t$  為

$$c_t = \frac{1}{\sum_{i=1}^N \tilde{\alpha}_t(i)} \quad (2.28)$$

然後再對  $\tilde{\alpha}_t(i)$  的值做調整，計算出  $\hat{\alpha}_t(i)$

$$\hat{\alpha}_t(i) = c_t \tilde{\alpha}_t(i) \quad (2.29)$$

如此將所有的  $\hat{\alpha}_t(i)$ 、 $c_t$  計算完。

而  $\beta$  則是先以原來的計算方式計算完  $\tilde{\beta}_t(i)$  後，再以計算  $\hat{\alpha}_t(i)$  時所得到的  $c_t$  來計算

$\hat{\beta}_t(i)$

$$\hat{\beta}_t(i) = c_t \tilde{\beta}_t(i) \quad (2.30)$$

如此計算出的  $\hat{\alpha}_t(i)$  和  $\hat{\beta}_t(i)$ ，就是經過 scale 的 forward variable 和 backward variable。而和原始定義下的  $\alpha_t(i)$ 、 $\beta_t(i)$  相比，其關係為

$$\begin{aligned} \hat{\alpha}_t(i) &= \left( \prod_{s=1}^t c_s \right) \alpha_t(i) = C_t \alpha_t(i) \\ \hat{\beta}_t(i) &= \left( \prod_{s=t}^T c_s \right) \beta_t(i) = D_t \beta_t(i) \end{aligned} \quad (2.31)$$

其中， $\prod_{s=1}^t c_s = C_t$ 、 $\prod_{s=t}^T c_s = D_t$ 。



而對於 HMM 的參數重估計，由於  $\hat{\alpha}_t(i)$  和  $\hat{\beta}_t(i)$  之間的關係， $c_t$  可影響會可以完全消除。以  $A = \{a_{ij}\}$  為例，經過調整所計算出的  $a_{ij}$  為

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N \hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j)} = \frac{\sum_{t=1}^{T-1} C_t \alpha_t(i) a_{ij} b_j(O_{t+1}) D_{t+1} \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N C_t \alpha_t(i) a_{ij} b_j(O_{t+1}) D_{t+1} \beta_{t+1}(j)} \quad (2.32)$$

而  $C_t D_{t+1} = \prod_{s=1}^t c_s \prod_{s=t+1}^T c_s = \prod_{s=1}^T c_s = C_T$ ，所以

$$\begin{aligned}
a_{ij} &= \frac{\sum_{t=1}^{T-1} C_t \alpha_t(i) a_{ij} b_j(O_{t+1}) D_{t+1} \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N C_t \alpha_t(i) a_{ij} b_j(O_{t+1}) D_{t+1} \beta_{t+1}(j)} = \frac{\sum_{t=1}^{T-1} (C_t D_{t+1}) \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N (C_t D_{t+1}) \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \\
&= \frac{\sum_{t=1}^{T-1} C_T \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N C_T \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} = \frac{C_T \sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{C_T \sum_{t=1}^{T-1} \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \\
&= \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}
\end{aligned} \tag{2.33}$$

可以得到和原來定義下相同的  $a_{ij}$ ；而  $B = \{b_j(v_k)\}$ 、 $\pi = \{\pi_i\}$  也是一樣的。

在經過這樣的調整後，無法再以  $\sum_{i=1}^N \hat{\alpha}_T(i)$  的方式來計算機率  $P(O | \lambda)$ ；但由  $c_t$  的定義，可以得到

$$\begin{aligned}
\prod_{s=1}^T c_s \sum_{i=1}^N \alpha_T(i) &= C_T \sum_{i=1}^N \alpha_T(i) \\
&= C_T \sum_{i=1}^N \frac{\hat{\alpha}_T(i)}{C_T} \\
&= \sum_{i=1}^N \hat{\alpha}_T(i) \\
&= \sum_{i=1}^N \frac{\tilde{\alpha}_T(i)}{\sum_{j=1}^N \tilde{\alpha}_T(j)} = 1
\end{aligned} \tag{2.34}$$

而  $\sum_{i=1}^N \alpha_T(i) = P(O | \lambda)$ ，所以可以得到

$$\left( \prod_{s=1}^T c_s \right) P(O | \lambda) = 1 \tag{2.35}$$

因此，可以藉由  $c_t$  來計算出  $P(O | \lambda)$

$$P(O | \lambda) = \frac{1}{\prod_{s=1}^T c_s} \tag{2.36}$$

而由於在  $T$  過大的情況下， $P(O | \lambda)$  也有可能因為過小，超過電腦所可以記錄的

範圍；因此，對於機率的使用，不再直接以  $P(O|\lambda)$  做比較，而改用  $\log(P(O|\lambda))$ ，也就是：

$$\log(P(O|\lambda)) = \sum_{t=1}^T -\log c_s \quad (2.37)$$

由於是對於機率值取  $\log$ ，因此他的值會介於  $0$  到  $-\infty$  之間。



## 三. 多觀察值型隱馬可夫模型的介紹

標準型的 Hidden Markov Model 中，在每一個狀態下一次只能處理一個觀察值，如此構成單一觀察值序列。但是在不少情況下，在同一時間有  $M$  個觀察值；我們藉這多重觀察值來描述系統的運作。對此，我們要擴充原有的標準型 HMM；我們考慮可能有下列幾個擴充方法：

### 1. 合併為單一序列

將  $M$  個序列中同一時間的觀察值組合成一個新的值，並利用組合出來的值的序列來當作新的單一觀察值序列。

這個方法的缺點是所組成的可能觀察值會是原來不同序列的可能觀察值的所有組合，總共的數目會非常大。假使原來的  $M$  個觀察值序列是由可能觀察值數目分別為  $K_1$ 、 $K_2$ 、 $\dots$ 、 $K_M$  的觀察值集合所組成，那所組合成的觀察值的可能總數目將會有  $\prod_{s=1}^M K_s$  個；這樣不但需要相當大的記憶空間，也需要相當長的運算時間。

### 2. 多個 HMM 組合

對於  $M$  個觀察值序列各別建立 HMM 模型，並個別訓練、識別；最後要計算其整體機率時，在將各序列的機率乘起來。而對於  $M$  個序列，也可以對  $M$  個模型加入權重，來修改其對整體的重要性。

這種想法很簡單，對於原始的 HMM 方法也不需要做出太大的修改；但是這樣的缺點是沒辦法考慮到同一時間下， $M$  個序列中觀察值之間的關係。

### 3. HMM with Multiple Feature Streams

參考論文 “Speech Recognition Using Context Vectors And Multiple Feature Streams” [15]。對於 HMM 中的每一個狀態，都同時對  $M$  個序列中觀察值紀錄個別的機率分布；整體的觀察值機率，就由  $M$  個觀察值的機率乘積來組成；這樣可以考慮在同一狀態中的  $M$  個觀察值，而所需的記憶空間 和運算時間都不會

增加太多。而他的特色就是各觀察值的關係是機率獨立的（無法考慮之間的條件機率）。

而本論文採用的方法是上述的方法 3.，以其 Hidden Markov Model with Multiple Feature Streams（HMM/MFS）為主要的模型。



### 3.1. HMM/MFS 的修改

由於基本的Hidden Markov Model無法夠同時接受多組觀察值，所以要將多特徵序列的觀念套用在HMM上，對於變數的定義、演算法都需要做修改。於論文"Speech Recognition Using Context Vectors And Multiple Feature Streams"[15]中，對HMM做了以下的修改：

1.  $M$  為同一個時刻所觀察到的觀察值數目。
2. 可能的觀察值集合  $V$ ，每組觀察值序列都有自己的可能觀察值集合，總共有  $M$  項， $V = \{V^{(1)}, V^{(2)}, \dots, V^{(M)}\}$ 。其中  $V^{(s)}$  為第  $s$  個觀察值序列的所有可能觀察值集合， $V^{(s)} = \{v^{(s)}_1, v^{(s)}_2, \dots, v^{(s)}_{K_s}\}$ ， $1 \leq s \leq M$ ； $K_s$  為第  $s$  個觀察值序列中各時刻的所有可能觀察值的個數。

$v^{(s)}_j$  是第  $s$  個觀察值序列的觀察值， $1 \leq j \leq K_s$ 。

3. 觀察值序列  $O = O_1 O_2 \dots O_T$ ，代表時間長度為  $T$  的觀察值序列。

$$O_t = \begin{bmatrix} O^{(1)}_t \\ O^{(2)}_t \\ \vdots \\ O^{(M)}_t \end{bmatrix}, \text{ 代表在時間 } t \text{ 的 } M \text{ 個觀察值；其中 } O^{(s)}_t \in V_s \text{ 代表於時間 } t \text{ 時，}$$

第  $s$  個序列中的觀察值， $1 \leq s \leq M$ 、 $1 \leq t \leq T$ 。

如此，原來 HMM 中的觀察值機率  $b_j(O_t)$  則變為

$$b_j(O_t) = b_j \left( \begin{bmatrix} O^{(1)}_t \\ O^{(2)}_t \\ \vdots \\ O^{(M)}_t \end{bmatrix} \right) \quad (3.1)$$

其中  $b_j \begin{bmatrix} O_t^{(1)} \\ O_t^{(2)} \\ \vdots \\ O_t^{(M)} \end{bmatrix}$  代表在狀態  $j$  時，觀察值為  $\begin{bmatrix} O_t^{(1)} \\ O_t^{(2)} \\ \vdots \\ O_t^{(M)} \end{bmatrix}$  的機率， $1 \leq j \leq N$ 。而整個狀態的

觀察值機率可由  $M$  個觀察值的機率乘積來表示

$$\begin{aligned}
 b_j \begin{bmatrix} O_t^{(1)} \\ O_t^{(2)} \\ \vdots \\ O_t^{(M)} \end{bmatrix} &= b^{(1)}_j(O_t^{(1)}) \times b^{(2)}_j(O_t^{(2)}) \times \dots \times b^{(M)}_j(O_t^{(M)}) \\
 &= \prod_{s=1}^M b^{(s)}_j(O_t^{(s)}) \\
 &, 1 \leq j \leq N
 \end{aligned} \tag{3.2}$$

其中， $b^{(s)}_j(O_t^{(s)})$  代表在狀態  $j$  時，在第  $s$  個序列所觀察到的觀察值為  $O_t^{(s)}$  的機率；而 HMM 中的觀察值分佈機率矩陣  $B$  也將由  $\{b_j(v_k)\}$  變為  $\{b^{(s)}_j(v^{(s)}_k)\}$ 。

而在 Baum–Welch Algorithm 中對  $b_j(O_t)$  的重估計，也需要修正、對每一個  $b^{(s)}_j(O_t^{(s)})$  來進行；修改後的式子為：

$$\begin{aligned}
 \hat{b}^{(s)}_j(v^{(s)}_k) &= \frac{\text{expected value in state } S_j \text{ with observation } v^{(s)}_k \text{ in stream } s}{\text{expected value of transition from } S_j} \\
 &= \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}_{O_t^{(s)}=v^{(s)}_k} \\
 &, 1 \leq s \leq M, 1 \leq j \leq N, 1 \leq k \leq K_s
 \end{aligned} \tag{3.3}$$

如此，就可以讓 HMM 可以適用於  $M$  個不同的觀察值序列。

## 3.2. 觀察值的權重

由於HMM/MFS能夠處理多個觀察值，而每個觀察值在大部分的情況下，可能會有不同的重要性；如果能夠考慮到此點，也可以提高HMM/MFS在訓練時對訓練資料的可能性。如果要考慮到不同的觀察值有不同的重要性，就需要再計算 $b_j(O_t)$ 時，對觀察值加入權重；在“Speech Recognition Using Context Vectors And Multiple Feature Streams” [15]中提出了對於 $M$ 個觀察值序列各別賦予不同權重的方法。而只要稍作修改，就可以使的每個狀態中的 $M$ 個觀察值，都能有各自獨立的權重。

### 3.2.1. 所有狀態單一權重

此方法中， $M$ 個觀察值序列都會有各自的權重。

所需的修改如下：

先對 $b_j \begin{bmatrix} O_t^{(1)} \\ O_t^{(2)} \\ \vdots \\ O_t^{(M)} \end{bmatrix} = \prod_{s=1}^M b_j^{(s)}(O_t^{(s)})$ 做類似 normalize 的調整

$$\begin{aligned} b_j \begin{bmatrix} O_t^{(1)} \\ O_t^{(2)} \\ \vdots \\ O_t^{(M)} \end{bmatrix} &= \sqrt[M]{\prod_{s=1}^M b_j^{(s)}(O_t^{(s)})} \\ &= \prod_{s=1}^M [b_j^{(s)}(O_t^{(s)})]^{1/M} \end{aligned} \quad (3.4)$$

而修改上式 $\frac{1}{M}$ 的部份，就可以用來調整各觀察值重要性的權重；所以將上式修改為：

$$b_j \left( \begin{bmatrix} O_t^{(1)} \\ O_t^{(2)} \\ \vdots \\ O_t^{(M)} \end{bmatrix} \right) = \prod_{s=1}^M [b_j^{(s)}(O_t^{(s)})]^{w_s} \quad (3.5)$$

如此， $w_s$  就是第  $s$  個序列的觀察值的權重；而  $w_s$  必須要符合  $\sum_{s=1}^M w_s = 1$  的限制。

如果考慮以機率的  $\log$  來看，則變為

$$\log \left( b_j \left( \begin{bmatrix} O_t^{(1)} \\ O_t^{(2)} \\ \vdots \\ O_t^{(M)} \end{bmatrix} \right) \right) = \sum_{s=1}^M w_s \log(b_j^{(s)}(O_t^{(s)})) \quad (3.6)$$

在這種架構下，代表一個 HMM/MFS 的參數  $\lambda$  將會多一個元素：觀察值的權重矩陣  $W = \{w_s\}$ 。所以變為  $\lambda = (\pi, A, B, W)$ 。

### 3.2.2. 各狀態獨立權重

上節所提的方法中，權重是對於各觀察值序列去做，也就是同一個觀察值序列的權重在不同的狀態中也是相同的。而如果考慮到有可能會因為狀態的不同，而各序列中的觀察值會有不同的重要性，因此又對上述的方法做了些修改，使每個狀態中的  $M$  個觀察值都有各自的權重。修改如下：

$$b_j \left( \begin{bmatrix} O_t^{(1)} \\ O_t^{(2)} \\ \vdots \\ O_t^{(M)} \end{bmatrix} \right) = \prod_{s=1}^M [b_j^{(s)}(O_t^{(s)})]^{w_{s,j}} \quad (3.7)$$

其中， $w_{s,j}$  就是在狀態  $j$  中，第  $s$  個觀察值的權重； $w_{s,j}$  要符合  $\sum_{s=1}^M w_{s,j} = 1$  的限制。

### 3.2.3. 權重的計算

權重計算的原則，是以提高訓練資料對模型的可能性 ( $P(O|\lambda)$ ) 為前提。對於權重的估計方式，考慮過可能可以使用類似求最佳解的方式，來找到一組最佳的權重分配；但是實際上，由於權重的特性限制 ( $0 \leq w_s \leq 1$ ,  $\sum_s w_s = 1$ )，對於此問題的「最佳解」的求法，least square 或 linear programming 這類的方法並不適合拿來用。因此，我們使用較為簡單的計算：利用觀察值的機率分布變異數來代表權重。

以「各狀態獨立權重」時來說，必須先計算出各狀態下， $M$  個觀察值分布變異數：

$$\text{Var}_i(s) = \text{Variance}(b^{(s)}_i), \quad 1 \leq i \leq N, \quad 1 \leq s \leq M \quad (3.8)$$

其中  $b^{(s)}_i$  代表狀態  $i$  中，序列  $s$  的所有觀察值出現機率分布； $\text{Var}_i(s)$  代表狀態  $i$  中，序列  $s$  的所有觀察值出現機率分布變異數。而對於狀態  $i$ ，第  $s$  個序列的觀察值的權重  $w_{s,i}$  就是

$$w_{s,i} = \frac{\text{Var}_i(s)}{\sum_{k=1}^M \text{Var}_i(k)}, \quad 1 \leq i \leq N, \quad 1 \leq s \leq M \quad (3.9)$$

而對「所有狀態單一權重」，則是先計算  $M$  個特徵在不同狀態下的變異數和

$$\text{Var}(s) = \sum_{i=1}^N \text{Variance}(b^{(s)}_i), \quad 1 \leq s \leq M \quad (3.10)$$

然後再以此計算出序列  $s$  的權重  $w_s$ ：

$$w_s = \frac{\text{Var}(s)}{\sum_{k=1}^M \text{Var}(k)}, \quad 1 \leq s \leq M \quad (3.11)$$

而運算的時機，可以在 Baum–Welch Algorithm 訓練時，每次遞迴時都計算、更新權重；也可以在訓練完成後，只計算、更新一次。這兩種方法對於訓練的結果也會有不同的影響，本論文也將在實驗時，對兩種算法的結果去做比較。

## 四. 雜訊干擾下的資料辨識

### 4.1. 理想模型無法辨識的原因

在進行辨識時，我們是以觀察值序列資料  $O$  在參數為  $\lambda$  的模型下發生的機率  $P(O | \lambda)$ ，來當作判別的依據；而實際上， $P(O | \lambda)$  就可以視為資料  $O$  和對參數為  $\lambda$  的模型的可能性 (likeliness)。而如果對於所有的模型的可能性皆為 0 的話，就代表這筆資料對所有的模型都不可能，系統只能將此資料視為無法辨識。這種情形的原因大多會是由於訓練資料不足或資料中有雜訊，造成資料超出訓練資料的範圍所造成的。

而會造成一筆資料對模型的可能性是 0 的原因，是因為在以 Forward Algorithm 計算可能性時，必須先依序計算出  $\alpha_t(i)$ ，並根據  $\alpha_t(i)$  計算出  $\alpha_{t+1}(i)$ ；但是如果在某個時間  $t$ ， $\sum_{i=1}^N \alpha_t(i) = 0$  的話，則在時間  $t$  之後的所有  $\alpha_{t'}(i)$  的值都會是 0，也就是


$$\alpha_{t'}(i) = 0, t' \geq t$$

如此的結果，就會使整筆資料的可能性為 0

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i) = \sum_{i=1}^N 0 = 0。$$

而實際上造成  $\sum_{i=1}^N \alpha_t(i) = 0$  是代表在時間  $t$  的時候，無法為給定的觀察值序列  $O$  找到任何可能的狀態轉移路徑；造成的原因以及其代表的意義有下面兩種可能：

1. 無法完全對應狀態
2. 無法做狀態轉換

## 4.2. 無法完全對應狀態的情況

無法完全對應狀態是指在時間  $t$  時，該組觀察值無法找到任何可以完全對應 (exact match) 的狀態。代表在這一時刻時，對所有的狀態  $j$ ， $b_j(O_t) = 0$ ；因此

$$\alpha_t(j) = \left[ \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_t) = \left[ \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] \times 0 = 0$$

,  $1 \leq j \leq N$

對於這種情形，在 HMM/MFS 中會更為常見。如果一筆資料  $O$  中，在時間  $t$  的第  $s$  個觀察值  $O^{(s)}_t$  是完全沒有在訓練資料中出現過的，那該觀察值的發生機率就會是 0 ( $b^{(s)}_j(O^{(s)}_t) = 0, 1 \leq j \leq N$ )；而既使其他序列的觀察值都有相當高的機率，該觀察值

所在的時間點的整組觀察值機率仍然會變為 0 ( $b_j \left( \begin{matrix} O^{(1)}_t \\ O^{(2)}_t \\ \vdots \\ O^{(M)}_t \end{matrix} \right) = 0$ )。而這也將導致在該時刻的觀察值無法找到任何對應的狀態，而使得整體的可能性會變為 0 ( $P(O | \lambda) = 0$ )。

實際上，我們可以利用 HMM/MFS 多觀察值序列的特性，在出現這個問題時去修改機率計算的方式，去忽略某些無法對應觀察值，只去計算可以對應的觀察值，讓它可以找一個比較近似的狀態來對應。

舉例來說，如果觀察值  $O_t = \begin{bmatrix} O^{(1)}_t \\ O^{(2)}_t \\ O^{(3)}_t \end{bmatrix}$  在所有的狀態中，都因為  $b^{(3)}_j(O^{(3)}_t) = 0$  而導致

$$b_j(O_t) = b_j \left( \begin{bmatrix} O^{(1)}_t \\ O^{(2)}_t \\ O^{(3)}_t \end{bmatrix} \right) = (b^{(1)}_j(O^{(1)}_t))^{w_{1,j}} \cdot (b^{(2)}_j(O^{(2)}_t))^{w_{2,j}} \cdot 0^{w_{3,j}} = 0$$

；那如果忽略  $b^{(3)}_j(O^{(3)}_t)$  的機

率，只考慮  $b^{(1)}_j(O^{(1)}_t)$  和  $b^{(2)}_j(O^{(2)}_t)$ ，就可以計算不考慮  $b^{(3)}_j(O^{(3)}_t) = 0$  情形下的機率  $b'_j(O_t) = (b^{(1)}_j(O^{(1)}_t))^{w_{1,j}} \cdot (b^{(2)}_j(O^{(2)}_t))^{w_{2,j}}$ 。但是由於這樣只計算兩個觀察值機率的乘積，會使的新的機率值和原來的計算方式相比下明顯的提高（因為少乘上一個小於 1 的數）；因此，要再去對權重做調整，使  $w'_{1,j} + w'_{2,j} = 1$ ， $b'_j(O_t) = (b^{(1)}_j(O^{(1)}_t))^{w'_{1,j}} \cdot (b^{(2)}_j(O^{(2)}_t))^{w'_{2,j}}$ 。而  $b'_j(O_t)$  這種忽略部份觀察值的觀察值機率，我們稱它為「容錯觀察值機率」。

而在計算出容錯觀察值機率  $b'_j(O_t)$  後，再用  $b'_j(O_t)$  來繼續計算  $\alpha_t(i)$  以及  $P(O|\lambda)$ ；如此，就能算出在忽略  $b^{(3)}_j(O^{(3)}_t) = 0$  的情形下，整筆資料和模型的可能性  $P'(O|\lambda)$ 。我們把這種利用「容錯觀察值機率」所計算出的可能性，稱為「容錯可能性」，並以此當做辨識時比較的依據。

實際上在進行運算時，我們提出下面兩種不同的計算方式：

1. 拋棄無法對應的部份觀察值

將所有  $b^{(s)}_j(O^{(s)}_t) = 0$  的觀察值都捨棄不計，只考慮  $b^{(s)}_j(O^{(s)}_t) \neq 0$  的觀察值

2. 拋棄權重小的部份觀察值

依照權重大小，依序拋棄比較不重要的觀察值。

### 4.2.1. 解決方法一：拋棄無法對應的部份觀察值

本方法的基本概念，就是在發現沒有完全對應狀態時，將無法對應的觀察值都忽略不計，只考慮可以對應的觀察值，使它可以找到一個可能的對應狀態。

計算時，就是重新利用 Forward Algorithm 計算可能性  $P(O|\lambda)$ 。先依序計算出  $\alpha_t(i)$ ，但是在  $\sum_{i=1}^N \alpha_t(i) = 0$  的時候，去調整觀察值的權重，並修改觀察值機率的計算方法，使系統在  $b^{(s')}_j(O^{(s')}_t) = 0$  時，去忽略  $s'$  這一個觀察值，只計算其他觀察值。

也就是先把在給定時間  $t$ 、狀態  $j$  的情形下，先去計算所有的  $b^{(s)}_j(O^{(s)}_t)$ ；如果  $b^{(s')}_j(O^{(s')}_t) = 0$ ，則定義  $w_{s',j} = 0$ ，然後再對權重作 **normalize** 計算出新的權重  $w'_{s',j}$ ，使  $\sum_{s=1}^M w'_{s,j} = 1$  仍然成立。

$$w'_{s,j} = \begin{cases} \frac{w_{s,j}}{\sum_{s=1}^M w_{s,j}} & , \text{if } b^{(s)}_j(O^{(s)}_t) \neq 0 \\ 0 & , \text{otherwise} \end{cases} \quad (4.1)$$

然後修改  $b_j \begin{pmatrix} O^{(1)}_t \\ O^{(2)}_t \\ \vdots \\ O^{(M)}_t \end{pmatrix}$  的計算方式為  $\prod_{b^{(s)}_j(O^{(s)}_t) \neq 0} [b^{(s)}_j(O^{(s)}_t)]^{w'_{s,j}}$ ，並以此為時間  $t$  的觀察值的機

率。

因此，容錯觀察值機率就是

$$b'_j(O_t) = \begin{cases} \prod_{b^{(s)}_j(O^{(s)}_t) \neq 0} [b^{(s)}_j(O^{(s)}_t)]^{w'_{s,j}} & , \text{if } \sum_{i=1}^N \alpha_i(i) = 0 \\ b_j(O_t) & , \text{otherwise} \end{cases} \quad (4.2)$$

而以容錯觀察值機率計算出的可能性，就是容錯可能性  $P'(O|\lambda)$ ；由於這種計算方式已經藉由調整權重來消除掉少乘數個機率的問題了，因此可以直接拿來作為比較的依據。

但是如果所有的觀察值都被忽略的話，那  $b'_j(O)$  還是會是 0；所以這種方法計算的容錯可能性還是有可能是 0。

## 4.2.2. 解決方法二：拋棄權重小的部份觀察值

本方法是由前一個方法「拋棄無法對應的部份觀察值」修改而來；在「拋棄無法

對應的部份觀察值」這個方法中，只要  $b^{(s')}_j(O^{(s')}_t) = 0$ ，就會去忽略  $s'$  這一個觀察值，因此對於不同的狀態，所忽略的觀察值可能也會不一樣。由於考慮到權重的大小可以代表觀察值在這個狀態的重要性，所以我們就修改方法「拋棄無法對應的部份觀察值」，先從較不重要的觀察值開始忽略。

其計算方法就是利用 Forward Algorithm 計算可能性，先依序計算出  $\alpha_t(i)$ ，但是在  $\sum_{i=1}^N \alpha_t(i) = 0$  的時候，試著依權重大小（由小到大），去忽略觀察值。這個方法必須計算多次，逐步增加忽略的觀察值數目，直到  $\sum_{i=1}^N \alpha_t(i) \neq 0$  或  $M$  個觀察值都被忽略。

但是在這種計算方式下，如果是權重最大的觀察值造成其機率為 0 的話，那麼使用此方法來計算的容錯可能性仍然會是 0；實際上在測試的時候，發現這種情況非常多。因此，如果發現這種情況，我們就將此時的容錯觀察值機率改為「拋棄無法對應的部份觀察值」的計算方式，並繼續計算容錯可能性  $P'(O|\lambda)$ 。

由於要依照權重來做忽略的順序，因此在計算前，要先將觀察值依權重來做排序，得到  $O^{(x_1)}_t, O^{(x_2)}_t, \dots, O^{(x_M)}_t$ ， $w_{x_1,j} \leq w_{x_2,j} \leq \dots \leq w_{x_M,j}$ 。然後定義「依權重順序，忽略  $n$  個觀察值的容錯觀察值機率」 ${}^n b_j(O_t)$ ，其計算方式為：

$${}^n b_j(O_t) = \begin{cases} \prod_{y=n+1}^M [b^{(x_y)}_j(O^{(x_y)}_t)]^{w'_{x_y,j}} & , \text{if } n < M \\ 0 & , \text{if } n = M \end{cases} \quad (4.3)$$

$$w'_{x_y,j} = \frac{w_{x_y,j}}{\sum_{y=n+1}^M w_{x_y,j}}$$

容錯觀察值機率  $b'_j(O_t)$  定義則為：

$$b'_j(O_t) = \begin{cases} b_j(O_t) & , \text{if } \sum_{i=1}^N \alpha_t(i) \neq 0 \\ {}^n b_j(O_t) & , \text{if } \sum_{i=1}^N \alpha_t(i) = 0 \text{ and } n < M \\ \prod_{\substack{s=1 \\ b^{(s)}_j(O^{(s)}_t) \neq 0}}^M [b^{(s)}_j(O^{(s)}_t)]^{w'_{s,j}} & , \text{otherwise} \end{cases} \quad (4.4)$$

其中  $n$  是使  $\sum_{i=1}^N \alpha_t(i) \neq 0$  的最小整數， $n = 1, 2, \dots, M$ ，如果  $n = M$  則代表所有的  ${}^n b_j(O_t)$  都

無法使  $\sum_{i=1}^N \alpha_t(i) \neq 0$ ；此時就改用  $b'_j(O_t) = \prod_{\substack{s=1 \\ b^{(s)}_j(O^{(s)}_t) \neq 0}}^M [b^{(s)}_j(O^{(s)}_t)]^{w'_{s,j}}$ ，

$$w'_{s,j} = \begin{cases} \frac{w_{s,j}}{\sum_{\substack{s=1 \\ b^{(s)}_j(O^{(s)}_t) \neq 0}}^M w_{s,j}} & , \text{if } b^{(s)}_j(O^{(s)}_t) \neq 0 \\ 0 & , \text{otherwise} \end{cases} \quad \circ$$

而以此容錯觀察值機率  $b'_j(O_t)$  計算出的可能性，就是最後拿來作為比較依據的容錯可能性  $P'(O|\lambda)$ 。



### 4.3. 無法做狀態轉換的情況

無狀態轉換是指在時間 $t$ 時，雖然觀察值可以找到對應的狀態，但是該組觀察值所找到的對應狀態，無法和上一時刻的結果做狀態轉換。如下圖，狀態內的數值代表其 forward variable  $\alpha_t(i)$ ；狀態 $i, j$ 之間有線相連則代表  $a_{ij} > 0$ ，沒有則代表  $a_{ij} = 0$ 。

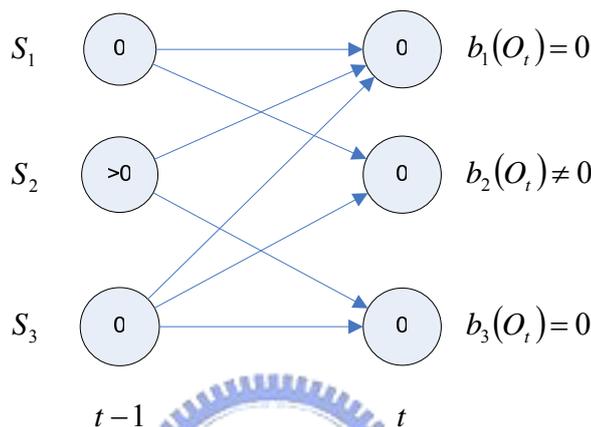


圖 4-1 無狀態轉換示意圖

如上圖的例子，雖然在時間 $t$ ，觀察值 $O_t$ 可以找到對應的狀態 $S_2$ ，但是由於上一時刻的結果中 forward variable  $\alpha_{t-1}(i) > 0$  的狀態只有 $S_2$ ，而 $S_2$ 沒有辦法轉移到 $S_2$

( $a_{22} = 0$ )；所以

$$\begin{aligned} \alpha_t(2) &= [\alpha_{t-1}(1)a_{12} + \alpha_{t-1}(2)a_{22} + \alpha_{t-1}(3)a_{32}]b_2(O_t) \\ &= [0 \times a_{12} + \alpha_{t-1}(2) \times 0 + 0 \times a_{32}]b_2(O_t) \\ &= 0 \end{aligned}$$

而對於狀態 $S_1$ 和 $S_3$ ，則由於 $b_j(O_t) = 0$ ，所以 $\alpha_t(j) = 0$  ( $j = 1, 3$ )。

對於這種可能，由於它已經有找到對應的狀態，所以不容易利用多觀察值的性質在去做處理；因此，我們試著以忽略該時間點的機率的方式，來計算「容錯可能性」 $P'(O|\lambda)$ 。

### 4.3.1. 解決方法：忽略無狀態轉移可能性的時刻

此方法並沒有用到多觀察值的性質，而是在  $\sum_{i=1}^N \alpha_t(i) = 0$  時，直接去忽略該時間的機率；也就是直接將  $b_j(O_t)$  的值設定為 1。所以容錯觀察值機率，就是

$$b'_j(O_t) = \begin{cases} 1 & , \text{if } \sum_{i=1}^N \alpha_t(i) = 0 \\ b_j(O_t) & , \text{otherwise} \end{cases} \quad (4.5)$$

而以容錯觀察值機率計算出的可能性，就是容錯可能性  $P'(O | \lambda)$ 。

不過由於在不同模型中不一定會被忽略掉同樣的數目，因此不能再直接用算出來的容錯可能性來比較；所以必須紀錄被忽略掉的數目，並在計算完容錯可能性後，作為主要比較的依據（忽略的越少越好）；如果被忽略的次數相同，再去比較其容錯可能性。

由於此方法並沒有用到多觀察值的性質，應該也可以應用於一般的 HMM。



# 五. 實驗與結果

## 5.1. 2D 實際資料試驗

### 5.1.1. 實驗設計

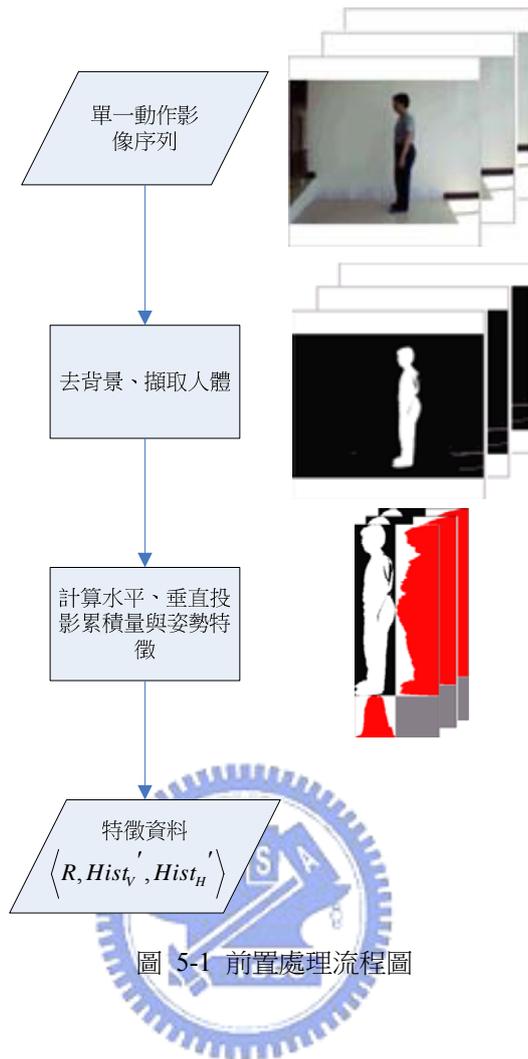
由於現行的大部分方法，要捕捉人體的 3D 動作都較為麻煩；而對於資料的處理，也需要花非常大的工夫；因此，在實際資料測試的部份，本論文選擇由 2D 的特定視角影片來做，直接利用在影像上看到的人體形狀來做分析。

本篇論文利用單一攝影機，針對特定視角（側面）的人體動作進行分析，並試著去辨識八種不同的動作：彎腰、彎腰起身、蹲下、蹲下起身、踢腿、出拳、上勾拳、正步。

主要的處理流程分為前置處理、訓練、辨識等幾部分。

### 前置處理

在對動作進行分析前，先必須先由影像序列中將特徵擷取出來，轉換為觀察值的序列。由於此部分並非本研究的重點，因此大多使用現有的簡單方法來進行，其流程如下：



## 擷取人體輪廓

對於去除背景的方法，也有不少影像處理方面的研究、方法可以作為參考或直接拿來使用，不過由於此部分並非本論文重點，因此在此是先行架構一個背景單純、光影變化不明顯的環境來做試驗。而採用的去除背景的方法，是先建立背景的影像，然後將動作影像序列和背景相減，來把有變化的部份找出來當作前景影像。



圖 5-2 去背景例圖

接著，再由前景影像中，去找最大的連續區塊當做人的形狀。



選取出的區塊

圖 5-3 擷取出的人形

當然，如果擷取出來的人形越完整，在計算特徵時的效果也會越好。而如果在找到的影像區塊中，有過大的破洞、缺塊，將嚴重影響特徵所代表的意義；所以在遇到此問題時，將要另行去做處理，將洞補起。

## 特徵計算

在特徵擷取的部份，主要參考論文“Ghost: A Human Body Part Labeling System Using Silhouettes”[5]，使用人體形狀對水平、垂直兩個方向的投影累積量（The vertical and horizontal normalized projection histograms）來當作主要的特徵資料。

選取出的人形二值影像 (binary image) IMAGE 投影出來的結果，可以用兩個一維陣列  $Hist_V$ 、 $Hist_H$  來表示：

$$Hist_V[x] = \sum_{y=Y_{min}}^{Y_{max}} IMAGE[x, y], \quad X_{min} \leq x \leq X_{max} \quad (五.1)$$

$$Hist_H[y] = \sum_{x=X_{min}}^{X_{max}} IMAGE[x, y], \quad Y_{min} \leq y \leq Y_{max}$$

$Hist_V$ 、 $Hist_H$  在某種程度上可以表現出人的姿勢，雖然對於細部的變化沒有辦法很明確的表現出來，但是對於變化比較大的情況，大多能做出區分。如下圖 (a) 所示，水平方向的投影很明顯可以看出有伸出一個部份，而圖 (b) 則是有兩個；而在垂直方向的投影方面，也可以看出圖 (a) 所凸出的方向是向左方，而圖 (b) 則是左右兩方均有

凸出。

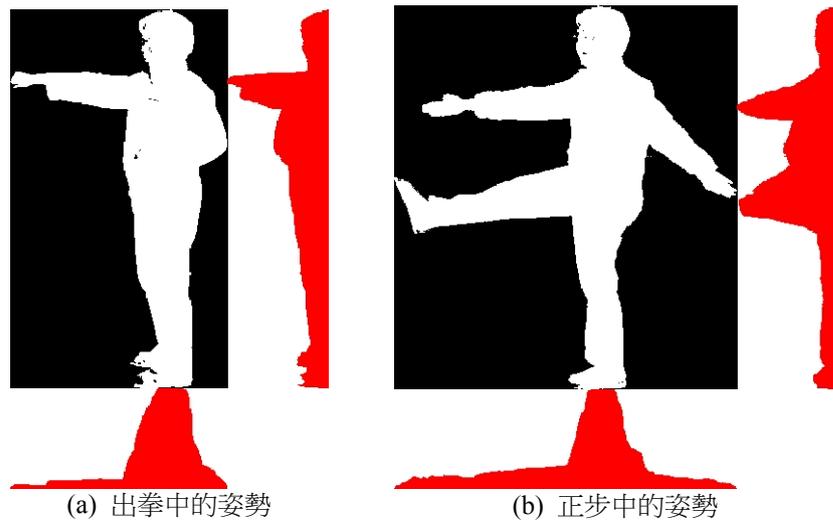


圖 5-4 投影累積量例圖

而實際要對  $Hist_V$ 、 $Hist_H$  做分群的話，要能夠計算之間的差異；因此要再對  $Hist_V$ 、 $Hist_H$  各別去調整。首先，要先使整個序列的  $Hist_V$ 、 $Hist_H$  的大小一致（於本實驗中，統一將其大小改為 128 的陣列），並做 normalize，使的

$$\sum_x Hist'_V[x] = 1, \sum_y Hist'_H[y] = 1 \quad (五.2)$$

如此，經過修改過的資料  $Hist'_V$ 、 $Hist'_H$  就可以利用 Euclidean Distance 的方法，來計算其差異性。

但在調整大小的時候，也會同時失去兩者之間的比例關係，因此也要先計算調整前兩者大小的比值  $R = \frac{IMAGE的高度}{IMAGE的寬度}$ 。

如此， $\langle R, Hist'_V, Hist'_H \rangle$  就是之後來用來做動作辨識用的三組特徵序列。

## 分群與符號化

由於  $\langle R, Hist'_V, Hist'_H \rangle$  這三組特徵序列都是連續的數值資料，因此在訓練、辨識前要先轉換為離散的符號。

在訓練階段時，當得到所有訓練資料的特徵值列後，對三組特徵的空間個別利

用「K-Means 分群法」去分群（要分成幾群要由使用者指定）；而分群後，再對各群各自賦予不同的標籤，並以各群的平均值（相當於中心點）當作代表該群的值。而對於代表一個動作的三組特徵序列，就可以依此轉換為三組觀察值序列；而這三組觀察值的符號序列，就是接下來 HMM/MFS 拿來做訓練的資料。

而在辨識階段，則是直接使用在訓練階段時所得到的各群的平均值拿來當作群中心，對辨識資料做分群；如此，則可以得到和訓練階段一致的符號序列。

例如下面是一組尚未經過符號化的特徵值：

$R$  : 0.237143  
 $Hist'_V$  : 5.77E-05 7.76E-05 0.00019 0.00012 ... 0.000057 0.000020  
 $Hist'_H$  : 0.00052 0.0015 0.00196 0.00235 ... 0.003057 0.002157

而經過分群後，可以發現  $R$ 、 $Hist'_V$ 、 $Hist'_H$  分別屬於各自空間中的第 3 群、第 2 群、

第 4 群，因此，可以將他轉換成  $\begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix}$ ，並以此來表示此時的姿勢。而對於動作影像中

的每一格（frame），都必須先經過這個轉換，使它變成由三個分群編號構成的特徵集合。

如此，每一個動作就會轉換會為這樣的三個分群編號所構成的序列如下：

$$\begin{bmatrix} 1 \\ 0 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 6 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \\ 12 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \\ 12 \end{bmatrix}, \dots$$

而如果根據各特徵來做排列、儲存，則會變成下列的形式：

$R$  : 1, 1, 3, 3, 0, 0, 0, 6, 6, 4, 4, 4, 2  
 $Hist'_V$  : 0, 0, 2, 2, 2, 2, 2, 14, 14, 8, 8, 8, 8  
 $Hist'_H$  : 6, 6, 12, 12, 14, 14, 14, 0, 0, 13, 13, 13, 13

但是如果以每秒 15 張（15FPS）來說的話，一段 30 秒的動作總共就會有 450 張影像；轉換為上述的序列後，序列長度也就會是 450 項。而為了降低減少計算上的複雜度，所以將連續的重複部份刪除掉，只留下有變化的部份，變成下面的形式：

$R$	:	1,	3,	0,	6,	4,	2
$Hist_V$	:	0,	2,	2,	14,	8,	8
$Hist_H$	:	6,	12,	14,	0,	13,	13,

最後，就以這種形式的資料來代表每一個連續的動作。

## 訓練、辨識

在本論文的實驗中，為了測試在各種情況下，原始 HMM/MFS 的辨識方法效果以及本論文「雜訊干擾下的資料辨識」方法的效果，所以除了一般較大量的訓練資料試驗外，也刻意降低訓練資料量來增加無法辨識的資料量。

而對於辨識結果的影響，主要分成「可控制的參數、方法」與「無法控制的因素」兩部份。

### 可控制的參數、方法

包含訓練資料的數量，主要可以拿來調整、影響辨識結果的部分有下列幾點：

#### 1. 訓練資料、測試資料的樣本數目

分別對每一種動作各取 5、10、20 筆資料來當作訓練資料，並取 10 筆資料當作測試用。

#### 2. 分群時的分群數目設定

也就是 K-Means 演算法中 K 的值。拿來測試的三組觀察值分群數目組合為 3, 7, 7、5, 10, 10、7, 15, 15、10, 20, 20 共四種。

#### 3. 權重的計算方式

包含不特別處理權重，直接使用均勻分布，總共有五種不同的方法：

##### A. 不加權重(Uniform)

各組特徵的權重相同

##### B. 各狀態獨立權重，每次計算(IW)

##### C. 各狀態獨立權重，計算一次(IWO)

- D. 各狀態單一權重，每次計算(SW)
- E. 各狀態單一權重，計算一次(SWO)

#### 4. 辨識方法

除了「雜訊干擾下的資料辨識」中所提的三種方法外，包括只使用原始的辨識方法，而不做額外處理，總共有四種不同的方法：

- 1. 不做額外處理(None)
- 2. 拋棄無法對應的部份觀察值(FeatureSkip)
- 3. 拋棄權重小的部份觀察值(StepFeatureSkip)
- 4. 忽略無狀態轉移可能性的時刻(TimeStataSkip)

在此，我們將對上面所提到所有可能的組合都進行測試。也就是對於資料，總共有 12 種組合（訓練資料數目、分群數目）；而對每一種資料，都再做 5 種不同的訓練（權重的計算）。最後，再對上述共 60 組模型，進行 4 種不同的辨識方式。



## 無法控制的因素

除了上述的參數會影響辨識結果外，另外還有三個無法控制的因素也會影響結果：

### 1. 訓練樣本

訓練的樣本的選取對於辨識的結果影響非常大，選到好的樣本，會大幅的提升辨識的正確率，反之則會大幅降低。所以在選取訓練資料和測試資料時，都是採用隨機選取的方法；而為了確保實驗的有效性，每一次測試都選出五組不同的訓練資料、辨識資料，並對他們的辨識結果取平均。

### 2. K-Means 的結果

目前在得到訓練資料後，必須先將所有訓練資料的各組特徵做分群，並以分群的結果將訓練資料和測試資料轉換為符號序列；但是由於目前所使用的 K-Means 分群的方法，必須用亂數取起始的分群中心，所以每次分出來的結果也不盡相同。因此，在實驗時也對每組不同的訓練資料、辨識資料分別進行數

次的分群，每一次的結果各別去做辨識。

### 3. 亂數的起始模型

在 HMM 中訓練用的 EM Method 必須要先給予一組起始的參數，讓他來作收斂；因此，不同的起始參數結果也會有不同的辨識率。而由於本論文對於 HMM 的起始參數有部份是採用亂數，因此，對同樣的資料也會產生不完全相同的結果。為了避免實驗時的結果碰到特例，也對同一組資料產生數個不同的起始模型進行訓練、辨識，並取以結果的平均值。

在此，由於考量到重複實驗次數過多時會耗費非常多的時間，所以對於上述所提及的因素，都各做 5 次取其結果來平均。



## 5.1.2. 實驗資料

目前拿來作為實驗的動作共八種，包括彎腰、彎腰直立、蹲下、蹲下起身、踢腿、出拳、上勾拳、正步共八種，每一種動作都有四十個以上的樣本。樣本中的動作分成數次拍攝，有部分有由不同的人來做；而每一個動作速度、樣子也有所不同，並沒有刻意去做的非常正確。下表為所有動作的數目列表。

動作名稱	樣本數目
彎腰	43
彎腰起身	43
蹲下	40
蹲下起身	40
踢腿	41
出拳	66
上勾拳	42
正步	50
總和	365

表格 1 動作數目列表

下面圖 5-1 到圖 5-8，各針對每一動作，選取一個來做代表；每個動作都由整個序列中選出幾張較有代表性的畫格來表示連續的動作。

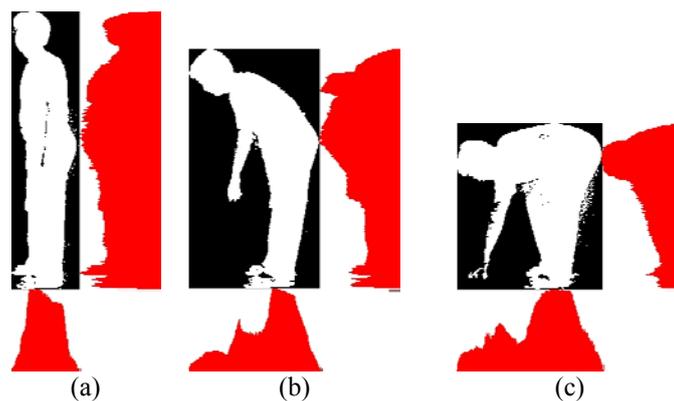


圖 5-5 動作一 彎腰

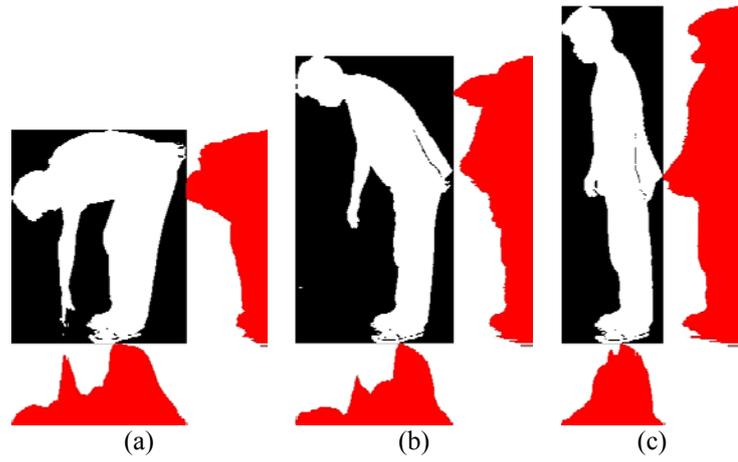


圖 5-6 動作二 彎腰起身

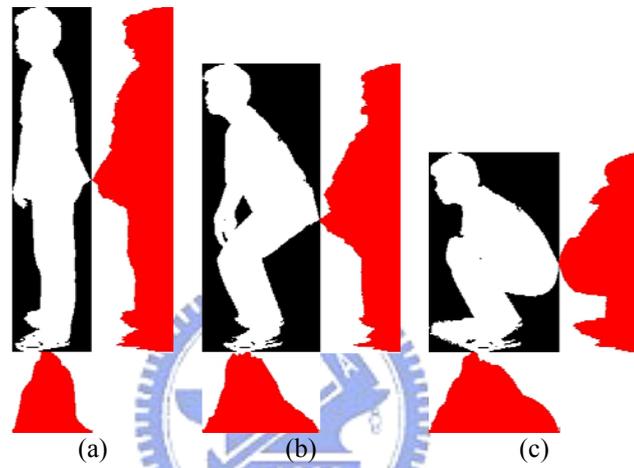


圖 5-7 動作三 蹲下

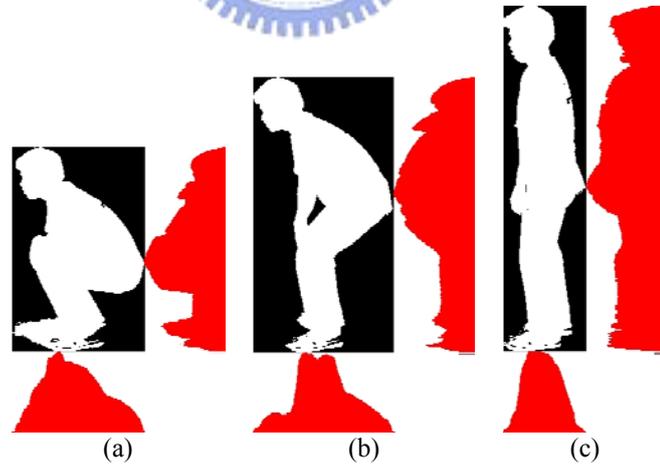


圖 5-8 動作四 蹲下起身

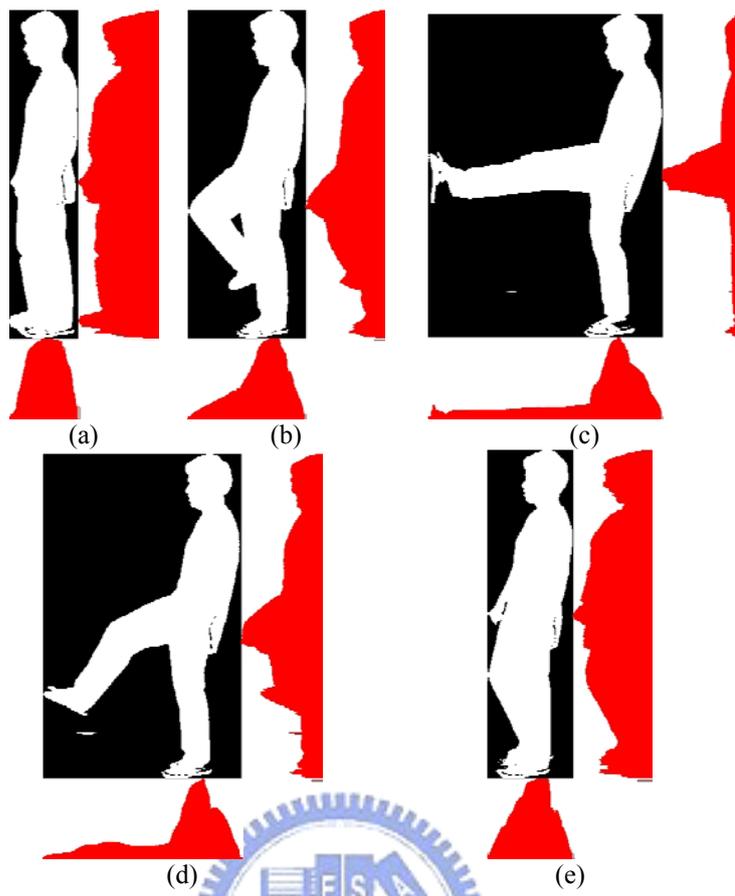


圖 5-9 動作五 踢腿

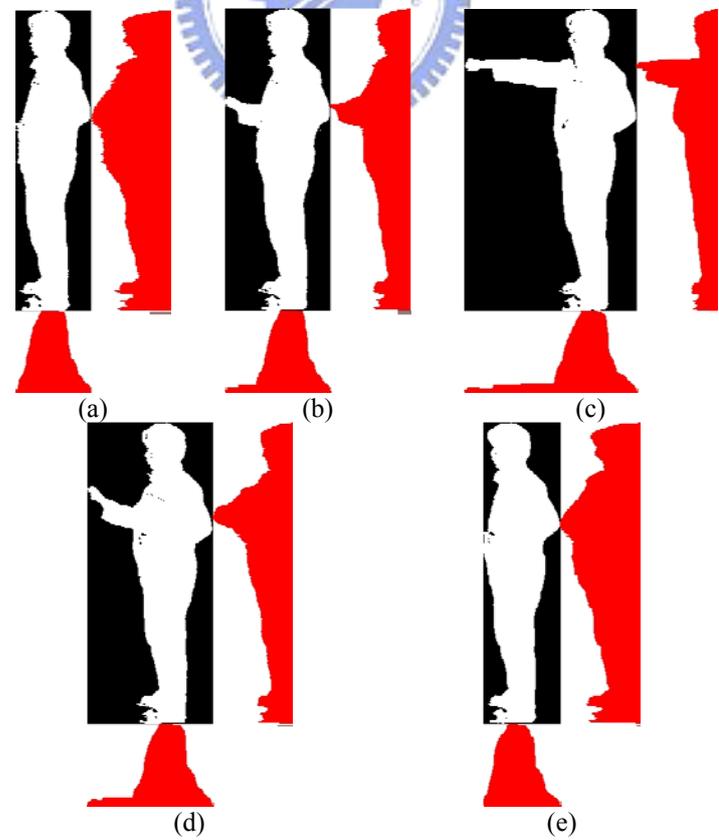


圖 5-10 動作六 出拳

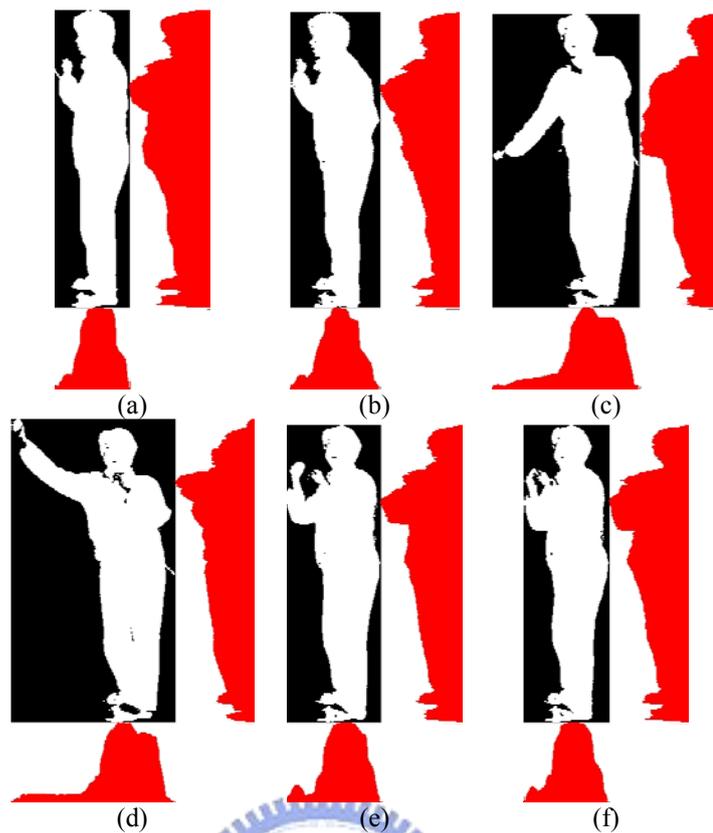


圖 5-11 動作七 上勾拳

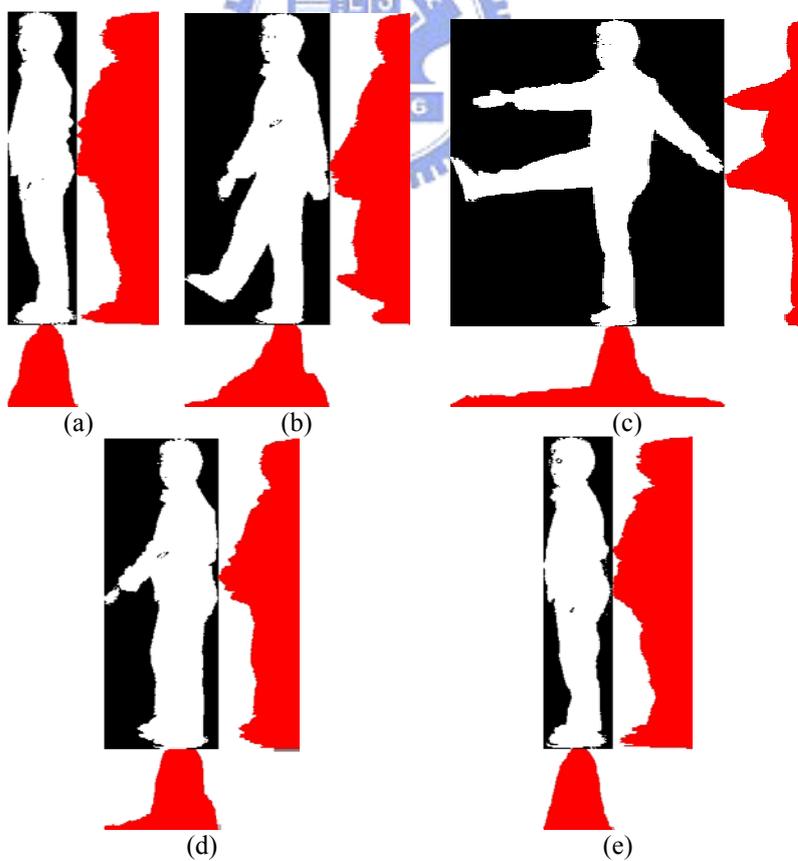


圖 5-12 動作八 正步

而將動作轉為符號序列後，即使同一動作之間也都有相當的差異度。下列各表就是各動作在實驗中轉換為符號的結果（選自訓練資料 10 筆，分群數目為 5, 10, 10 的實驗）：

Sample1	3, 3, 3, 0, 2, 2 3, 5, 5, 5, 5, 5 9, 9, 3, 0, 0, 5
Sample2	4, 4, 4, 3, 0, 2, 2 9, 1, 5, 5, 5, 5, 5 9, 9, 4, 3, 0, 0, 5
Sample3	1, 1, 1, 4, 4, 3, 0, 0, 2, 2 4, 6, 8, 9, 9, 5, 5, 5, 5, 5 9, 9, 9, 9, 3, 3, 0, 5, 5, 1
Sample4	1, 4, 4, 3, 0, 2, 2 4, 8, 9, 5, 5, 5, 5 9, 9, 9, 3, 0, 5, 1

表格 2 動作一 觀察值序列列表

Sample1	2, 2, 0, 3, 3, 3 5, 5, 5, 5, 5, 3 1, 5, 0, 3, 9, 9
Sample2	2, 2, 0, 3, 4, 1, 1, 1 5, 5, 5, 2, 9, 9, 7, 8 5, 0, 3, 3, 9, 9, 9, 9
Sample3	2, 2, 0, 0, 3, 3, 4, 4, 1, 4, 1 5, 5, 5, 3, 9, 2, 8, 4, 4, 4, 4 1, 5, 0, 0, 3, 3, 9, 9, 9, 9, 9
Sample4	2, 0, 3, 4, 1, 1 5, 5, 5, 9, 7, 6 5, 0, 3, 3, 9, 9

表格 3 動作二 觀察值序列列表

Sample1	1, 1, 4, 3, 0 6, 4, 8, 2, 2 9, 9, 9, 7, 7
Sample2	1, 1, 4, 3, 3, 0, 2 9, 3, 9, 9, 2, 2, 2 9, 9, 9, 9, 9, 7, 7
Sample3	1, 1, 4, 4, 3, 0, 0, 0 8, 4, 8, 8, 2, 2, 1, 1 9, 9, 9, 3, 3, 3, 3, 7
Sample4	1, 1, 4, 3, 3, 0 6, 4, 8, 8, 2, 2 9, 9, 9, 7, 7, 7

表格 4 動作三 觀察值序列列表

Sample1	0, 3, 4, 4, 1, 1 2, 1, 1, 8, 8, 6 7, 7, 9, 9, 9, 9
Sample2	2, 0, 0, 3, 4, 4, 1 2, 2, 1, 1, 2, 9, 8 7, 7, 7, 3, 9, 9, 9
Sample3	0, 3, 4, 4, 4, 1, 1 1, 1, 1, 1, 4, 4, 6 7, 7, 7, 9, 9, 9, 9
Sample4	3, 3, 3, 4, 4, 1, 1 1, 1, 8, 8, 9, 7, 6 7, 3, 3, 3, 3, 9, 9

表格 5 動作四 觀察值序列列表

Sample1	1, 1, 4, 3, 0, 2, 0, 3, 4, 1, 1 7, 9, 3, 0, 0, 0, 5, 5, 7, 6 9, 9, 9, 9, 4, 4, 9, 9, 9, 9, 9
Sample2	1, 4, 3, 0, 2, 0, 3, 1 7, 9, 5, 0, 0, 0, 5, 9 9, 9, 4, 4, 4, 9, 9, 9
Sample3	4, 1, 1, 4, 3, 0, 2, 0, 4, 4, 4 4, 4, 7, 7, 5, 5, 0, 5, 1, 8, 1 9, 9, 9, 9, 4, 4, 4, 4, 8, 8, 8
Sample4	1, 1, 4, 3, 2, 2, 2, 0, 3, 4, 1, 4 4, 9, 3, 5, 0, 0, 0, 5, 2, 2, 4, 8 9, 9, 9, 4, 4, 3, 4, 4, 9, 8, 8, 9

表格 6 動作五 觀察值序列列表

Sample1	1, 4, 3, 4, 4, 4, 1 7, 9, 3, 3, 3, 9, 7 8, 8, 6, 6, 8, 8, 8
Sample2	1, 4, 3, 0, 3, 3, 4, 1, 1 7, 9, 3, 0, 3, 3, 3, 7, 9 9, 8, 6, 2, 6, 8, 8, 9, 9
Sample3	1, 1, 1, 4, 3, 3, 4, 1, 1 7, 9, 9, 3, 0, 3, 3, 9, 7 8, 8, 6, 6, 2, 6, 6, 6, 8
Sample4	1, 1, 4, 4, 3, 0, 0, 0, 3, 4, 4, 1, 1, 1, 1 6, 7, 9, 3, 3, 0, 3, 3, 3, 9, 8, 8, 4, 8, 6 9, 9, 9, 8, 6, 2, 2, 6, 8, 8, 8, 8, 8, 8, 8

表格 7 動作六 觀察值序列列表

Sample1	1, 1, 3, 3, 3, 4, 4, 1 8, 7, 3, 3, 3, 3, 9, 7 8, 8, 8, 6, 2, 6, 6, 8
Sample2	1, 4, 4, 0, 0, 3, 3, 3, 4, 4, 1 6, 8, 9, 3, 3, 3, 3, 9, 7, 8, 6 8, 3, 3, 8, 6, 6, 8, 8, 8, 8, 8
Sample3	4, 1, 1, 1, 4, 3, 4, 4, 4, 1, 1 8, 6, 7, 6, 9, 3, 9, 9, 9, 7, 7 8, 8, 8, 3, 8, 6, 6, 8, 6, 6, 8
Sample4	1, 1, 4, 3, 3, 3, 3, 3, 3, 0, 0, 3, 4, 3, 3, 4, 1, 1, 1 6, 4, 8, 9, 2, 3, 5, 2, 1, 2, 2, 2, 2, 5, 5, 2, 8, 4, 6, 7 8, 8, 8, 8, 9, 9, 9, 9, 8, 8, 6, 8, 8, 8, 8, 8, 8, 8, 8, 8

表格 8 動作七 觀察值序列列表

Sample1	1, 1, 4, 4, 3, 0, 2, 2, 2, 2, 0, 0, 3, 4, 1, 1, 1 6, 7, 9, 3, 5, 5, 2, 2, 2, 2, 2, 2, 2, 3, 9, 7, 6 9, 9, 9, 9, 9, 9, 8, 4, 6, 8, 8, 9, 9, 9, 9, 9
Sample2	1, 4, 0, 2, 2, 0, 4, 1 6, 9, 5, 2, 2, 2, 9, 6 9, 9, 9, 4, 6, 8, 9, 9
Sample3	1, 1, 3, 0, 2, 2, 0, 4, 1, 1 6, 7, 3, 5, 5, 2, 2, 9, 9, 6 9, 9, 9, 8, 6, 6, 8, 9, 9, 9
Sample4	1, 1, 3, 0, 2, 2, 0, 0, 4, 1, 1 6, 7, 3, 5, 2, 2, 2, 5, 3, 9, 6 9, 9, 9, 8, 8, 6, 6, 8, 8, 9, 9

表格 9 動作八 觀察值序列列表

而由上述例子，可以發現：實際上對一個動作轉換為符號序列後，他的觀察值數目、符號的轉換，都不是用簡單的比對方法就可以做到的；而使用 HMM 這一類的模型，則可以利用他的機率來吸收這些變化。

### 5.1.3. 實驗結果

#### 整體辨識結果

整體來說，如果我們對「無法辨識」的部分不去做額外的處理，則當訓練資料的數目越多，辨識的正確性就越高；而在不同的訓練資料數目下，也會有不同的最佳分群數目。

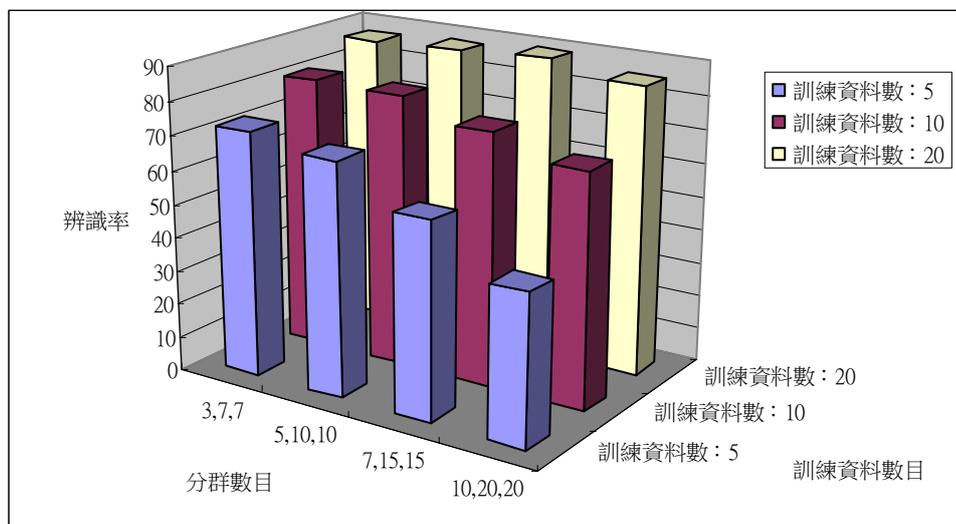


圖 5-13 不同訓練資料數目與分群數目下的辨識正確率

而不管什麼組合的設定，或多或少也都還是會有無法辨識的部份，其機率如下圖：

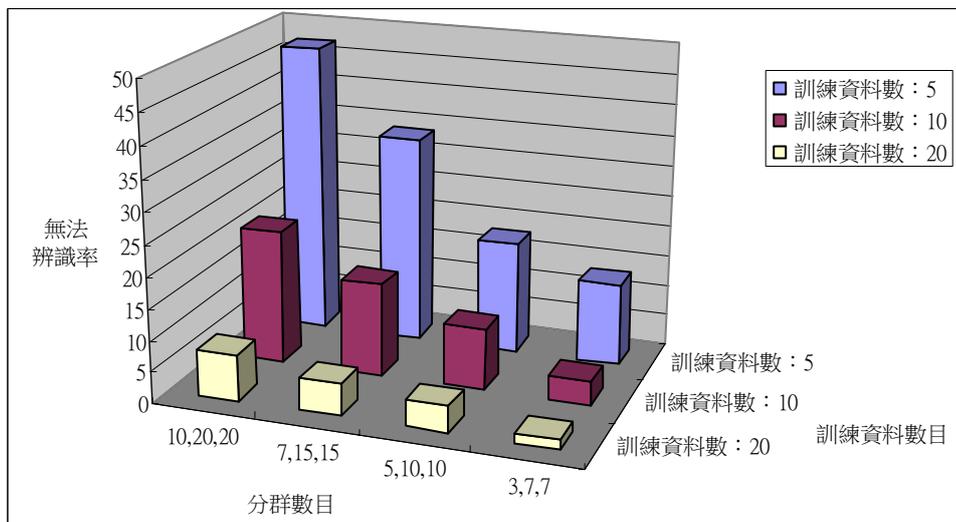


圖 5-14 不同訓練資料數目與分群數目下的無法辨識機率

我們也可以發現，當訓練資料的量越少或者分群數目越多，無法辨識的資料量就會越大；這是由於在這些情況下，觀察值沒有出現在訓練資料的機率會越高所造成的。

而如果使用本論文於「雜訊干擾下的資料辨識」所提出的三種容錯可能性計算方法來辨識本來無法辨識的資料的話，在大部分的情形下，都能以一定程度的正確性，將無法辨識的部份辨識出來，並使整體的辨識率提高到一定的程度；尤其是當訓練資料較少，或分群數目設定不佳時，效果較為明顯。下表即為測試的結果（在同一訓練資料數目下的四組數值，分別是分群數目為 3, 7, 7、5, 10, 10、7, 15, 15、10, 20, 20 的結果）。

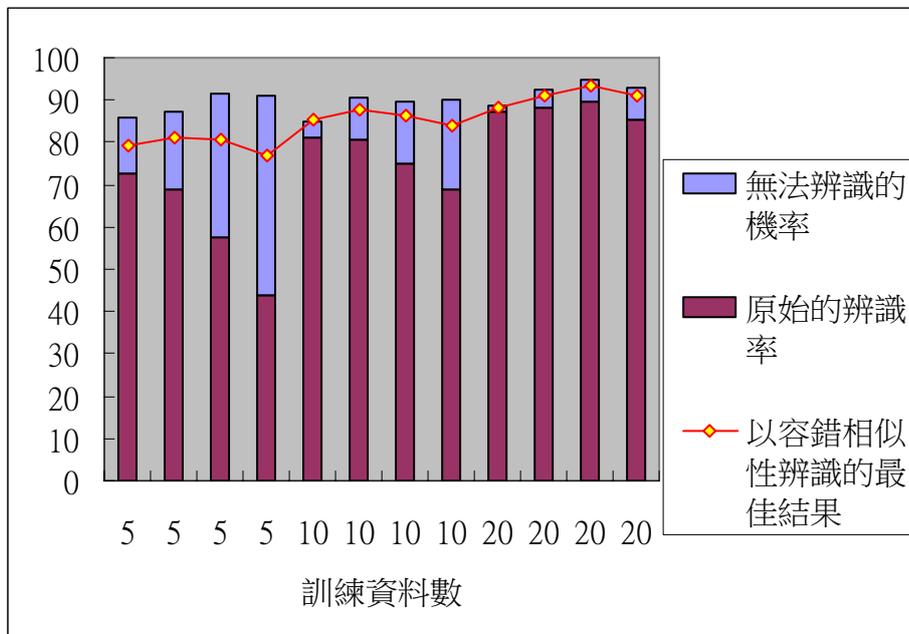


圖 5-15 以容錯可能性辨識的最佳結果

上述數值詳細如下：

訓練資料數目	分群數目	不另作處理			以容錯可能性辨識資料的最佳結果		
		正確	無法辨識	錯誤	正確	無法辨識	錯誤
5	3,7,7	72.5	13.35	14.15	79.41	0	20.59
	5,10,10	69.04	18.23	12.73	81.02	0	18.98
	7,15,15	57.73	33.58	8.69	80.88	0.07	19.05
	10,20,20	44.08	47.14	8.78	76.89	0	23.11
10	3,7,7	81.1	3.77	15.13	85.18	0	14.82
	5,10,10	80.62	9.81	9.57	87.51	0	12.49
	7,15,15	74.78	15	10.22	86.19	0	13.81
	10,20,20	68.71	21.6	9.69	84.16	0	15.84
20	3,7,7	87.17	1.56	11.27	88.12	0	11.88
	5,10,10	88.075	4.195	7.73	91.065	0	8.935
	7,15,15	89.425	5.255	5.32	93.395	0	6.605
	10,20,20	85.445	7.37	7.185	90.93	0	9.07

表格 10 辨識結果整體簡表

## 辨識結果細節

各種不同參數設定、辨識方法的辨識詳細結果如下列各表。列為不同的權重計算方式，其代碼所代表的意義如下：

Uniform	不加權重
IW	各狀態獨立權重，每次計算
IWO	各狀態獨立權重，計算一次
SW	各狀態單一權重，每次計算
SWO	各狀態單一權重，計算一次

欄的部份，則是五種不同的辨識計算方式，其代碼代表意義如下：

None	不做額外處理
FeatureSkip	拋棄無法對應的部份觀察值 (4.2.1)
StepFeatureSkip	拋棄權重小的部份觀察值 (4.2.2)
TimeStataSkip	忽略無狀態轉移可能性的時刻 (4.3.1)

Right 代表辨識正確、none 代表無法辨識、Wrong 代表辨識錯誤，反白的部份代表最佳辨識率。而由於使用均勻分布當權種沒有辦法判斷各特徵的重要性，因此無法適用於「拋棄權重小的部份觀察值」(StepFeatureSkip)這種方法。

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	72.5	13.35	14.15	79.07	0	20.93	N/A			77.17	0	22.83
IW	72.26	12.4	15.34	78.63	0	21.37	78.31	0	21.69	77.37	0	22.63
IWO	72.8	12.67	14.53	79.41	0	20.59	79.23	0	20.77	77.46	0	22.54
SW	70.56	12.34	17.1	75.89	0	24.11	76.36	0	23.64	74.82	0	25.18
SWO	72.12	13.2	14.68	78.67	0	21.33	78.92	0	21.08	76.79	0	23.21

表格 11 訓練資料 5 筆，分群數目 3, 7, 7

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	69.04	18.23	12.73	80.45	0	19.55	N/A			77.96	0	22.04
IW	69.27	15.93	14.8	79.12	0	20.88	79.33	0.01	20.66	76.96	0	23.04
IWO	69.35	17.41	13.24	80.61	0	19.39	80.2	0.01	19.79	77.91	0	22.09
SW	68.86	17.55	13.59	79.02	0	20.98	79.21	0	20.79	77.22	0	22.78
SWO	69.36	17.33	13.31	80.57	0	19.43	81.02	0	18.98	78.05	0	21.95

表格 12 訓練資料 5 筆，分群數目 5, 10, 10

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	57.73	33.58	8.69	78.42	0	21.58	N/A			78.25	0	21.75
IW	58.84	31.72	9.44	80.27	0	19.73	80.88	0.07	19.05	78.25	0	21.75
IWO	58.06	32.98	8.96	79.22	0	20.78	80.6	0.32	19.08	78.14	0	21.86
SW	58.59	32.08	9.33	78.67	0	21.33	80.79	0.01	19.2	77.98	0	22.02
SWO	58.22	33.17	8.61	79.42	0	20.58	80.84	0	19.16	78.57	0	21.43

表格 13 訓練資料 5 筆，分群數目 7, 15, 15

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	44.08	47.14	8.78	75.22	0	24.78	N/A			70.61	0	29.39
IW	44.87	45.51	9.62	76.89	0	23.11	76.47	0.23	23.3	70.68	0	29.32
IWO	44.13	46.34	9.53	76.75	0	23.25	75.59	0.18	24.23	70.15	0	29.85
SW	44.15	45.76	10.09	74.43	0	25.57	76.24	0	23.76	70.01	0	29.99
SWO	44.09	46.78	9.13	76.69	0	23.31	76.83	0	23.17	70.65	0	29.35

表格 14 訓練資料 5 筆，分群數目 10, 20, 20

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	81.1	3.77	15.13	83.75	0	16.25	N/A			82.97	0	17.03
IW	82.49	3.54	13.97	85.18	0	14.82	84.45	0	15.55	84.2	0	15.8
IWO	81.45	3.61	14.94	84.33	0	15.67	83.82	0	16.18	83.37	0	16.63
SW	81.62	3.56	14.82	84.17	0	15.83	84.1	0	15.9	83.18	0	16.82
SWO	80.81	3.72	15.47	83.59	0	16.41	83.44	0	16.56	82.67	0	17.33

表格 15 訓練資料 10 筆，分群數目 3, 7, 7

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	80.62	9.81	9.57	87.37	0	12.63	N/A			85.36	0	14.64
IW	80.42	9.69	9.89	87.18	0	12.82	87.45	0	12.55	85.19	0	14.81
IWO	80.81	9.71	9.48	87.39	0	12.61	87.51	0	12.49	85.73	0	14.27
SW	79.58	9.62	10.8	86.43	0	13.57	86.95	0	13.05	84.28	0	15.72
SWO	80.43	9.68	9.89	86.84	0	13.16	87.21	0	12.79	85.13	0	14.87

表格 16 訓練資料 10 筆，分群數目 5, 10, 10

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	74.78	15	10.22	84.23	0	15.77	N/A			84.07	0	15.93
IW	74.93	14.24	10.83	85.25	0	14.75	85.93	0	14.07	83.31	0	16.69
IWO	74.97	15.02	10.01	85.36	0	14.64	86.19	0	13.81	84.16	0	15.84
SW	74.05	14.43	11.52	84.29	0	15.71	83.96	0	16.04	82.81	0	17.19
SWO	74.61	15.17	10.22	84.75	0	15.25	84.36	0	15.64	83.91	0	16.09

表格 17 訓練資料 10 筆，分群數目 7, 15, 15

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	68.71	21.6	9.69	83.5	0	16.5	N/A			81.11	0	18.89
IW	69.01	20.96	10.03	84.16	0	15.84	82.88	0.16	16.96	81.55	0	18.45
IWO	68.36	21.58	10.06	83.79	0	16.21	82.24	0.19	17.57	80.89	0	19.11
SW	67.84	21.09	11.07	82.43	0	17.57	81.61	0	18.39	80.09	0	19.91
SWO	68.28	21.87	9.85	83.72	0	16.28	82.34	0	17.66	81.05	0	18.95

表格 18 訓練資料 10 筆，分群數目 10, 20, 20

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	87.17	1.56	11.27	88.12	0	11.88	N/A			87.87	0	12.13
IW	85.96	1.5	12.54	86.94	0	13.06	86.77	0	13.23	86.58	0	13.42
IWO	86.44	1.42	12.14	87.33	0	12.67	87.31	0	12.69	87.05	0	12.95
SW	83.73	1.51	14.76	84.71	0	15.29	84.78	0	15.22	84.47	0	15.53
SWO	85.92	1.53	12.55	86.88	0	13.12	87.04	0	12.96	86.63	0	13.37

表格 19 訓練資料 20 筆，分群數目 3, 7, 7

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	88.075	4.195	7.73	91.065	0	8.935	N/A			90.22	0	9.78
IW	87.55	4.295	8.155	90.59	0	9.41	90.685	0.005	9.31	89.79	0	10.21
IWO	87.46	4.27	8.27	90.475	0	9.525	90.795	0	9.205	89.585	0	10.415
SW	85.975	4.255	9.77	88.85	0	11.15	89.015	0	10.985	88.245	0	11.755
SWO	86.94	4.25	8.81	89.97	0	10.03	89.97	0	10.03	89.06	0	10.94

表格 20 訓練資料 20 筆，分群數目 5, 10, 10

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	89.425	5.255	5.32	93.235	0	6.765	N/A			92.155	0	7.845
IW	88.46	5.035	6.505	92.385	0	7.615	91.95	0	8.05	91.125	0	8.875
IWO	89.25	5.315	5.435	93.08	0	6.92	93.09	0	6.91	92.025	0	7.975
SW	86.79	5.08	8.13	90.33	0	9.67	90.925	0	9.075	89.445	0	10.555
SWO	89.05	5.285	5.665	92.75	0	7.25	93.395	0	6.605	91.88	0	8.12

表格 21 訓練資料 20 筆，分群數目 7, 15, 15

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	85.445	7.37	7.185	90.89	0	9.11	N/A			89.485	0	10.515
IW	84.87	7.265	7.865	90.265	0	9.735	89.915	0.005	10.08	89	0	11
IWO	85.12	7.41	7.47	90.71	0	9.29	90.21	0.005	9.785	89.205	0	10.795
SW	83.9	7.51	8.59	89.05	0	10.95	89.195	0	10.805	88.08	0	11.92
SWO	85.165	7.495	7.34	90.66	0	9.34	90.93	0	9.07	89.325	0	10.675

表格 22 訓練資料 20 筆，分群數目 10, 20, 20

由上列各結果表格可知，實際上並沒有哪一種權重計算方式或辨識方法在任何情形下都有較好的結果；而實際上，大部分的方法差異也都不是非常的大。

由於對於權重的計算都是以「提高模型與訓練資料的可能性」為前提、而非以「提高模型間的差異性」來計算的（有經過實驗，有加入權重的模型的確可以得到更佳的可能性）；所以在某些情形下，不另行計算權重（均勻分布）反而會有更佳的鑑別性。而以目前的權重計算方式來說，在訓練資料量較低的情況（訓練資料量 5 或 10）下，對於辨識的正確性較有益。

而對於相同權重方法，「每次計算」的辨識正確率大致都會比「計算一次」來的低；這應該是由於「每次計算」會使模型的收斂程度沒有「計算一次」的模型來的高所造成。

在辨識方法方面，三種方法大多都能把辨識率提高到一定的程度；而利用多觀察值性質來找近似對應狀態的「忽略無狀態轉移可能性的時刻」、「拋棄權重小的部份觀察值」兩種方法，在大部分的情形下，也都能得到最佳的辨識率。

## 5.2. 3D 模擬資料試驗

### 5.2.1. 實驗設計、資料

由於 3D 的人體資料取得不易，因此本論文在 3D 的動作實驗方面，採用模擬的資料來進行。

本論文對於 3D 的實驗，資料型態的假設是將一個人的全身動作分為右手、左手、右腳、左腳和身體五個部份，並在取得人體的各部位的 3D 座標後，轉換為各關節的角度；然後再針對各肢節的動作，以角度、轉軸的變化，個別進行基礎動作（primitive motion）的分析、切割。因此，用以訓練、辨識的資料，就是右手、左手、右腳、左腳和身體這五個部份的基礎動作序列。

目前僅先選「新國民健康操」兩個動作，利用人工定義可能的基礎動作序列，來做初步的測試。這兩個動作的定義資料如下：

#### 1. 動作一 踏步擺手

左手：	前擺,後擺,前擺,後擺,前擺,後擺,前擺,後擺
右手：	後擺,前擺,後擺,前擺,後擺,前擺,後擺,前擺
左腳：	前踏,後踏,前踏,後踏,前踏,後踏,前踏,後踏
右腳：	後踏,前踏,後踏,前踏,後踏,前踏,後踏,前踏
身體：	不動,不動,不動,不動,不動,不動,不動,不動

#### 2. 動作二 踏併擺手

左手：	前擺,後擺,前擺,後擺,前擺,後擺,前擺,後擺
右手：	後擺,前擺,後擺,前擺,後擺,前擺,後擺,前擺
左腳：	前踏,後踏,前踏,後踏,外跨,縮回,外跨,縮回
右腳：	後踏,前踏,後踏,前踏,外跨,縮回,外跨,縮回
身體：	不動,不動,不動,不動,不動,不動,不動,不動

而為了模擬實際資料可能發生的錯誤，以人工的方式，在上述兩種資料中，隨機加入可能的基礎動作辨識錯誤、切割錯誤等「雜訊」。於此模擬試驗中，對兩種動作，

都各以人工的方式建立十個加了雜訊的資料，並取五個當訓練資料、五個當作測試資料。對於資料內容，取機筆資料的詳細列表如下：

### 1. 動作一

前擺,後擺,前擺,後擺,前擺,後擺,前擺,後擺 後擺,前擺,後擺,前擺,後擺,前擺,後擺,前擺 前踏,後踏,前踏,後踏,前踏,後踏,前踏,後踏 後踏,前踏,後踏,前踏,後踏,前踏,後踏,前踏 不動,不動,不動,不動,不動,不動 a, 不動,不動
前擺,後擺,前擺,前擺 a, 後擺,後擺 a, 後擺,前擺,後擺,前擺,後擺 後擺,前擺,後擺,後擺,前擺,前擺,前擺,後擺 a, 前擺 a, 後擺,前擺 前踏 a, 後踏,前踏,前踏,後踏,後踏,後踏,前踏,後踏,前踏,後踏 後踏,前踏,後踏,後踏,前踏,前踏,前踏,後踏,前踏,後踏,前踏 不動,不動,不動,不動 a, 不動 a, 不動,不動,不動,不動,不動
前擺,後擺,前擺,後擺 a, 前擺,後擺,後擺,前擺,後擺 後擺,前擺,後擺,前擺,後擺 a, 前擺 a, 前擺,後擺,前擺 前踏 a, 後踏,前踏,後踏,前踏,後踏,後踏,前踏,後踏 後踏,前踏,後踏,前踏,後踏,前踏,前踏,後踏 a, 前踏 不動,不動,不動,不動,不動,不動,不動,不動,不動
前擺,後擺,前擺,後擺,後擺 a, 前擺 a, 後擺,前擺,後擺 後擺,前擺 a, 後擺,前擺,前擺,後擺,前擺 a, 後擺,前擺 前踏,後踏,前踏 a, 後踏,後踏,前踏,後踏 a, 前踏,後踏 後踏,前踏 a, 後踏,前踏,前踏,後踏,前踏 a, 後踏,前踏 不動,不動,不動,不動,不動 a, 不動,不動,不動,不動
前擺,後擺,前擺,後擺,前擺,後擺 a, 前擺,後擺 後擺,前擺,後擺 a, 前擺,後擺 a, 前擺,後擺,前擺 前踏,後踏,前踏,後踏,前踏,後踏,前踏 a, 後踏 後踏,前踏,後踏 a, 前踏,後踏,前踏,後踏,前踏 不動,不動,不動,不動,不動,不動 a, 不動,不動

### 2. 動作二

前擺,後擺,前擺,後擺,前擺,後擺,前擺,後擺 後擺,前擺,後擺,前擺,後擺,前擺,後擺,前擺 前踏,後踏,前踏,後踏,外跨,縮回,外跨,縮回 後踏,前踏,後踏,前踏,外跨,縮回,外跨,縮回 不動,不動,不動 a, 不動,不動,不動,不動,不動
前擺,後擺,前擺 a, 後擺,前擺 a, 後擺,後擺,前擺,後擺 後擺,前擺 a, 後擺,前擺,後擺,前擺,前擺,後擺 a, 前擺 前踏,後踏 a, 前踏,後踏,外跨,縮回,縮回,外跨,縮回 後踏,前踏,後踏 a, 前踏,外跨,縮回 a, 縮回,外跨,縮回 不動,不動,不動,不動,不動 a, 不動,不動,不動,不動
前擺,後擺,前擺 a, 後擺,後擺,前擺,後擺,前擺,後擺 後擺,前擺 a, 後擺,前擺,前擺 a, 後擺,前擺,後擺,前擺 前踏,後踏,前踏,後踏,後踏,外跨,縮回,外跨,縮回 後踏,前踏 a, 後踏,前踏,前踏,外跨,縮回 a, 外跨,縮回 不動,不動,不動 a, 不動,不動,不動,不動,不動

前擺, 後擺, 前擺 a, 後擺, 後擺, 前擺, 前擺, 後擺, 後擺, 前擺, 後擺
後擺, 前擺 a, 後擺, 前擺, 前擺, 後擺, 後擺 a, 後擺, 前擺, 後擺 a, 前擺
前踏, 後踏, 前踏, 後踏 a, 後踏 a, 外跨, 外跨, 外跨, 縮回, 外跨 a, 縮回
後踏, 前踏, 後踏 a, 前踏, 前踏, 外跨, 外跨, 外跨, 縮回 a, 外跨, 縮回
不動, 不動 a, 不動, 不動, 不動, 不動 a, 不動, 不動, 不動, 不動, 不動
前擺, 前擺 a, 後擺, 前擺 a, 後擺, 前擺 a, 後擺, 前擺, 後擺
後擺, 後擺, 前擺 a, 後擺, 前擺, 後擺, 前擺, 後擺 a, 前擺
前踏, 前踏, 後踏 a, 前踏, 後踏, 外跨, 縮回, 外跨, 縮回
後踏, 後踏, 前踏, 後踏 a, 前踏, 外跨, 縮回 a, 外跨, 縮回
不動, 不動, 不動, 不動, 不動, 不動, 不動, 不動 a, 不動, 不動

其中，雜訊的加入方法是假設每一種基礎動作都可能會被誤判為另一種基礎動作（在名稱後加上 a 的），並考慮可能有多的動作或少的動作。

## 5.2.2. 實驗結果

對於上述的資料，我們使用五組不同的起始亂數模型進行測試，並取其平均結果如下表：

	None			FeatureSkip			StepFeatureSkip			TimeStataSkip		
	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong	Right	None	Wrong
Uniform	76.000	24.000	0.000	92.000	0.000	8.000	N/A			100.000	0.000	0.000
IW	88.000	12.000	0.000	98.000	0.000	2.000	90.000	0.000	10.000	100.000	0.000	0.000
IWO	72.000	28.000	0.000	94.000	0.000	6.000	96.000	0.000	4.000	96.000	0.000	4.000
SW	88.000	10.000	2.000	98.000	0.000	2.000	98.000	0.000	2.000	98.000	0.000	2.000
SWO	72.000	28.000	0.000	88.000	0.000	12.000	84.000	0.000	16.000	98.000	0.000	2.000

表格 23 3D 動作模擬測試結果

雖然模擬實驗的資料量並沒有很大，但是以本研究實驗結果的辨識正確率，應該也足以代表這種方法應用在這方面的實用性。不過應該是模擬的資料無法完全和真實資料相似、且基本性質不盡相同，因此對於使用不同方法的結果，和用 2D 的實際試驗有些許的差異。

雖然利用多觀察值性質來找近似對應狀態的「拋棄無法對應的部份觀察值」、「拋棄權重小的部份觀察值」兩種方法大致上也都能將辨識率提高到一定的程度；不過最好的結果，卻是在實際資料實驗中較差的「忽略無狀態轉移可能性的時刻」，這應該是由於資料性質不同所造成的影響。目前拿來做模擬實驗的資料，實際上在雙手以及身

體的部份動作的是一樣的，不一樣的是在雙腳的部份；也就是在五個觀察值中，有三個是一樣的、兩個不一樣。所以對於忽略觀察值這一類的方法，很有可能會把不一樣的部份忽略掉，而導致判斷錯誤。確實情況可能要使用較大量的真實 3D 資料來做測試後，才能釐清問題所在。



## 六. 結論與未來工作

### 6.1. 結論

本研究試著將 Hidden Markov Model with Multiple Feature Streams 的多特徵觀念，應用在人體動作的分析上；並嘗試去解決無法辨識的情況。

以實驗結果來看，在有足夠的訓練資料的情況下，將 HMM/MFS 應用在動作分析上，的確可以使用的單一攝影機、固定視角，對於多個動作能有不錯的辨識率。而對於模擬的 3D 資料，也有相當的辨識成功率。

對於訓練資料不足，或分群數目不恰當的情形產生的無法辨識的資料，在實際資料測試時，「拋棄無法對應的部份觀察值」和「拋棄權重小的部份觀察值」這兩種利用多觀察值性質來找近似對應狀態的方法，都可以將辨識的正確性提升到一定的程度；而以模擬資料測試時，則由於資料的性質，是以「忽略無狀態轉移可能性的時刻」有較好的結果。



### 6.2. 未來工作

針對目前所使用的 HMM/MFS 中，實際上還存在不少還可以進行修改的地方；其中比較重要，或比較可以加入的觀念，大致有下列幾點：

1. 改用連續型的隱馬可夫模型 (Continuous Hidden Markov Model)

目前所使用的 HMM 模型都是以離散型為基礎的，所以要先將連續的數值資料轉換為離散的符號；再這個轉換過程中，其實已經有失去一些資訊，同時也會造成機率為 0 的問題。而使用連續性的數值資料來當觀察值，應該能解決這方面的問題。

2. 起始參數  $B$  的估計

目前的方法是使用亂數當作起始的參數，但是這樣的缺點就是每次計算出的結果不會相同，也會影響到訓練完後的可能性。因此，最好是可以有辦法可以利用訓練資料就先行對起始參數計算出一個夠好的估計，以增加之後訓練時的效果。

### 3. 各觀察值權重的計算方式

目前是以「提高模型與訓練資料的可能性」為前提、而非以「提高模型間的差異性」來計算的；如果改以「提高模型間的差異性」為前提來計算，相信也能提高辨識的正確性。

### 4. 考慮狀態的時間長度

現在的方法中，對於動作內各段小動作的時間其實是完全不去在乎的，有考慮的，只有之間轉換的過程。如果能將時間長度的比例也考慮進去，或許也會有所幫助。

### 5. 考慮對於狀態的額外處理

主要有兩種，一個是無用狀態的刪除、一個是同意義狀態的合併。降低狀態數目可以有效的減低運算的複雜程度。



而對於特徵的資料來源，可以考慮由其他更好的特徵來進行；目前的構想中，大致有下面幾個方向可以繼續做嘗試：

#### 1. 以「基礎動作」(primitive motion) 來進行

也就是本研究 3D 模擬資料的部份。先行利用 2D 或 3D 推斷人體的姿勢，而後由姿勢的連續變化，分析出人體的各肢節的基礎動作，然後再將各肢節的基礎動作當作特徵，用在 HMM/MFS 中來進行訓練、辨識。

#### 2. 以「相對特徵」來進行

目前的特徵都是只考慮當下的時間的資料而已，或許可以嘗試考慮以和上一刻的差異、相對關係來做；比如果手的位置是「提高」、「往前」這一類的相對關係。如此，應該可以使動作辨識的通用性更大；但實際上到底有沒有用，

也還是要經過實驗才知道。

3. 以「多視角影像」進行

原則上還是以 2D 的影像來進行，不過可以利用多視角的資訊。先由各不同視角的影像，判斷出最接近訓練時所使用的視角，並以此視角的資料拿做辨識。如此，應該可以在某種程度上解決目前的方法無法利用在不同視角的問題。

4. 加入其他獨立特徵

可以考慮加入其他可以代表行為的特徵，比如說移動的路徑；利用多觀察值型隱馬可夫模型各觀察值獨立的特性，應該可以同時處理不同、沒有關聯性的資料。而同時考慮移動的方法和本身的動作，應該可以再對動作辨識做一些延伸。



## 參考文獻

- [1] L. Wang, W. Hu, and T. N. Tan, “Recent developments in human motion analysis”, *Pattern Recognition*, 36(3), pp. 585–601, 2003.
- [2] J. K. Aggarwal, Q. Cai, “Human Motion Analysis: A Review”, *Computer Vision and Image Understanding*, 73(3), pp. 428–440, 1999.
- [3] Christopher D. Manning, Hinrich Schütze, Foundations of Statistical Natural Language Processing, Fourth printing, The MIT Press, 2001.
- [4] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, *Proc. of IEEE*, 77(2), pp. 257-285, February 1989.
- [5] I. Haritaoglu, D. Harwood, and L. S. Davis, “Ghost: A human body part labeling system using silhouettes”, *Proc. of International Conference on Pattern Recognition*, pp. 77-82, 1998.
- [6] Tz-Hau Tzeng, “A Model-Based Human Motion Analysis System in Multiple-Views”, 國立清華大學 電機工程學系, 碩士論文, 2003.
- [7] J. Yamato, J. Ohya, and K. Ishii, “Recognizing Human Action in Time-Sequential Images Using Hidden Markov Model”, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 379–385, 1992.
- [8] T. Mori, Y. Segawa, M. Shimosaka, and T. Sato, “Hierarchical Recognition of Daily Human Actions Based on Continuous Hidden Markov Models”, *Proc. of Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 779-784, May 2004.
- [9] I-Cheng Chang, Chung-Lin Huang, “Skeleton-based Walking Motion Analysis Using Hidden Markov Model and Active Shape Models”, *Journal of Information Science And Engineering*, 17, pp. 371-403, 2001.
- [10] M. Brand, N. Oliver, and A. Pentland, “Coupled Hidden Markov Models for Complex Action Recognition”, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 994–999, 1996.
- [11] A. F. Bobick, J. Davis, “Real-Time Recognition of Activity Using Temporal Templates”. *Proc. of IEEE CS Workshop on Applications of Computer Vision*, pp. 39-42, 1996.
- [12] M. Zimmermann, H. Bunke. “Hidden Markov Model Length Optimization For Handwriting Recognition Systems”, *Proc. of the 8th International Workshop on Frontiers in Handwriting Recognition*, pp. 369–374, 2002.
- [13] R. I. A. Davis, B. C. Lovell, and T. Caelli “Improved Estimation of Hidden Markov Model Parameters from Multiple Observation Sequences”, *International Conference on Pattern Recognition*, 2, pp. 168-171, August 2002.

- [14] P. M. Baggenstoss, “A modified Baum–Welch algorithm for hidden Markov models with multiple observation spaces”, IEEE Trans. Speech Audio Processing, 9, pp. 411–416, May 2001.
- [15] P. Somervuo, “Speech Recognition Using Context Vectors And Multiple Feature Streams”, Helsinki University of Technology, Master Thesis, 1996.
- [16] L. Xie, S-F. Chang, A. Divakaran, and H. Sun, “Structure analysis of soccer video with Hidden Markov Models”, International Conference on Acoustic, Speech and Signal Processing, May 2002.
- [17] J. J. Lee, J. Kim, and J. H. Kim, “Data-Driven Design of HMM Topology For Online Handwriting Recognition”, International Journal of Pattern Recognition and Artificial Intelligence, 15(1), pp. 107–121, 2001.
- [18] S. Tao, R. Guerin, “On-line Estimation of Internet Path Performance: An Application Perspective”, Proc. of IEEE INFOCOM, March 2004.

