

## 第二章 文獻探討

本章節主要以「文字資訊萃取技術」及「獨立成份分析法之原理」為兩大主軸來做探討。首先是介紹文字資訊萃取的定義，及其相關研究。接下來，將對獨立成份分析的演算方法以及處理過程，作一詳細的說明。

### 第一節 文字資訊萃取技術 (Text Information Extraction, TIE)

在影像中的文字通常包含了與影像內容義意相關的重要資訊，從影片檢索的角度來看，文字資訊的萃取是自動註解影像的重要關鍵。而若是要從影像中萃取出這些文字資訊(以下稱為 TIE 技術)一般會牽涉到某些程序[3]，其中包括：文字偵測(detection)、定位(localization)、追蹤(tracking)、萃取與強化(extraction, enhancement)以及文字辨認(recognition)。然而在很多的情況下會使得抽取影像中的文字變得相當困難，例如：字體的尺寸大小不一、字型的改變、文字的排列方式、低解析度之影像以及複雜的影像背景等，都是造成文字資訊不易萃取的主要原因。因此所謂的 TIE 技術便被提出來解決這些問題，而其中包含了許多的方法。在本章節中將概要的介紹文字資訊萃取技術的基本概念及其相關之應用。

#### 壹、文字資訊萃取(TIE)之定義

一般來說，所謂的 TIE 系統是去辨認出一張或一連串影像中的文字資訊[3]，而這些影像可以是灰階或彩色、壓縮或未壓縮、亦可是靜止或移動的文字。而 TIE 通常會牽涉到下列幾項問題：(1) 偵測(detection)，(2) 定位(localization)，(3) 追蹤(tracking)，(4) 萃取與強化(extraction, enhancement)以及(5) 辨認(recognition)，整個 TIE 系統如下圖 2.1 所示：

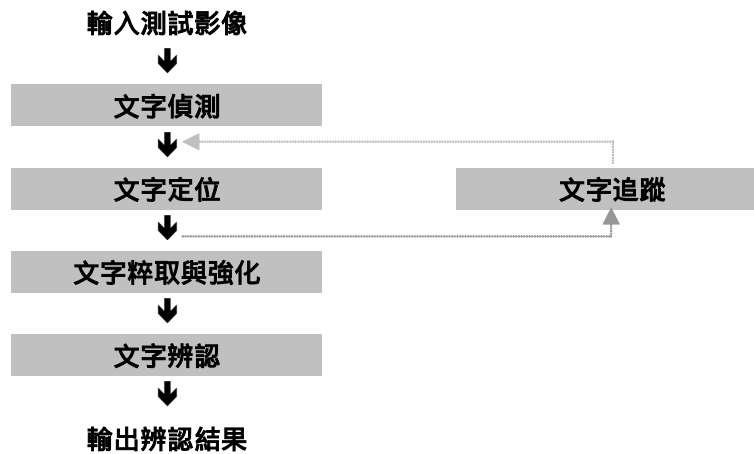


圖 2.1 TIE 系統之架構[3]

在文獻[8]中，Antani 等人為上述之術語做了以下幾項定義：

- 一、 文字偵測 (Text Detection)：通常是確定影像中是否有文字的程序，也就是粗略的將影像分成文字區及非文字區(見下圖 2.2-(b))。
- 二、 文字定位 (Text Localization)：此程序主要在決定文字在影像中的位置，並且會利用所謂的 bounding boxes 框選出文字區域(見下圖 2.2-(c))。
- 三、 文字追蹤(Text Tracking)：此一過程主要是在降低文字定位的處理時間，快速的找出文字的位置。
- 四、 文字萃取與強化(Text Extraction/ Enhancement)：是將文字從背景中分離的一個階段，並且將影像轉為二階的形式(白底黑字或黑底白字，如圖 2.2-(d))，而為了提高之後辨認的準確度會對低解析度的文字區域做強化的處理。
- 五、 文字辨認(Text Recognition)：萃取出來的文字影像，可利用光學文字辨認技術(Optical Character Recognition, OCR)轉換成真正的文字。


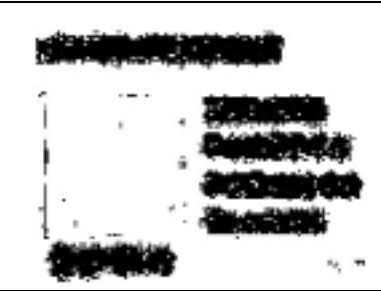

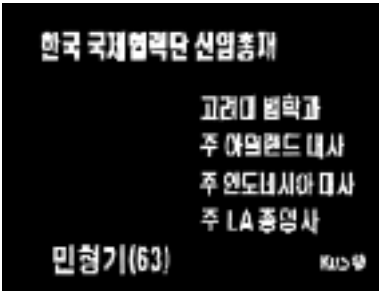
	
(a) 原始影像	(b) 文字偵測
	
(c) 文字定位	(d) 文字萃取

圖 2.2 文字偵測、定位、萃取之範例圖示 [9]

## 貳、文字資訊萃取技術之相關研究

從前段文章中，簡單的介紹了 TIE 技術中的幾項程序，本段落將繼續介紹這幾個程序的相關研究。在表 2.1 中回顧了從 1994-2005 近十年來與 TIE 技術的相關之研究相關的研究。

表 2.1：2003 – 2005 年 TIE 技術相關研究[3]

年份	作者	TIE 技術	使用方法	應用範圍
1994 [10]	Ohya, Shio, and Akamatsu.	localization recognition	thresholding, 與 relaxation operations	景物中之文字 (火車、招牌)
1995 [11]	Lee, and Kankanhalli.	localization recognition	connected component (CC) generation.	景物中之文字 (貨物)
1995 [12]	Smith, and Kanade.	localization	$3 \times 3$ filter	字幕文字
1995 [13]	Zhong, Karu, and Jain.	localization	CC-based method, variance-based method, 以及 hybrid method.	景物中之文字 (CD封面)
1996 [14]	Lienhart, and	recognition	split-and-merge algorithm	字幕文字

	Stuber		Dilation, motion analysis contrast analysis	
1996 [15]	Yeo, and Liu.	localization	large inter-frame difference	字幕文字
1998 [1]	Shim, Dorai, and Bolle	localization	Gray-level difference between pairs of pixels.	字幕文字
1998 [16]	Sato, Kanade, and Hughes	recognition	Smith and Kanade's localization method and recognition-based character extraction	字幕文字
1999 [17]	Agnihotri, and Dimitrova	Detection	Channel Separation Image Enhancement Edge Detection Edge Filtering	字幕文字
1999 [18]	Wu, Manmatha, and Riseman	recognition	multi-scale texture segmentation	影片、景物中之文 字 (報紙、廣告)
2000 [19]	Hasan, and Karam	localization	Morphological approach	景物中之文字
2000 [20]	Li, Doerman, and Kia	localization enhancement tracking	Wavelet-based feature extraction 與 neural network for texture analysis	景物中之文字
2000 [21]	Lim, Choi, and Lee	localization	DCT coefficient 與 macroblock type information	MPEG壓縮影像 字幕文字
2001 [22]	Hase, Shinokawa, Yoneda, and Suen	extraction	Color segmentation multi-stage relaxation Selection of character strings	書籍封面文字
2001 [23]	K. Jung	localization	Gabor filter -like multi-layer perceptron	彩色文件 字幕文字
2002 [24]	Strouthpoulos, Papamarkos, and Atsalakis	localization	color reduction	彩色文件
2003 [25]	Wanga, and Kangasb	location	Color clustering Aligning and merging analysis	景物中之文字
2003 [5]	Ye, Gao, Wang, and Zeng	detection	edge feature operation morphology operation wavelet 及 SVM classifier	字幕文字

2004 [26]	Hua, and Wenyin	detection	ground truthing	字幕文字
2004 [27]	Jung, and Han	extraction	mean shift algorithm (CAMShift) Connected component X-Y recursive cut	彩色影像
2005 [28]	Zheng, Zhao, and Wang	location	Image enhancement Vertical edge extraction Background curve and noise removing	景物中之文字 (車牌)
2005 [2]	Lyu, Song, and Cai	detection localization extraction	local thresholding 與 Hysteresis edge recovery	字幕文字

從文獻中可以看到 TIE 系統大致被應用於以下幾個層面，如圖 2.3-2.6 所示。像是圖 2.3 文件中的文字影像。再者，亦有像是圖 2.4 中書籍封面上之文字、圖 2.5 中影片字幕、或是圖 2.6 中景物中出現的文字，皆可運用 TIE 技術萃取影像中的文字並進行辨認。



圖 2.3 文件中之文字影像 [24, 27]



圖 2.4 書籍封面之文字影像 [22]



圖 2.5 影片中之文字影像 [27]



圖 2.6 景物中之文字影像[25, 28]

在文字萃取的實驗當中，最常被使用到的特徵為影像中的顏色資訊，像是在文獻[17]中 L. Agnihotri 等人認為影片中字幕的顏色通常為黃、白、黑三色，因此基於這個假設發展出一套能自動偵測影片字幕的方法，但也由於這樣的限制，此

方法並沒辦法適用於某些特殊的影片，例如 KTV 或是 MTV 影片中常出現的藍色字幕、以及男女對唱的綠色紅色字幕 就不適用 Agnihotri 所提出的方法。除了顏色特徵之外，像是利用 Gabor filter [23]、Wavelet [5]、FFT [29]、及 spatial variance[13] 等方式所偵測出的紋理(texture)特徵也是 TIE 技術常用的特徵，但是這些方法並不適用於過度單純的背景之下。再來就是所謂的邊緣(edge)特徵，此特徵主要是依據背景與文字呈現高反差的情況下來萃取文字[4, 17]，但是像網路下載的影片，通常會有解析度不高的問題，如此會造成字幕與背景的亮度資訊相差不夠大，在這樣的條件下，並不適合用高反差方法偵測文字。而利用分類器(classifier)來區分文字與非文字特徵也常使用在 TIE 技術中，像是[9]中，使用 SVM 分類器來偵測影片中的文字資訊。而分類器的方法的正確率常取決於訓練樣本(training samples)的數目以及種類，若是訓練樣本數目不足或是不夠多樣性很有可能造成偵測率不佳。再者，像是使用移動分析(motion analysis)的方式來分析文字及非文字區，這樣的方法通常是需要一段連續影像，才能對前後張影像作分析，進而區分出文字區域，換句話說，此方法無法處理單張影像中的文字。

從上述的文獻當中，可以得到一個結論：單一的方法通常不適用於不同情況下的文字萃取。因此傳統上，以結合多種方法的方式來提高文字萃取的正確性。例如[14]中 Lienhart 等人就使用了多種演算法來分離影像中的文字區域以及非文字區域。以下圖 2.7 來說，他們認為文字區域有相同或相似的色彩及尺寸，因此首先就利用分合演算法(split-and-merge algorithm)將影像區分成某些單色色塊(圖 2.7-(b))，接著限定文字影像的色塊大小，濾除太大或太小的色塊(如圖 2.7-(c))，再利用型態學上的擴張法(dilation)來強化萃取的效果(圖 2.7-(d))，之後如圖 2.7-(e)所示，為了增強文字萃取的結果，使用 block-matching 演算法以平均絕對誤差法則(mean absolute difference criterion)來做移動分析(motion analysis)，最後在藉由幾何分析將非文字區濾除掉(圖 2.7-(f))。

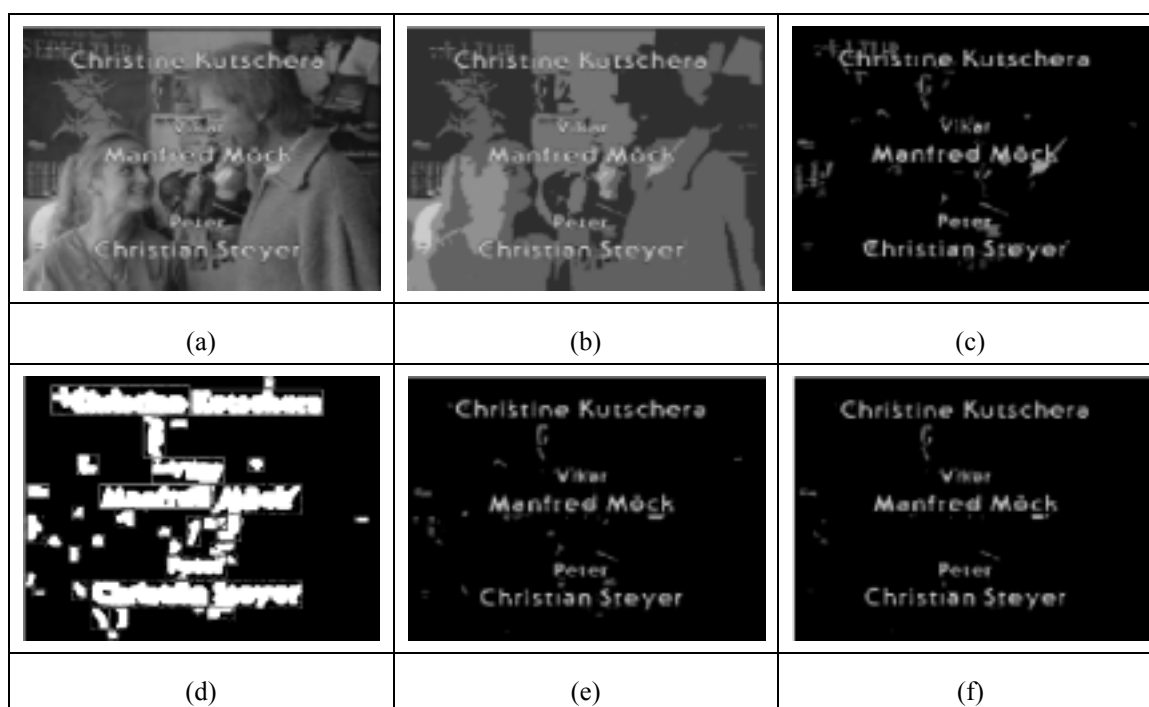


圖 2.7 Lienhart 等人提出的影片字幕抽取技術[14] (a) 原本的圖片(b) 使用分合演算法對影像做分割 (c) 色塊大小限制後的文字萃取結果(d) 經過二階化與型態學的擴張法之後(e) 移動分析結果(f) 反差分與長寬比限制

從上述幾項例子看來，單一的方法無法解決所有情況下文字的萃取，但結合多種方法又會有許多參數調整的問題，並且使得整個系統的處理程序變得相當複雜。為了解決上述問題，在本論文中提出了一個系統化的文字萃取方法，其中採用獨立成份分析法(Independent Component Analysis, ICA)，此方法可以處理大部分影像特徵，而且沒有複雜的參數調整，可以有效率解決文字萃取的問題。關於 ICA 的原理以及計算方法將於下一節中有詳細說明。

## 第二節、獨立成分分析法之原理

獨立成份分析法(Independent Component Analysis, ICA)是一種統計及計算的技術，主要是用來找出隨機變數或訊號中的隱藏因子(hidden factors)[30]。在 ICA 的模型(ICA model)中，假設觀察的多變量資料是由某些未知的潛在變數(latent variable)經由線性或非線性的方法混合而成[30]。這些潛在的變數被稱之為觀察資



料中的獨立成份(Independent Component)，並且假設他們為非高斯分佈且互相獨立。而 ICA 就是利用分析的方法找出這些獨立成份。ICA 這個方法最早用來解決神經生理學方面的問題[31]，之後被應用於 blind signal deconvolution 中的訊號處理[32]，到 1997 年，由 Aapo Hyvärinen 提出 FastICA [6]之後，由於其計算效率高，獨立成份的技術開始被廣泛的應用於各個領域，其中包括語音訊號處理(speech signal processing)[7]、資料分析(data analysis)[33]、電信學(telecommunication) [34]、及影像去雜訊(reduce noise) [35] 等領域，皆曾經使用 ICA 的技術來純化量測出來的混合訊號。

## 壹、ICA 模型的基本概念

ICA 這項技術最常被應用於語音訊號分離(blind signal separation)，而其中最有名的是解決「雞尾酒派對問題」(cocktail party problem)。所謂雞尾酒派對問題就是在同一個空間中，同時有很多聲音一起出現，縱使如此，在我們大腦構造與神經傳導的運作機制下，仍可以聽到想聽的聲音[33]。本段落將以聲源訊號為例，在假設聲源訊號是獨立的情況下，介紹 ICA 的運作方式，並且將之推廣至影像文字萃取的應用。

假設有兩個人  $s_1, s_2$  在同一空間中同時發出聲音，經過空間中混合(mix)過程之後，由兩支麥克風同時接收這兩個聲源，這兩支麥克風在不同時間所接收到的聲源訊號分別為  $x_1(t), x_2(t)$ ，為了簡化問題，在此只考慮聲源與所量測到的訊號為線性組合，其混合方式如公式 2.1 所示：

$$\begin{aligned}x_1(t) &= a_{11}s_1 + a_{12}s_2 \\x_2(t) &= a_{21}s_1 + a_{22}s_2\end{aligned}\tag{2.1}$$

其中  $a_{11}, a_{12}, a_{21}, a_{22}$  為放大倍率，取決於聲源與麥克風的距離。而在一般的情況下，通常只能取得混合訊號  $x_1(t), x_2(t)$  的數值，若是在  $a_{11}, a_{12}, a_{21}, a_{22}$  皆未知的狀況下想要取的最原始的聲源訊號  $s_1$  與  $s_2$  將變得相當困難。ICA 就是在只能知道混合訊號

$\mathbf{x}$  的情況下，藉由統計的分析，估計出未混合前的原始的訊號  $\mathbf{s}$ ，而 ICA 的基本模型如以下公式 2.2 所示，其中， $\mathbf{A}$  為  $\mathbf{x}$  與  $\mathbf{s}$  之間的混合矩陣。

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (2.2)$$

接下來，根據公式 2.2 進一步的將這個問題擴大為有  $m$  個麥克風來接收  $n$  個聲源，可得到下列公式 2.3：

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} = \mathbf{A}\mathbf{s} \quad (2.3)$$

其中  $a_{ij}$  代表第  $j$  個聲源訊號  $s_j$  在第  $i$  個麥克風  $x_i$  的混合權重。ICA 技術就是在只有混合訊號  $\mathbf{x}$  已知的情況下，去找一個被稱之為 demixing matrix 的矩陣  $\mathbf{w}$ ，經過轉換後，得到原來的聲源訊號(見公式(2.4))。

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \approx \mathbf{s} \quad (2.4)$$

估計  $\mathbf{w}$  主要是利用聲源訊號與訊號之間呈現獨立的特性，換句話說，就是經由判斷  $\mathbf{y}$  中每個訊號是否獨立來決定  $\mathbf{w}$ ，如果此  $\mathbf{w}$  轉換出來的  $\mathbf{y}$  是彼此獨立的，就將  $\mathbf{y}$  當作是原來彼此獨立的訊號源，這就是 ICA 的基本觀念。

## 貳、ICA 之限制

在使用 ICA 演算法之前，要注意此演算法本身有一些限制，以下說明幾項 ICA

的限制：

## 一、對原始訊號(未混合前之訊號)之限制

### (一) 原始訊號在統計上獨立(independent)

ICA 的演算法的基本條件就是利用原始訊號彼此獨立的特性所發展出來的，因此若要使用 ICA 來純化混合訊號，原始的訊號就必須具有獨立的特性。而定義獨立的概念，首先會考慮兩個純量的隨機變數  $y_1$  與  $y_2$ ，基本上若是  $y_1$  與  $y_2$  或是  $y_2$  與  $y_1$  沒有任何關係，就稱  $y_1$  與  $y_2$  相互獨立。以統計的觀點，獨立可以利用 probability density 來定義，當隨機變數  $s_1, s_2, \dots, s_n$  是獨立的，其 joint probability density function 可拆為 marginal probability density function 的乘積(見以下公式 2.5)。

$$P(s_1, s_2, s_3, \dots, s_n) = P(s_1) P(s_2) P(s_3) \cdots P(s_n) \quad (2.5)$$

其中， $p_i(s_i)$  為  $s_i$  之 marginal probability density function

### (二) 獨立元素必須為非高斯分佈(non-Gaussian distributions)

非高斯分佈的獨立元素是在估計 ICA 模型中一個重要的關鍵，最主要的原因是 ICA 在判定元素之間是否獨立，是利用所謂的中央極限定理(Central Limit Theorem)，此定理中說明了：兩個獨立隨機變數的總合，通常比原始隨機變數更接近高斯分佈。根據這個原理以及公式 2.4，若是能找出一組  $w$ ，此  $w$  轉換的結果使得  $y$  最不接近高斯分佈時，也就相當於找出原來的獨立訊號。基於這個原因，會希望各個原始訊號  $y_i$  儘可能不是高斯分佈。

## 二、對混合訊號與獨立訊號數目的限制

假設有  $m$  個混合訊號訊  $x$  和  $n$  個原始訊號  $s$ ，其中  $x = As$ ， $A$  為混合矩陣。若  $m = n$ ，則  $s = A^{-1}x$ ，有解；若  $m > n$ ，則方程式比未知數多，在這種情形之下，可先將維度降回  $n$ ，即可求解；若是  $m < n$ ，則方程式比未知數少，在這樣的情況下，便沒有足夠的條件找出未知的獨立元素。

### 三、ICA 方法的不確定性

在  $\mathbf{x} = \mathbf{A}\mathbf{s}$  的 ICA 模型中可以看出以下兩點不確定性

(一) 無法決定獨立成份的 variance :

這種不確定性是由於獨立成份  $\mathbf{s}$  與混合矩陣  $\mathbf{A}$  皆為為之所造成的。根據 ICA 模型，第  $i$  個混合訊號可表示成如下公式 2.6 :

$$x_i = \sum_{j=1}^n a_{ij} s_j \quad (2.6)$$

假設  $s_i$  被放大  $\alpha_j$  倍，則  $a_{ij}$  只要除以  $\alpha_j$  就可以互相抵銷，如以下公式 2.7 :

$$x_i = \sum_{j=1}^n (a_{ij} / \alpha_j) (s_j \alpha_j) \quad (2.7)$$

因此假設原始獨立訊號的 variance = 1，如以下公式 2.8 :

$$E\{s_i^2\} = 1 \quad (2.8)$$

(二) 無法確定獨立成份的順序 :

這點不確定性也跟未知的  $\mathbf{A}$  與  $\mathbf{s}$  有關，由於每個混合訊號的次序是可以任意的被掉換的，因此無法定義某個獨立訊號是第幾個。換句話說，ICA 所估計出來的轉換矩陣  $\mathbf{w}$  所求得的獨立成份，與原始獨立訊號的順序是不相同的。

### 參、ICA 演算法

#### 一、ICA 演算法之架構

整個 ICA 演算法的基本架構會如下圖 2.8 所示。為了簡化運算的複雜性，通常會先將取得的混合訊號  $\mathbf{x}$  執行 centering 與 whitening 兩個前處理，得到一組新的混合訊號  $\mathbf{z}$ 。之後定義一個目標函數(object function)，然後對此目標函數設計出一

個合適的最佳化演算法，藉此得到獨立成份  $y$ 。

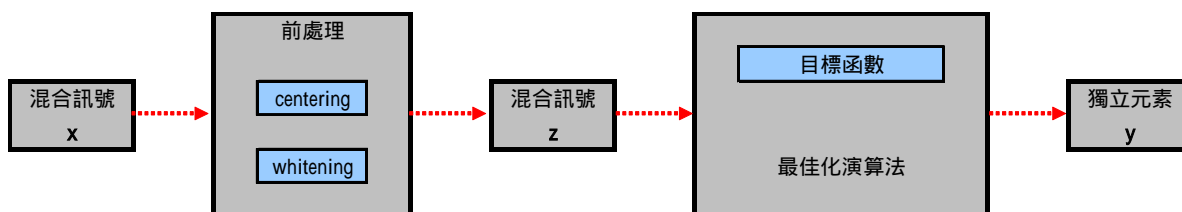


圖 2.8 ICA 演算法之架構

## 二、ICA 之前處理 [6]

### (一) Centering

以公式 2.3 來說，若是接收到的混合訊號與原始的獨立訊號的平均為零 (zero-mean)，則在演算法的推導過程中可以簡化許多，因此 centering 這項前處理最主要的目的就是將混合訊號  $x$  中的每個訊號  $x$  減去其平均值  $E(x)$ ，如此一來，便可以得到一個新的訊號  $x_c$ ，且  $x_c$  中每個訊號的平均為零。

$$\mathbf{x}_c = \mathbf{x} - E\{x\} \quad (2.9)$$

### (二) Whitening

在 ICA 的流程中，另一個非常重要的前處理程序為 whitening，此過程可以大量的降低分析的參數數目，對於簡化 ICA 問題有很大的幫助。以統計的觀點來看，比獨立(independent)弱一點的形式是非相關(uncorrelatedness)，而獨立與非相關之間的關係便是 whiteness。當混合的訊號  $x$  經過 centering 得到  $x_c$ ，若是再經過 whitening 的線性轉換(見公式 2.10)，便會得到一組新的訊號  $z$ ，此訊號的 variances 將等於 1(見公式 2.11)，換句話說，也就是  $z$  的 covariance matrix 是 identity matrix。

$$\mathbf{z} = \mathbf{V}\mathbf{x}_c \quad (2.10)$$

$$E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I} \quad (2.11)$$

Whitening 最常使用的方法是對訊號的 covariance matrix  $E\{\mathbf{x}_c\mathbf{x}_c^T\}$  做 eigen-value decomposition (EVD)

$$E\{\mathbf{x}_c\mathbf{x}_c^T\} = \mathbf{E}\mathbf{D}\mathbf{E}^T \quad (2.12)$$

其中  $\mathbf{E}$  是由  $E\{\mathbf{x}_c\mathbf{x}_c^T\}$  的 eigenvector 所構成的正交矩陣， $\mathbf{D}$  是由其相對應的 eigenvalue 所組成的對角線矩陣(diagonal matrix)。所以整個 whitening 的結果可以被表示成如以下公式 2.13：

$$\mathbf{z} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{x}_c \quad (2.13)$$

其中  $\mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T$  為 whitening matrix，也就是公式 2.10 中的  $\mathbf{V}$ 。

執行 whitening 之後，再參考公式 2.2 中的 ICA 模型，可將混合矩陣  $\mathbf{A}$  轉換成另一個新的矩陣  $\tilde{\mathbf{A}}$ ，如下公式 2.14。

$$\mathbf{z} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{A}\mathbf{s} = \mathbf{V}\mathbf{A}\mathbf{s} = \tilde{\mathbf{A}}\mathbf{s} \quad (2.14)$$

若將 whitening 後的訊號之 covariance matrix 以(2.14)展開可得如以下公式 2.15。

$$E\{\mathbf{z}\mathbf{z}^T\} = E\{\tilde{\mathbf{A}}\mathbf{s}\mathbf{s}^T\tilde{\mathbf{A}}^T\} = \tilde{\mathbf{A}}E\{\mathbf{s}\mathbf{s}^T\}\tilde{\mathbf{A}}^T \quad (2.15)$$

因為原始訊號  $\mathbf{s}$  的平均值等於零且 variance 為 1，而此訊號間又彼此獨立因此：

$$E\{\mathbf{s}\mathbf{s}^T\} = \mathbf{I} \quad (2.16)$$

所以

$$E\{\mathbf{z}\mathbf{z}^T\} = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^T = \mathbf{I} \quad (2.17)$$

這意味著經過 whitening matrix  $\mathbf{V}$  轉換過後之混合矩陣  $\tilde{\mathbf{A}}$  是一個正交矩陣。這也就表示在尋找 demixing matrix  $\mathbf{w}$  時，只要尋找互相垂直的向量即可，這對整個演算法的簡化有相當大的幫助。

### 三、 目標函數 (objective function)

根據中央極限定理，使用 ICA 方法所估計的  $\mathbf{w}$  所轉換出來的獨立成份  $\mathbf{y}$  是否正確(見公式 2.18)，通常與此成份是否呈現非高斯分佈有很大的關係，因此必須要有一種量測的方法可以用來測量  $\mathbf{y}$  的高斯程度，一般最典型的量測方法為 kurtosis 以及 neg-entropy。介紹這兩個方法之前，在此先假設  $\mathbf{y}$  的平均為零以及 variance 等於 1。

$$\mathbf{y} = \mathbf{w}^T \mathbf{z} \quad (2.18)$$

#### (一) Kurtosis

基本上， $\mathbf{y}$  的 kurtosis 定義如下公式 2.19

$$kurt(\mathbf{y}) = E\{\mathbf{y}^4\} - 3(E\{\mathbf{y}^2\})^2 \quad (2.19)$$

假設當  $\mathbf{y}$  已經被調整成 variance 等於 1 時，公式 2.19 可被簡化成公式 2.20。

$$kurt(\mathbf{y}) = E\{\mathbf{y}^4\} - 3 \quad (2.20)$$

這也就表示當 kurtosis 被簡化成公式 2.20 時，此算式與  $E\{\mathbf{y}^4\}$  有極大的關係。若是

$y$  呈現高斯分佈，其  $E\{y^4\}$  等於  $3(E\{y^2\})^2$ ，因此對於呈現高斯分佈的隨機變數其 kurtosis 將等於零。kurtosis 可以是正值或負值，若是隨機變數的 kurtosis 大於零，稱之為 sub-gaussian, 反之，若是小於零則稱之為 super-gaussian, 也就是說當 kurtosis 不為零，估計出來的隨機變數  $y$  將呈現非高斯分佈。綜合以上所述，kurtosis 的平方值或絕對值可以做為目標函數，此函數越小代表越接近高斯分佈；越大則越接近非高斯分佈。

然而，kurtosis 方法在實務的應用上有其缺點，最主要的問題來自於 4 次方的量測，在如此高次方的計算過程中，所得到的數值將會受到某些少數值的影響，換句話說，以量測隨機變數的高斯程度而言，kurtosis 並非是一個強韌性(robust)佳的方法。因此，以下提出 negentropy 及其近似形式(approximations of negentropy) 來解決高斯量測上的問題。

## (二) Negentropy

Negentropy 主要是根據資訊理論(information theory)所發展出來的量測方法，若是隨機變數的 entropy 越大，代表所觀察的變數是越無法預期的。對於一個 probability density function 為  $f(\mathbf{y})$  的隨機向量  $\mathbf{y}$ ，它的 entropy  $H$  定義如以下公式 2.21：

$$H(\mathbf{y}) = -\int f(\mathbf{y}) \log f(\mathbf{y}) d\mathbf{y} \quad (2.21)$$

以資訊理論的觀點，具有相同 variance 的隨機變數當中，呈現高斯分佈的變數將有很大的 entropy，因此為了得到一個非高斯量測方法，會希望一個呈現高斯分佈的隨機變數其函數值為零，且對於所有隨機變數之函數值皆大於零，於是定義了以下公式 2.22 的量測函數  $J$ ，稱之為 negentropy。

$$J(\mathbf{y}) = H(\mathbf{y}_{\text{gauss}}) - H(\mathbf{y}) \quad (2.22)$$



其中  $\mathbf{y}_{\text{gauss}}$  是與  $\mathbf{y}$  具有相同 covariance matrix 的高斯隨機向量。根據以上公式 2.22，由於高斯變數的 entropy 最大，因此對於隨有隨機變數的 negentropy 皆為正值，而當  $\mathbf{y}$  呈現高斯分佈時，函數值將為零。換句話說，只要找出 negentropy 的最大值，就是找出呈現最非高斯變化的隨機向量。然而，negentropy 在計算上十分複雜，為了提高估計的方便性，通常會使用 negentropy 較簡化的近似，而此方法將於下個段落中作詳細的說明。

### (三) Negentropy 近似形式(approximations of negentropy)

對於 negentropy 近似形式，典型的做法如以下公式 2.23[36]

$$J(\mathbf{y}) \approx 1/12 E\{\mathbf{y}^3\}^2 + 1/48 \text{kurt}(\mathbf{y}^2) \quad (2.23)$$

但公式 2.23 高次計算過程，同樣會遇到不夠強韌的問題，為了避免這種狀況，Hyvärinen [37]根據 maximum-entropy 理論，發展出如以下公式 2.24 的近似形式：

$$J(\mathbf{y}) \approx \sum k_i [E\{G_i(\mathbf{y})\} - E\{G_i(\mathbf{v})\}]^2 \quad (2.24)$$

公式中的  $k_i$  為大於零的常數， $\mathbf{y}$  為隨機變數， $\mathbf{v}$  為高斯隨機變數，其中  $\mathbf{y}$  與  $\mathbf{v}$  平均皆等於零且 variance 皆為 1，而函數  $G_i$  為某種 nonquadratic 函數，有以下公式 2.25 三種選擇，其中  $1 \leq a_1 \leq 2$  是較適合的常數，通常取  $a_1 = 1$ 。

$$\begin{aligned} G_1(\mathbf{y}) &= \frac{1}{a_1} \log \cosh a_1 y \\ G_2(\mathbf{y}) &= -\exp(-y^2 / 2) \\ G_3(\mathbf{y}) &= y^4 \end{aligned} \quad (2.25)$$

在選定  $G$  函數之後，公式 2.24 可以被改寫成如以下公式 2.26：

$$J(\mathbf{y}) \propto [E\{G(\mathbf{y})\} - E\{G(\mathbf{v})\}]^2 \quad (2.26)$$

尋找公式 2.26 中  $J(\mathbf{y})$  的最大值，就表示尋找出呈現最非高斯分佈的隨機變數。

#### 四、FastICA 演算法

在前面的段落中，介紹了數種量測高斯程度的方法，簡單的說，就是用來估計 ICA 模型的目標函數，接下來就必須為所選擇的目標函數設計出一個最佳化的演算法。FastICA 是由 Aapo Hyvärinen 所提出之演算法[6]。在這個演算法中利用公式 2.26 做為目標函數，假設欲量測之  $\mathbf{y} = \mathbf{w}^T \mathbf{z}$ ，故 2.26 式可改寫成以下公式 2.27：

$$J(\mathbf{w}^T \mathbf{z}) \propto [E\{G(\mathbf{w}^T \mathbf{z})\} - E\{G(\mathbf{v})\}]^2 \quad (2.27)$$

公式 2.27 的最大值可藉由尋找  $E\{G(\mathbf{w}^T \mathbf{z})\}$  之極值得到，FastICA 是根據 fixed-point iteration 來找出  $J(\mathbf{w}^T \mathbf{z})$  的最大值[38, 39]。根據 Kuhn-Tucker conditions[40]，在  $E\{(\mathbf{w}^T \mathbf{z})^2\} = \|\mathbf{w}\|^2 = 1$  之 constraint 之下， $E\{G(\mathbf{w}^T \mathbf{z})\}$  發生極值之  $\mathbf{w}$  需滿足公式 2.28：

$$E\{\mathbf{z}g(\mathbf{w}^T \mathbf{z})\} + \beta \mathbf{w} = 0 \quad (2.28)$$

其中  $g$  為  $G$  之微分，接下來利用牛頓法來解出公式 2.28。將公式 2.28 左式表為  $F$ ，可以得到如公式 2.30 的 Jacobian matrix  $JF(\mathbf{w})$ 。

$$JF(\mathbf{w}) = E\{\mathbf{z}\mathbf{z}^T g'(\mathbf{w}^T \mathbf{z})\} - \beta \mathbf{I} \quad (2.29)$$

而對於 whitening 後之訊號，可做以下近似，如公式 2.30

$$E\{\mathbf{z}\mathbf{z}^T g'(\mathbf{w}^T \mathbf{z})\} \approx E\{\mathbf{z}\mathbf{z}^T\} E\{g'(\mathbf{w}^T \mathbf{z})\} = E\{g'(\mathbf{w}^T \mathbf{z})\} \mathbf{I} \quad (2.30)$$

接下來利用牛頓法，可得到以下公式 2.31 的 iteration：

$$\mathbf{w}_{n+1} = \mathbf{w} - [E\{\mathbf{z}g(\mathbf{w}^T \mathbf{z})\} - \beta \mathbf{w}] / [E\{g'(\mathbf{w}^T \mathbf{z})\} - \beta] \quad (2.31)$$

最後將等式乘上  $\beta - E\{g'(\mathbf{w}^T \mathbf{z})\}$  可將公式 2.31 簡化成公式 2.32

$$\mathbf{w}_{n+1} = E\{\mathbf{z}g(\mathbf{w}^T \mathbf{z})\} - E\{g'(\mathbf{w}^T \mathbf{z})\} \mathbf{w} \quad (2.32)$$

若是  $\mathbf{w}_{n+1}$  與  $\mathbf{w}_n$  同方向即為收斂。假設有  $p$  個需要分析的獨立成份  $\mathbf{y}$ ，就必須取得  $p$  個 demixing matrix  $\mathbf{w}$ ，而  $\mathbf{y}$  經過 whitening 後，要尋找  $\mathbf{w}$  時，只要尋找互相垂直的向量即可，因此在尋找第  $p+1$  個  $\mathbf{w}$  時，需將前  $p$  個找出  $\mathbf{w}$  之方向減去，根據 fixed-point 演算法，可得到下列公式 2.33 的 iteration：

$$\mathbf{w}_{p+1} = \mathbf{w}_{p+1} - \sum_{j=1}^p (\mathbf{w}_{p+1}^T \mathbf{w}_j \mathbf{w}_j) \quad (2.33)$$

綜合以上所述，FastICA 演算法之流程如下表 2.2：

表 2.2 FastICA 演算法流程

Step 1	輸入混合訊號 $\mathbf{x}$
Step 2	將訊號 $\mathbf{x}$ 作 centering，得到平均為零的訊號 $\mathbf{x}_c$
Step 3	將訊號 $\mathbf{x}_c$ 作 whitening，得到 variance 為 1 的訊號 $\mathbf{z}$
Step 4	設定 $p = 1$ ， $m$ 為獨立成份的個數
Step 5	隨機選擇一初始之 $\mathbf{w}_p$
Step 6	$\mathbf{w}_{p+1} = E\{\mathbf{z}g(\mathbf{w}_p^T \mathbf{z})\} - E\{g'(\mathbf{w}_p^T \mathbf{z})\} \mathbf{w}_p$ , $\mathbf{w}_p = \mathbf{w}_{p+1} / \ \mathbf{w}_{p+1}\ $
Step 7	減去先前已找到之方向 $\mathbf{w}_{p+1} = \mathbf{w}_{p+1} - \sum_{j=1}^p (\mathbf{w}_{p+1}^T \mathbf{w}_j \mathbf{w}_j)$ , $\mathbf{w}_p = \mathbf{w}_{p+1} / \ \mathbf{w}_{p+1}\ $
Step 8	若 $\mathbf{w}_p$ 尚未收斂，回到第 6
Step 9	設定 $p = p + 1$ 。若 $p < m$ ，回到第 4