# 國立交通大學

## 資訊科學與工程研究所

## 碩 士 論 文

基於 AdaBoost.MH 之模糊化文件分類法

Document Classification based on Fuzzy AdaBoost.MH

研 究 生：方士元

指導教授：李嘉晃　教授

中 華 民 國　一 百 年 六 月

# 基 於 AdaBoost.MH 之 模 糊 化 文 件 分 類 法
# Document Classification based on Fuzzy AdaBoost.MH


研 究 生：方士元　　Student：Shih-Yuan Fang

指導教授：李嘉晃　　Advisor：Chia-Hoang Lee


國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年六月

# 基 於 AdaBoost.MH
# 之 模 糊 化 文 件 分 類 法

學生：方士元　　指導教授：李嘉晃 教授

國立交通大學資訊學院 資訊科學與工程研究所碩士班

## 摘要

本論文中，我們提出了一個 Fuzzy AdaBoost.MH 演算法，而且將此 Fuzzy AdaBoost.MH 方法運用在文件分類上。Boosting 的主要觀念為利用許多 weak hypotheses，透過 Boosting 架構得到這些 weak hypothesis 權重，最後將這些 weak hypotheses 予以合併，形成一個高準確度的強分類法。我們使用 fuzzy rule 作為 weak hypothesis，利用 decision stump rule 為基礎的方法來當作我們判別的依據，而每一個 fuzzy rule 則是以文件中的 term 為依據。在文件特徵表示法中，每一個 n-gram term 常作為文件最基本的特徵;然而每一文件所包含的 n-gram 數目常會是一個巨大的數量，因此在系統的設計中，我們使用 term 出現的頻率來當作 term 篩選的方法，並且將通過篩選的 term 放入我們的 rule pool 中。每一回合，Fuzzy AdaBoost.HM 從 rule pool 中挑選出最好的 fuzzy rule，所有 fuzzy rule 的集合則是系統分類的依據。

同時，我們提出了一個 Fuzzy Number 的表示法，來表示每一條 fuzzy rule 的信心度。這些 fuzzy rule 的信心度訊息是我們做為推論分類結果的依據。當訓練的過程結束之後，我們可以經由程度轉化的過程推論我們最後的模糊化分類結果。本論文中也使用了三種文章集進行實驗，而在實驗的數據中，Fuzzy AdaBoost.MH 皆能有不錯的分類結果。

# Document Classification based on Fuzzy AdaBoost.MH

Student：Shih-Yuan Fang　　Advisor：Prof. Chia-Hoang Lee

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University

## Abstract

In this paper, we propose a fuzzy AdaBoost.MH algorithm and apply fuzzy AdaBoost.MH to document classification domain. The main idea of boosting is to generate many, relatively weak hypotheses and to combine these weak hypotheses into a single highly accurate classifier. In rule design, we employ decision stump rule as the basic discriminative function and each rule is correspondent to a weak hypothesis. In system design, we employ term frequency as filtering criterion to construct a rule pool. On each round, the best fuzzy rule can be selected from the pool using AdaBoost framework.

Meanwhile, we propose a fuzzy number representation to represent each rule's confidence. These fuzzy rules with confidence information are the bases of classification inference. When the training phase is completed, the final fuzzy classification result can be obtained from the inference result with a degree transformation process. The experimental results show that fuzzy AdaBoost.MH works very well in three data corpora.

# 誌謝

首先，感謝指導教授李嘉晃老師對我的悉心指導，才能有今日的成果。老師就像我的良師益友，時而嚴厲，時而慈祥，不論是研究討論或課堂授課時，所教導我的專業知識和處世道理，都著實讓我獲益良多。這些過程與經驗，都將成為我一生受用無窮的寶庫。接著要感謝三位辛苦的口試委員，陳柏琳教授、張嘉惠教授與張道行教授，謝謝教授們的建議，讓本論文的內容能夠更加完整。

同時，我亦感謝這兩年來陪伴在我身邊的實驗室同學們、學長以及學弟。尤其是我的同學們，智愷、俊憲、而益，總是不斷的鼓勵我，對我的幫助更是多不勝數。兩年的時間，雖然不是很長，但是曾經有過的歡笑淚水，這些回憶會一輩子永存在我的心中。

最後，我要感謝我的爸爸、媽媽、姐姐，感謝你們對我的愛護和包容。謝謝你們在背後默默的支持，使我能夠順利的完成碩士學位。

心中有太多的感謝不知道如何表達，在此僅以本篇論文表示我對你們最誠摯的感謝，並祝福你們身體健康、萬事如意，謝謝。

方士元　謹誌

資訊科學與工程研究所

智慧型系統實驗室

中華民國一百年七月

# 目錄

# 圖目錄

# 表目錄

# 第一章、緒論

## 1.1 研究動機

近年來，在這資訊蓬勃發展和思想自由發達的時代，伴隨著我們的是諸多的文字和資訊，如何在如此龐大的文章中，快速且有效的對文章進行分析、分類，是一個相當大的難題。近年來，越來越多研究使用機器學習方法來進行文件分析或分類;在實際應用上，除了時間之外，如何提高文章分類的準確度，也都是值得我們研究的議題。

由於時間上的累積，人們寫作的文章數量越來越龐大，以至於各種不同類別的文章數量相當龐大，而且每天無時無刻都在增加當中。若是要將新的一篇文章分析出是屬於哪一個類別，如果是由人工來進行分類，那麼，遇到龐大的資料量或是類別時，將需要耗費相當多的時間;另外文章內容可能很冗長，或是用詞模糊不清，此外每個人的看法與觀點也不一定相同，主觀的想法也會造成分類錯誤的發生，進而演變成人力資源上的浪費。對於現代的社會，有很多種類的文章，例如電影影評的文章，或是公司的問卷調查，都是需要在少量的時間下就必須要將所有的文章分類出來，因為電影公司或是有些電影網站需要知道某一部電影在社會上一般人的觀後感;或著是對於公司的問卷，我們想要知道關於此次的問卷內容經由填寫者填出來的結果是偏向正面評價或是負面評價。就以上的兩個簡單的例子中，文件分類已經成為一樣重要的研究;所以本論文希望發展一套系統，能夠同時兼顧高準確度以及加快機器分類的時間。

一般來說，使用模糊化的技術在文件分類上，能夠擁有較高的文件分類準確度，也能夠提升原先分類法的準確度。而在資訊量較小的情況下，一般分類法往往無法訓練出一個良好的分類模型，分類結果沒有辦法有滿意的結果;加入模糊化技術之後，有機會可以提高更多的準確度。所以本論文加入模糊化技術，希望能夠比傳統分類法有較好的分類結果。

## 1.2 研究目的與方法

本論文希望能將各種不同類別的文章，有效率的分類到正確的類別中。本論文考慮到各種不同領域以及類別的文章，在論文的實驗中，將會考慮在不同的文章集下以及同一文章集但是不同排列組合下的情況，另外本研究會與其他分類法作比較。

相對於傳統的監督式學習方法，Ensemble Learning[1]結合了多個不同權重的分類器去解決各種不同的分類問題。Ensemble Learning 主要是用來改善分類或是預測、效能的一個模型。而目前在所有以 Ensemble 為基礎的演算法中最有名的為 AdaBoost[2]分類法，它的各種演算法的變化已經運用在各種不同的領域中，也都具有不錯的成效[3][4][5][6][7]。

本論文方法部分是以 AdaBoost.MH[8]為基礎，結合模糊化的技術，形成一個更準確且新穎的分類法。會使用 AdaBoost.MH 主要原因為，它可以對多類別以及多標籤文章進行分類，加上訓練時間也較有些分類法來的快，除此之外，也具有不錯的準確度，所以在時間以及準確度兩者皆要兼顧的考量上是個不錯的分類法。而模糊化的技術自從 20 世紀開始，就一直是個讓許多研究者所研究的議題，在傳統的自然語言處理上，往往會有許多的資訊是屬於不確定的、不準確的，但是在二元的邏輯思維模式中，要解決這些問題，是有一定的困難和挑戰性，往往到了最後只能以隨機的或是猜測的方式來對這些資訊進行處理。模糊理論用在分類的方法上，是一種能夠更有效提高準確度的方法。模糊理論跟傳統的分類方法中，在本質上有些的不同，模糊理論是屬於多元邏輯，代表說除了二元邏輯的非真即假觀念，還多了有漸進的值，不再是只有 0 或 1 而已。結合 AdaBoost.MH 和模糊化的技術，擴大了模糊理論在自然語言領域的運用，是一個新穎的想法，但是兩者方法的結合，是否有比原先 AdaBoost.MH 的準確度更高，又或著是在時間上是否需要花費更多的時間，都是本論文中所需要研究的議題。本論文的目的主要是要將模糊化的技術運用在分類的方法上，可以利用模糊化技術提升本來方

法的效能之外;達到比傳統其他的方法在時間和準確度上都有更好的優勢。

本論文也嘗試了將本系統演變為 Semi-Supervised 的形式，稱為 Semi-Boosting，只需要提供少量的 training data 給系統作學習，就能將大量的文章資料進行分類。不同於傳統的 Semi-Supervised 分類法對於文章的分類只進行一個循環，代表只有一次的 Training 以及一次的 Testing，本系統 Semi-Boosting 將會對所要分類的文章集進行多次的循環，經由多次的循環 Training 以及 Testing，能夠在準確度上有提高的機會。

實驗比較的部分，其他比較著名的機器學習的演算法，例如，Support Vector Machine(SVM)[9][10]、以及 Naïve Bayes[11]，本論文也用了相同的文章集去實作了這些方法當作我們比較的數據。文章集皆為英文，本論文使用了 Pang 的電影影評文章集、20-Newsgroups、和 Reuters，其實驗的設計以及實驗的數據在本論文中會詳細介紹。

## 1.3 論文架構

第一章：緒論，簡單的介紹論文研究的動機，以及探討研究的目的與方法。

第二章：相關研究，概述本論文中所使用的技術背景知識，以及其他關於分類方法的研究和探討。

第三章：系統設計，將本研究之系統架構與演算法作一個詳細的介紹。

第四章：實驗過程與結果討論，包含實驗的設計以及其它分類法的比較數據。

第五章：結論與未來展望，對本論文的研究作總結以及統整，並且提出結論與未來本系統可以改善和研究的方向。

# 第二章、相關研究

## 2.1 模糊理論

模糊邏輯，自從 Lotfi Zadeh[12][13]教授在 1965 年發表之後，就被廣泛的使用在各種層面上，從控制理論到人工智慧等等。傳統的邏輯問題中，用所謂的二元邏輯來解決問題，而在我們所生活的現實世界中，充滿著太多的不確定性以及不明確性，無法單單使用 true 或 false 來解決所有的問題。在模糊邏輯的技術上，對於二元邏輯無法解決的問題，可以有程度的概念，不用再拘泥於傳統的思維。模糊邏輯的基礎是由許多的 if-then 結構所形成的，if 部份就是條件的前提部分，可以辨別這個敘述句是否成立，而 then 則是這個敘述句成立時，所要做出的結論或是回應。這些模糊法則一般是由專家經驗或經由訓練樣本所建立，但是缺點就是這些模糊法則無法自動從資料中學習而得到，因此近年來有許多結合模糊邏輯與機器學習的模型，希望可以同時具備模糊邏輯的彈性與機器學習的自動學習能力。神經模糊系統(Neuro-fuzzy systems)融合了模糊邏輯和神經網路，結合推論式模糊系統的彈性與神經網路的學習能力，在近二十年當中，眾多的神經模糊系統被廣泛的使用在控制領域中[14][15]。

此外，在神經網路當中，越來越多的研究者採用演化式計算的演算法或是 AdaBoost 從資料中進行學習[16][17][18][19][20][21]。Scherer[21]建立了一個 AdaBoost 結合神經模糊關係性的分類法。Del Jesus 等人[16]設計了演化式計算的 boosting 演算法，他們將每一條的模糊規則視為一個 weak hypothesis，而每一個模糊規則庫可以解釋為 weak hypothesis 以及權重的結合。基本上，他們的分析結果符合 weak hypothesis 的設計原則，也就是說 weak hypothesis 可以是任何類型的分類器。Otero 與 Sanchez[20]與 Del Jesus 等人[16]的方法類似，但是他們採用 Logitboost[22]演算法去解決模糊規則在分類上的問題。

而在所謂的集合方面，包含了明確集合與模糊集合，在模糊邏輯的領域中，我們所使用到的集合為模糊集合，明確集合簡單來說，就只是有或是沒有，例如

有一個明確集合{a, b, c}，那這個集合有沒有包含 a?有沒有包含 e?前者的答案為有，後者為沒有，這就是一個很簡單的明確集合的例子。而模糊系統中所使用到的模糊集合，由圖 2-1 就代表了一個模糊集合的例子，它的論域 U 的範圍界在 2 到 3 之間，對應的歸屬度函數為 $0.1x^2$，一般的情況下歸屬度(membership grade)會界定在 0 和 1 之間，而 x 稱作元素。

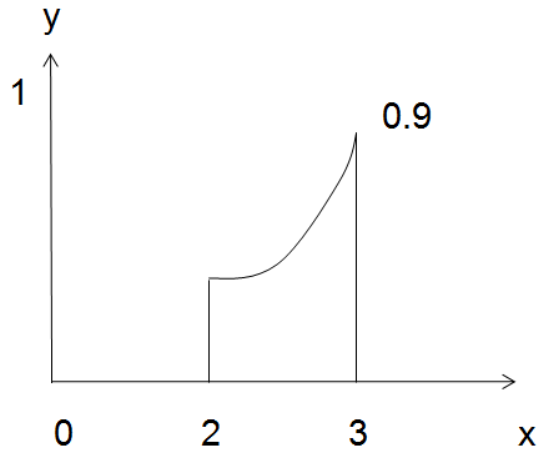


圖2-1. 模糊集合

模糊集合的基本運算，也可以分成五種，在此我們假設 A 和 B 為模糊集合，論域為 $U_X$，其歸屬函數分別為 $f_A(x), f_B(x)$，且 $\forall x \in U_X$。

(1)補集(complement)

$$B = A^c \Leftrightarrow f_A(x) = 1 - f_B(x)$$

(2)包含(containment)

$$A \subseteq B \Leftrightarrow f_A(x) \le f_B(x)$$

(3)相等(equality)

$$A = B \Leftrightarrow f_A(x) = f_B(x)$$

(4)聯集(union)

$$f_{A\cup B}(x)=\max\left(f_B(x),f_B(x)\right)$$

(5)交集(intersection)

$$f_{A\cap B}(x)=\min\left(f_B(x),f_B(x)\right)$$

## 2.2 AdaBoost

### 2.2.1 AdaBoost 演算法

AdaBoost 演算法是由 Freund 和 Shapire 在 1995 年所發表[2]，詳細演算法如 Algorithm 2-1 所示;此演算法將$(x_1,y_1)$, $(x_2,y_2)$,…, $(x_m,y_m)$，此 m 個 pair 當作演算法的輸入，$x_1,x_2,\cdots,x_m$，是已知 training sample，他們的 label 分別是 $y_1$ $y_2,\cdots,y_m$，而$y_i$屬於{+1, -1}，並且假設這 m 個點的權重一開始皆是$D_1(i)=1/m$。

**Algorithm 2-1** AdaBoost Algorithm

Given: $(x_1,y_1)$, $(x_2,y_2)$,…, $(x_m,y_m)$, where $x_1\epsilon X$ , $y_m\epsilon Y=\{+1,-1\}$

Initialize: $D_1=1/m$.

**for** t=1,…,T **do**

Train weak learner using distribution $D_t$

Get weak hypothesis $h_t: X\rightarrow\{+1,-1\}$ with error $\sum_{i:h_t(X_i)!=Y_i}D_t(i)$

Choose $\alpha_t=\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)/2$.

Update:

$$D_{t+1}(i)=\frac{D_t(i)}{Z_t}*\begin{cases}e^{-\alpha_t}, \text{if } Y_i=h_t(X_i)\\ e^{\alpha_t}, \text{if } Y_i\neq h_t(X_i)\end{cases}.=\frac{D_t(i)*\exp(-\alpha_t*Y_i*h_t(X_i))}{Z_t}$$

where $Z_t$ is a normalization factor

**end for**

Output the final hypothesis:

$H(X_i)=sign(\sum_{t=1}^{T}(HT_t(X_i)*\alpha_t)$

### 2.2.2 AdaBoost 演算法分析

首先已知 training sample 有 m 個點，$x_1, x_2, \cdots, x_m$，他們的 label 分別是 $y_1\ y_2, \cdots, y_m$，而$y_i$屬於{+1, -1}，並且假設這 m 個點的權重一開始皆是 $D_1(i) = 1/m$。並且預設有 n 個基本分類器。

接下來我們會跑 T 個回合的迴圈，每一個迴圈主要目的是調整此 m 個點權重，並挑選一個錯誤率最低的基本分類器。

以下為演算法的迴圈:

{

1. 計算每一基本分類器的錯誤率，錯誤率計算$\sum_{i:h_t(X_i)!=Y_i} \mathbf{D_t(i)}$，$h_t$:基本分類器，$D_t(i)$:第 t 回合第 i 點的錯誤率。

   Ex:在第一回合，要決定初始基本分類器，我們會從已經預設好的基本分類器集裡選擇一個最好一個基本分類器。選擇方法如下:

   利用每一個基本分類器分別去測試此 m 個點的分類結果，看預測出來的結果有無跟 $y_1\ y_2, \cdots, y_m$的 label 是一樣的，如果沒有，增加此點的權重給此分類器當作錯誤率，初始錯誤率計算$\sum_{i:h_t(X_i)!=Y_i} D_1(i)$，$h_t$:基本分類器。

   並找出錯誤率最低的基本分類器當作我們的初始基本分類器

2. 找出錯誤率最低的基本分類器當作第 t 回合的基本分類器$HT_t$。

3. 接下來要決定 $\alpha_t$ 的值，目的在於在下一個回合中使整體的錯誤率會最低。

   當 $\alpha_t = \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)/2$ 時，整體的 error 會最低。

4. 一開始此 m 個點的權重都是一樣的，在每一回合中，我們要提升這回合中分錯的點的權重，以及降低此回合被分對的點的權重，此舉的目的是讓下一回合所挑選的基本分類器能夠將此回合被分錯的點分對：

(1)for i=1,…,m do

$$Temp(i) = D_t(i) * \begin{cases} e^{-\alpha_t}, if\ Y_i = h_t(X_i) \\ e^{\alpha_t}, if\ Y_i \neq h_t(X_i) \end{cases}$$

end for

(2)$Z_t = \sum_{i=1}^{m} Temp(i)$

(3)for i=1,…,m do

$$D_{t+1}(i) = \frac{Temp(i)}{Z_t} * \begin{cases} e^{-\alpha_t}, if\ Y_i = h_t(X_i) \\ e^{\alpha_t}, if\ Y_i \neq h_t(X_i) \end{cases}$$

end for

當此回合所挑出的基本分類器錯誤率 > 50%，則跳出此迴圈。

}

1~5:說明跑T 個回合的迴圈，並在每回合中更新training sample $x_1, x_2, \cdots, x_m$的權重$D_t(i)$，在下一回合中，利用更新過的權重選擇一個錯誤率最小基本分類器，提升這回合分錯的點的權重，以及降低此回合被分對的點的權重，此舉的目的是讓下一回合所挑選的基本分類器能夠將此回合被分錯的點分對。此舉的目的是讓不同的基本分類器可以互相填補各自在分類方法上不足的地方，也就是結合多個基本分類器，讓這些基本分類器變成一個比較強大的基本分類器。

所以最終的強分類器就是由一堆基本分類器的線性組合：

$H(X_i)=sign(\sum_{t=1}^{T}(HT_t(X_i) * \alpha_t)$，H:代表強分類器.

## 2.3 AdaBoost.MH

### 2.3.1 AdaBoost.MH 演算法

AdaBoost.MH 可以針對多群數目以及多標籤數目的文章集進行分類，其詳細演算法如 Algorithm 2-2 所示。而 Schapire 與 Singer[4]在 2000 年所發表的 BoosTexter 系統為 AdaBoost.MH 的一個實作系統。

演算法中，$\mathcal{X}$ 定義為所有的 Training Data 文章，$\mathcal{Y}$ 為所有類別的集合，我們將 $\mathcal{Y}$ 集合的大小設定為 $k=|\mathcal{Y}|$。假設有一篇 training 的文章為 $(x,Y)$，$x\in\mathcal{X}, Y\subseteq\mathcal{Y}$，且 $l\in\mathcal{Y}$，可得知以下結果:

$$Y[l]=\begin{cases}+1 \ if \ l\in Y\\ -1 \ if \ l\notin Y\end{cases}$$

$S$ 為所有訓練的資料，包含了 $\langle(x_1,Y_1),\ldots,(x_m,Y_m)\rangle$，$x_i\in\mathcal{X}, Y_i\subseteq\mathcal{Y}$。每一 Round 所挑選出的 weak hypothesis $h_t:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}$，$h(x,l)$ 解釋為一個預測此文章 $x$ 是否有被分配到類別 $l$，而 $|h_t(x_i,l)|$ 可以解釋為此次預測的信心度。每一個 training 的項目 $(x,Y)$ 對應到 $k$ 個二元的 label，就像是 $((x,l),Y[l]),\ for\ all\ l\in\mathcal{Y}$，一樣。而演算法中的 $Z_t$ 為一個正規化的變數，如方程式(2.1)所示:

$$Z_t=\sum_{i=1}^{m}\sum_{l\in\mathcal{Y}}D_t(i,l)\exp\left(-\alpha_t Y_i[l]h_t(x_i,l)\right) \qquad (2.1)$$

每一 Round $t$ 輸出的 hypothesis 形式如下所示。$w$ 為一個 term，$w\in x_i$ 則代表 $w$ 出現在第 $i$ 篇文章中，$c_{jl}$ 為一個數值，代表 weak hypothesis $h_t$ 的輸出結果; $j$ 為 0 或是 1，各自代表了 $w$ 出現在文章中或是不出現在文章中的兩種情況。

$$h_t(x_i,l)=\begin{cases} C_{0l} \text{ if } w\notin x \\ C_{1l} \text{ if } w\in x \end{cases}$$

**Algorithm 2-2** AdaBoost.MH Algorithm

Given: $(x_1,Y_1),...,(x_m,Y_m)$ where $x_i\in\mathcal{X}, Y_i\subseteq\mathcal{Y}$

Initialized: $D_1(i,l)=1/(mk), i=1,...,m$

**for** $t=1,...,T$ **do**

Pass distribution $D_t$ to weak learner.

Get weak hypothesis $h_t:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}$.

Choose $\alpha\in\mathbb{R}$

Update:

$$D_{t+1}(i,l)=\frac{D_t(i,l)\exp(-\alpha_t Y_i[l]h_t(x_i,l))}{Z_t}$$

where $Z_t$ is a normalized factor

**end for**

Output the final hypothesis:

$$f(x,l)=\sum_{t=1}^{T}\alpha_t h_t(x,l)$$

### 2.3.2 Weak hypothesis

每一篇文章 $x_i$ 在類別 $l$ 的權重一開始皆是 $D_1(i,l)=1/(mk)$。而 $X_j$, j ∈ {0,1}，$X_0=\{x:w\notin x\}$ 為代表 term $w$ 沒有出現在文章中，$X_1=\{x:w\in x\}$ 為有出現。

先給予當下第 $t$ Round 的 $D_t$，以及 term $w$，使得 $X_j, j\in\{0,1\}$，且對每一個類別 $l$ 作相對應的計算。$W_+^{jl}$（$W_-^{jl}$）可以稱為根據 $D_t$ 在文章發生 $X_j$ 的情況下且文章是(不是) $l$ 這個類別所計算出來的權重。方程式(2.2)為計算 $W_b^{jl}$ 之方法:

$$W_b^{jl}=\sum_{i=1}^{m}D_t(i,l)\left[\!\left[x_i\in X_j\wedge Y_i[l]=b\right]\!\right] \tag{2.2}$$

而 $Z_t$ 可以用 $W_b^j$ 來表示，如方程式(2.3)。

$$\begin{aligned}Z_t&=\sum_j\sum_{i:x_i\in X_j}D(i)\exp\left(-Y_ic_j\right)\\&=\sum_j\left(W_+^je^{-c_j}+W_-^je^{c_j}\right)\end{aligned} \tag{2.3}$$

接著再對 $Z_t$ 對作微分，可以得出 $c_{jl}$。$c_{jl}$ 可以被解釋為這個 feature 對於類別 $l$ 有多少的影響，$c_{jl}$ 為一個向量，其大小為總類別數目。$c_{jl}$ 的公式如方程式(2.4)所示。

$$c_{jl}=\frac{1}{2}\ln\left(\frac{W_+^{jl}}{W_-^{jl}}\right) \tag{2.4}$$

如果 $W_+^{jl}$ 或是 $W_-^{jl}$ 很小甚至趨近於 0，從方程式(2.4)公式中則會得到很大的 $c_{jl}$，這樣將會發生許多問題。為了避免這樣的問題發生，設定了 $\varepsilon=1/mk$ 加入到公式中，原先計算 $c_{jl}$ 的公式變成如方程式(2.5)所示:

$$c_{jl}=\frac{1}{2}\ln\left(\frac{W_+^{jl}+\varepsilon}{W_-^{jl}+\varepsilon}\right) \tag{2.5}$$

每一 Round 所選出的 weak hypothesis 需要是計算出來最小的$Z_t$。而將方程式(2.4)代入到方程式(2.3)可得出方程式(2.6)。

$$Z_t = 2\sum_{j\in\{0,1\}}\sum_{l\in\mathcal{Y}}\sqrt{W_+^{jl}W_-^{jl}} \tag{2.6}$$

每一 Round 我們所要選取的 term $w$就是$Z_t$為最小的$w$，而每一 Round 將會挑選最小的$Z_t$來當作 weak hypothesis。假如一個 feature 對於每一個類別能夠提供較大的區別性，這代表了系統可以根據這個 feature 的出現或是不出現在文章中辨別這篇文章的類別。舉例來說，若是一個 feature 出現在正向的類別文章中很多次，而在負向的類別文章出現次數很少，那麼計算出來的$c_{jl}$在預測正向類別的值將會很高，反之在預測負向類別的值將會很低，若是總類別為 2 時，所計算出來的$c_{jl}$在兩個類別的值將會呈現對稱，例如 0.8 與-0.8。當$W_+^{jl}$與$W_-^{jl}$差異最大時，將會得到最小化的$Z_t$。

## 2.4 SVM(Support Vector Machine)

文章的分類問題中，越來越多研究者使用machine-learning的技術來解決這方面的議題。Joachims[9]使用了Support Vector Machines(SVM)[10]來對文章進行分類。SVM為目前表現較好的一種分類演算法，其概念為事先給予一群分類好的資料集，如圖2-2所示，利用這些已知的資料訓練產生預測模型。爾後，若有尚未分類的資料時，都可以直接使用該模型預測該資料的結果。簡而言之，我們可以把模型想像成是一個黑箱，當任意資料通過模型後都會被對應至符合條件的區域，且作出分類結果，如圖2-3所示。

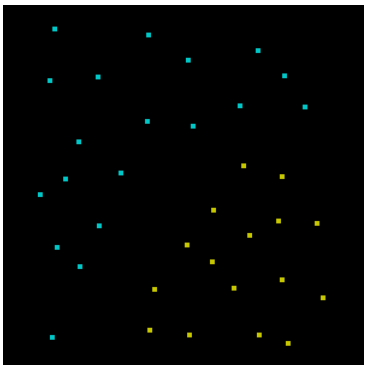

圖2-2. 原始資料集的分布圖

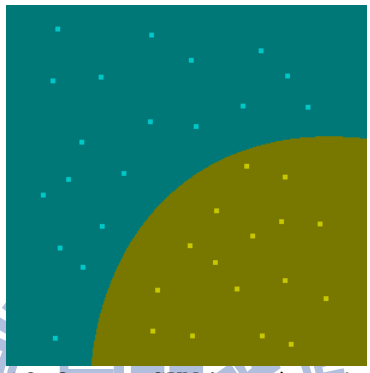

圖2-3. 經過SVM分類後的結果

然而，有時候原始空間中的資料分布，並非線性可分割（Non-linearly Separable）。因此，我們必須將資料映射至較高的維度中，才有機會以超平面將資料分割開來，如下圖 2-4 所示，在原本二維空間中無法線性分割的資料，再將資料點轉換至更高維度的三維空間後，可找到一超平面將資料點線性分割。

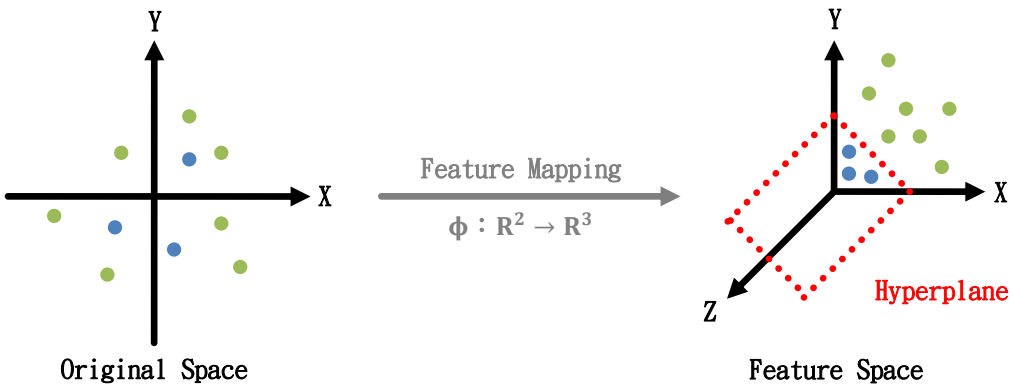


圖2-4. 將原始資料轉換至高維度中進行切割

為達成上述的目的，我們將利用訓練資料來尋找空間中的超平面，透過該超平面將資料順利的切開，如圖 2-5 所示的實線，並且期望該平面將兩側類別的距離分開的越遠越好，讓該超平面可以達到最一般化的效果（Generalization），否則容易使預測結果偏向某一類別，而過於迎合（Overfitting）訓練資料，造成未來使用該模型預測測試資料時，分類的結果不盡理想。下列為支援向量機的各項基本定義。

訓練資料集：$D = \{(X_i, Y_i) \mid X_i \in R^d, Y_i \in \{+1, -1\}\}$，$i = 1, \cdots, n$

$X_i$：第 i 個資料的特徵屬性，表示為 d 維度的向量。

$Y_i$：第 i 個資料的類別，於此表示為兩種類別的其中一種，+1 或−1。

分隔的超平面表示式：$w \bullet x - b = 0$

w：代表為平面的法向量（Normal Vector），$w \in R^d$。

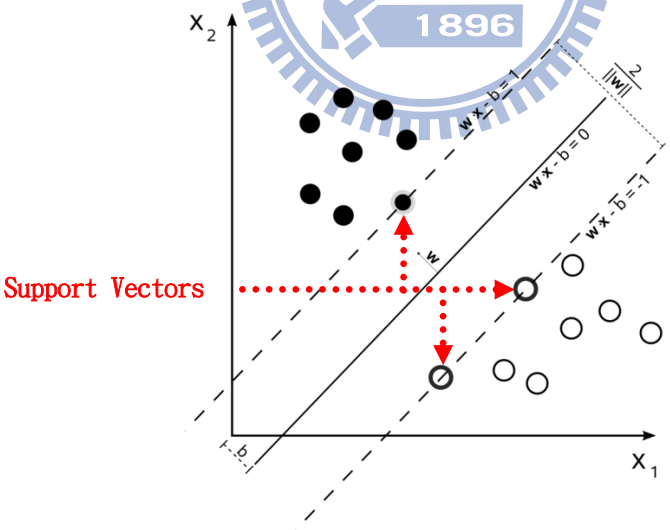


圖2-5. 超平面示意圖

如圖 2-5 所示，假設 P：$w \bullet x - b = 0$ 為一可將兩種類別資料分隔之超平面，藉由適當的重新調整（Rescaling），我們可以定義兩個平行於 P 的輔助超平面，並且這兩個輔助超平面會分別通過兩種類別距離 P 最近的所有資料點

（Support Vectors），圖 2-5 中所示之虛線，其定義如下：

$$w \bullet x - b = +1 \tag{2.7}$$

$$w \bullet x - b = -1 \tag{2.8}$$

考量到分類器的一般化情況，支援向量機的目標為使得兩個輔助超平面的距離越大越好，利用幾何學的原理，發現兩個輔助的超平面，方程式(2.7)和方程式(2.8)，中間的距離為 $\frac{2}{\|w\|}$。因此，欲讓兩者間距有最大值，必須使得$\|w\|$的數值越小越好。

目前支援向量機的現成工具方面，LIBSVM[23]為目前最熱門及方便的支援向量機工具軟體之一，本研究亦採用此軟體來做為實驗數據的比較方法之一。

## 2.5 Naïve Bayes

Naïve Bayes(NB)[11]分類法，是基於Bayes定理獨立性假設的一個簡易機率分類法。在以下公式中，$c$ 為類別變數，$T_1,...,T_n$ 為features變數，這些變數的交集機率，就如方程式(2.9)所示:

$$\begin{aligned}\Pr(c,T_1,...,T_n) &= \Pr(c)P(T_1,...,T_n \mid c) \\ &= \Pr(c)\Pr(T_1 \mid c)\Pr(T_2 \mid c,T_1)...P(T_n \mid c,T_1,...,T_{n-1}) \\ &= \Pr(c)\Pr(T_1 \mid c)\Pr(T_2 \mid c)...P(T_n \mid c) \\ &= \Pr(c)\prod_{i=1}^{n}\Pr(T_i \mid c) \end{aligned} \quad (2.9)$$

文章的分類當中，每一篇文章可以被表示成一個term的向量或是feature的向量。基於NB的假設:當給定文章類別資訊，文章內的特徵會互相獨立。每一篇文章 $d$ 被分到類別 $c$ 的機率的公式如方程式(2.10)所示:

$$\Pr(c \mid d) \propto \Pr(c,d) = \Pr(c,T_1,...,T_n) = \Pr(c)\prod_{i=1}^{n}\Pr(T_i \mid c) \quad (2.10)$$

如方程式(2.11)所示，當我們給定一個類別集 $C$，和一篇文章 $d$，文章 $d$ 將會被分類到最大機率的類別 $c$ 中，其中 $\Pr(c)$ 為一篇文章發生在類別 $c$ 的先前機率，而 $\Pr(T_i \mid c)$ 為term $T_i$ 發生在類別 $c$ 的文章中的機率。

$$\begin{aligned} c^* &= \arg\max_{c \in C} \Pr(c \mid d) \\ &= \arg\max_{c \in C} \Pr(c)\prod_{i=1}^{n}\Pr(T_i \mid c) \end{aligned} \quad (2.11)$$

## 2.6 Fuzzy Rule Methods Comparison

建立 fuzzy rule 的方法有很多種，一般來說模糊法則是依據專家經驗或經由訓練樣本所建立，每一個法則是以 If-Then 的形式來表達條件敘述語句。If 可以說是前提部分，是用來提供這個條件法則語句是否成立，Then 的部分則為如果條件法則語句成立所執行的結果。模糊法則的數量可依據訓練樣本數來增加或是減少，法則的數量越多，則越有更精確的結果。

AdaBoost[2]的方法建立 fuzzy rule，是一種以 Boosting 為基礎的方式產生出 fuzzy rule，就像之前所提到的 Del Jesus 等人在[16]所發表的方法。雖然這種方式可以產生出各種形式的 fuzzy rule，精確度相較之下也有較高的效果，但是僅限制在當 input 數量較少或是種類較小的情況之下，當我們的 input 數量大或著是種類多的時候，這種 Boosting Fuzzy Rule 的方法就會耗費相當多時間，進一步導致無法運算的可能性也提高，以下將會有更詳細的說明。

以下我們假設一個例子，在此小節的例子中，$x_i$ 為此 fuzzy rule 的第 $i$ 個 input，$R_i^j$ 為第 $i$ 個 input 的所有有可能發生的情況，$j=1...N$ 代表此 input $i$ 有可能的情況有 $N$ 種，$i=1...n$ 代表此 fuzzy rule 一共有 $n$ 個 input，若都符合條件的需求，則結果 $\mathcal{Y}$ 為 0。

$$if\ x_1\ is\ R_1^j\ and\ ...\ and\ x_n\ is\ R_n^j\ then\ \mathcal{Y}=0$$

若此時我們假設 $N$ 以及 $n$ 都持續的增加。因為 Boosting Fuzzy Rule 的方法是將所有種類的 fuzzy rule 全部都計算出來其結果，可以如圖 2-6 的結果知道，時間複雜度將會相當高，若是處理大量文章集的情況，可能會耗時相當大的時間，而在本系統中，假如 training 的 Round 數目為 500，只管 term 出現或是不出現在文章中兩種情況，代表說我們的 $n=500, N=2$ 則我們所需要運算的 fuzzy rule 種類為 $2^{500}$ 種，也代表說在此種情況下一共有 $2^{500}$ 條的 fuzzy rules，因為數量太過於龐大，所以在該情況下完全無法做運算，所以我們可以說，Boosting Fuzzy Rule 的方法可行，但是必須要在文章集的量較小或是 input 數目較少以及每一

個 input 的可能性不多的情況，若是要在大量的文章集情況，則不可行，代表無法在各種變化多的形式下作處理，也無法適應多種不同的文章集。

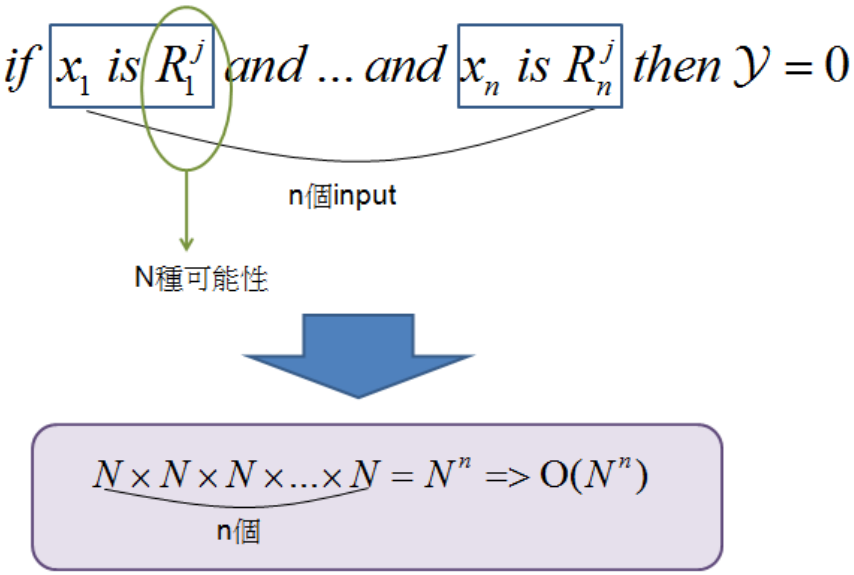


圖2-6. Boosting fuzzy rule時間複雜度計算

本系統中所使用到的 fuzzy rule 是屬於可使用在多 input 的情況下，每一條 fuzzy rule 只有一個 input，也就是 term，每一個 input 也只有兩種情況，包含了出現以及不出現，代表說每一條 fuzzy rule $n=1, N=2$，而 fuzzy rule 的數目由 training 的 Round 數目來決定。所以本系統所有 fuzzy rule 需要計算的次數為 $Training's\ Rounds \times 2$，其時間複雜度為 $O(Training's\ Rounds)$。以下舉兩個例子為本系統所使用的 fuzzy rule，可行性較高也可用在各種多變化的文章集和情況。

$if\ "bad"\ is\ "present\ in\ document"\ then\ return\ "c_{1l}"$

$else\ return\ "c_{0l}"$

$if\ "wast"\ is\ "present\ in\ document"\ then\ return\ "c_{1l}"$

$else\ return\ "c_{0l}"$

# 第三章、系統設計

## 3.1 概念

AdaBoost[2]為一種群體學習演算法，它允許其它的分類法當成其中的一個弱分類器;例如決策樹、簡單的 decision rule、甚至於 SVM[9][10]。AdaBoost 扮演的角色是一個框架的角色，將這些弱分類器列為它框架的一部份，AdaBoost 可藉由調整訓練資料的權重，每一回合挑出最佳的弱分類法，最後找出一組弱分類法集合和此集合裡每個弱分類法的權重，最後再予以合併，形成一強分類器。Schapire 與 Singer 所發表的 AdaBoost.MH 是將 AdaBoost 做延伸，將原先 weak hypothesis 的 output 從+1,-1 延伸到實數 $\mathbb{R}$[8]，並且加入了 Multi-Class、Multi-Label 的處理能力，是一種相當實用的分類法。

模糊邏輯的技術在不同的領域都可以看到它的蹤跡，自從 Zadeh 教授的 "Fuzzy Sets"論文在 1965 年發表後[13]，就帶動了這方面的研究風氣。由於現實世界的許多問題都充滿著不確定性及不可預料，如果想要靠傳統的二元邏輯的思考模式來解決所有的問題，是幾乎不可能的。本研究結合 Fuzzy 與 AdaBoost.MH，提出一個分類的方法;本研究也針對傳統的模糊化技術在訓練模糊規則需時過長的問題做了改善。

本論文使用 AdaBoost.MH[8] framework 設計出模糊化分類系統，而 AdaBoost.MH 能夠在每一回合挑選出最好的 weak hypothesis，且每一個 weak hypothesis 將會對應到一個 if-then rule，根據 if-then rule 所判斷出的結果，取出相對應的 weak hypothesis 數值，本研究提出使用 fuzzy number 代表分類的結果，詳細的流程將會在後面章節有更多的說明。Del Jesus 等人[16]提出一個結合 fuzzy 與 AdaBoost 之方法，他們將每一個 fuzzy rule 視為一個 weak hypothesis，且 fuzzy rule base 能被解釋為 weak hypothesis 與權重的結合。基本上，他們的分析結果符合 weak hypothesis 的設計原則，即 weak hypothesis 可以是任何類型的分類器。

本論文也將本系統的運作原理發展出 Semi-Supervised 的形式，稱為 Semi-Boosting，可用少量的 labeled 資料來進行文件分類，實驗結果有不錯之成效，詳細的流程在後續章節中會詳細介紹。此新方法為本系統的延伸想法，其演算法以及準確度在未來可望改善以及提升。除此之外在時間方面在未來也有改善的空間和可能性。

## 3.2 系統架構

如圖 3-1 所示。本系統分成幾個步驟來完成:首先，在 Fuzzy rule base 架構中，rule selector 會在每一回合從 rule pool 中選擇最好的 fuzzy rule 出來，其目標是為了減少訓練的 error。而選擇的標準將會在之後的章節會有詳細的介紹。當 rule base 準備好之後，系統將會從訓練資料中計算出每一個 rule 的訓練錯誤次數。之後將使用 logistic function 將這些錯誤轉換成 accuracy ratio value，其範圍在 0~1 之間。此外，本論文提出一個以面積為基礎的表示法，藉此來表示每一個 rule 的信心度(fuzzy confidence value)。最後，本系統的分類結果可以透過一個程度轉換(degree transformation)方法，得出文件歸屬到各類別的程度。

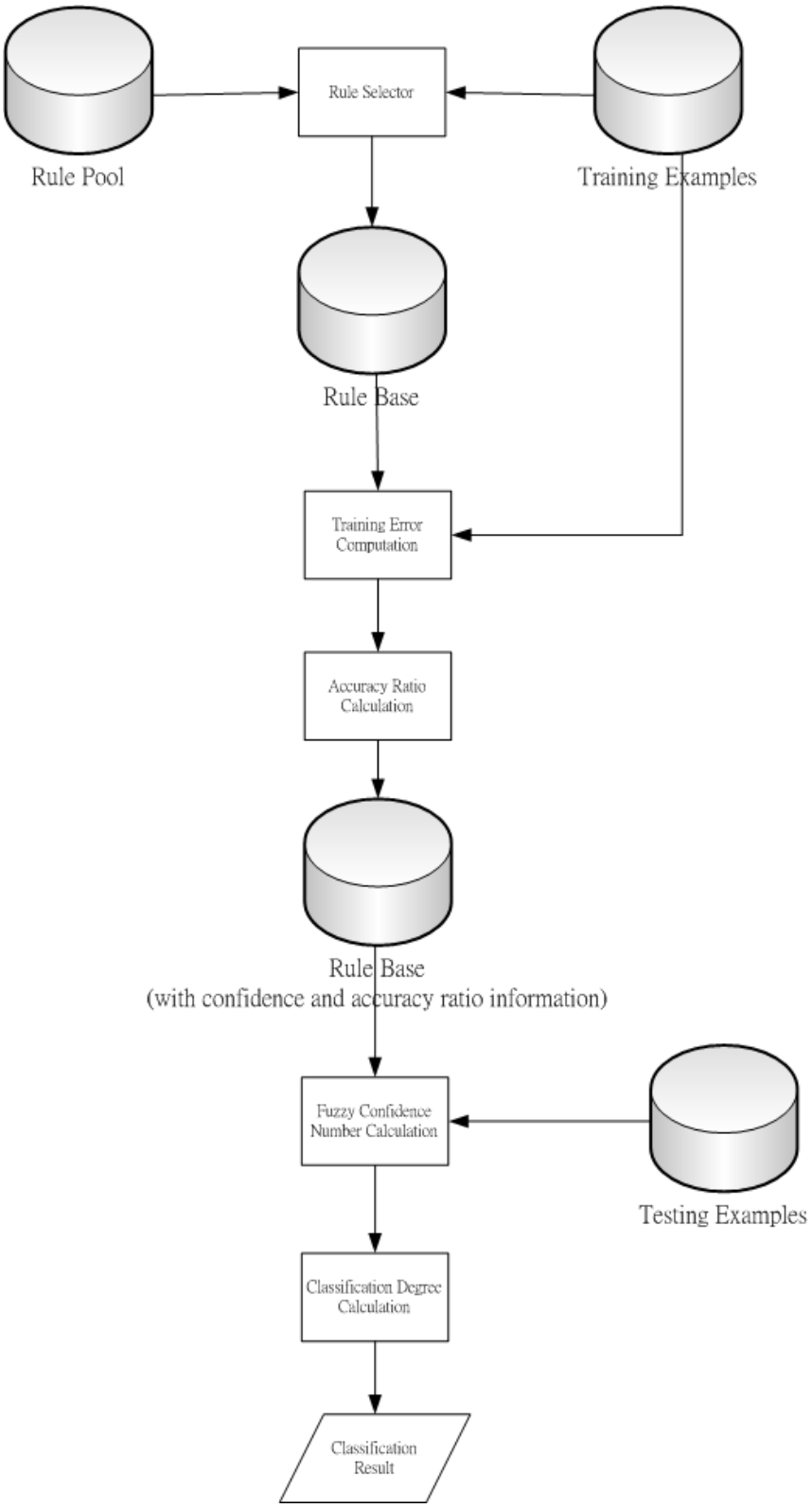


圖3-1.Fuzzy Classification系統流程架構圖

## 3.3 系統演算法

本論文是延伸自 AdaBoost.MH，將每一個 weak hypothesis 以 fuzzy rule 表示，最後結果則是以 fuzzy number 來計算。Algorithm 3-1 為本系統之整體演算法，包含了 AdaBoost.MH 以及本系統之主要演算法，而本系統主要的演算法將在這節會有詳細的介紹。

---

**Algorithm 3-1** Fuzzy AdaBoost.MH Algorithm

---

Given: $(x_1,Y_1),...,(x_m,Y_m)$ where $x_i \in \mathcal{X}, Y_i \subseteq \mathcal{Y}$

Initialized: $D_1(i,l)=1/(mk), i=1,...,m$

**for** $t=1,...,T$ **do**

Find the weak hypothesis $h_t : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that minimizes the error with respect to the Distribution $D_t$

Choose $\alpha \in \mathbb{R}$

Update:

$$D_{t+1}(i,l)=\frac{D_t(i,l)\exp\left(-\alpha_t Y_i[l]h_t(x_i,l)\right)}{Z_t}$$, where $Z_t$ is a normalized factor

**end for**

Given: $h_1(x_i,l),...,h_T(x_i,l)$ and $Y_i$ where $i=1,...,m$ and $Y_i \subseteq \mathcal{Y}$, $\mathcal{Y}$

**for** $t=1,...,T$ **do**

$$Error_t=\sum_{i=1}^{m}\left[\left[\mathcal{Y}\left[\arg\max_{l\in\mathcal{Y}}\left[h_t(x_i,l)\right]\right]\notin Y_i\right]\right]$$

**end for**

Initialize: $Max=max[E_1,...,E_T]$, $Min=\min[E_1,...,E_T]$, $Avg=\frac{1}{T}\sum_{t=1}^{T}E_t$

$a=\text{rounding off}\left\{1+\left[\left(Max/Min\right)-c\right]\times 10\right\}$, b

---

for $t = 1, \ldots, T$ do

$$\tau = \frac{Avg - E_t}{(Max - Min)/10}, \quad A_t = \frac{1}{1 + a \times e^{-b\tau}}$$

end for

for $i = 1, \ldots, z$ do

$$S(x_i, l) = \frac{1}{2} \sum_{t=1}^{T} h_t(x_i, l)[1 - (1 - A_t)^2]$$

end for

Initialize: $S_{\max} = Max[S(x_1, l), \ldots, S(x_z, l)]$, $S_{\min} = \min[S(x_1, l), \ldots, S(x_z, l)]$

if $(S_{\min} < 0)$ then

for $i = 1, \ldots, z$ do

$$S(x_i, l) = S(x_i, l) + (-1) \times S_{\min} + 1$$

end for

end if

$$S_{average} = \frac{1}{Z \times k} \sum_{i=1}^{z} S(x_i, l)$$

for $i = 1, \ldots, z$ do

$$Degree(x_i, l) = \frac{1}{1 + e^{-\tau}}, \quad \tau = \frac{S(x_i, l) - S_{average}}{(S_{\max} - S_{\min})/10}$$

end for

Output: $\mathcal{Y}\left[\arg\max\{Degree(x_i, l)\}\right]$ where $i = 1, \ldots, z$

接著將介紹本系統的四大主要演算法，分別為(1)Error Count Calculation (2)Accuracy Ratio Calculation (3)Fuzzy Confidence Calculation (4)Degree Transformation，以及本系統的延伸 Semi-Boosting。傳統的 machine learning 中，每一篇文章 $x \in \mathcal{X}$ 將會被分類到單一的類別 $y \in \mathcal{Y}$ 中。而在 multi-label 中，每一篇文章 $x \in \mathcal{X}$ 將有可能被分類到多個類別中 $Y \subseteq \mathcal{Y}$。

本系統的演算法中，定義 $\mathcal{X}$ 為所有 Training Data 的文章集合，$\mathcal{Y}$ 為所有的 label 類別，其類別數量為 $|\mathcal{Y}| = k$，$x_i$ 為第 $i$ 篇 Training Data，$i = 1,...,m$ 代表 Training Data 文章數目為 $m$，$T$ 為 training 的 Round 數目，$Y_i$ 為每一篇文章 $x_i \in \mathcal{X}$ 所被標記到的類別集合且 $Y_i \subseteq \mathcal{Y}$；本系統可對 multi-label 的 dataset 進行處理，故集合中的類別數量未必為 1，而 Testing Data 的數目為 $z$。而 $w$ 為一個 term，$w \in x_i$ 則代表 $w$ 出現在第 $i$ 篇文章中，$h_t(x_i, l)$ 為第 $t$ Round 所挑選出來的 weak hypothesis，其 $h_t$ 的值為根據 input document $x_i$ 所計算出的數值，$l$ 為 label 類別，而 weak hypothesis 的值可以針對 $w$ 是否出現於文章中來決定其數值。

$$h_t(x_i, l) = \begin{cases} C_{0l} \; if \; w \notin x \\ C_{1l} \; if \; w \in x \end{cases}$$

$c_{jl}$ 為一個數值，$j$ 為 0 或是 1，各自代表了 $w$ 出現在文章中或是不出現在文章中的兩種情況。Algorithm 3-2 中的 output 為 $E_t$ (where $t = 1,...,T$)，代表第 $t$ Round 的 Hypothesis 的 Error Count，而 Algorithm 3-3 中的 output 為 $A_t$ ($t = 1,...,T$)則為第 $t$ Round 的 Hypothesis 的 Accuracy Ratio，$S(x_i, l)$ 為第 $i$ 篇文章在類別 $l$ 所計算出來的面積值，$Degree(x_i, l)$ 是第 $i$ 篇文章在類別 $l$ 的 degree。關於 AdaBoost.MH 的詳細定義可以參考本論文第二章的相關研究的部分，有更詳細的解說以及介紹。

### 3.3.1 Error Count Calculation

本小節所介紹的演算法中，方程式(3.1)為第$t$ Round 的 weak hypothesis 在第$i$篇文章中，回傳最大值類別的 index，並且根據回傳類別的 index，到所有類別集合$\mathcal{Y}$中取出相對應的 label 名稱，並且檢查第$i$篇 training 文章是否有被分類到此類別，如果不符合，代表此規則為成立，回傳 1，則此 hypothesis 的 Error Count 加 1，其中$[\![\cdot]\!]$為一個 indicator function。最後 output 的是每一個 hypothesis 的 Error Count。

$$Error_t = \sum_{i=1}^{m} [\![ \mathcal{Y}[\arg\max[h_t(x_i, l)]] \notin Y_i ]\!] \tag{3.1}$$

Ex:

$\mathcal{Y} = \{pos, neg\}, Y_i = \{pos\}$

假設此例中 $w \notin x_i$ 所以回傳 $c_{0l} = (-0.1, 0.1)$，故 $\arg\max[h_t(x_i, l)]$ 的回傳值為 2，代表 $\mathcal{Y}[\arg\max[h_t(x_i, l)]]$ 所取出來的 label 為 $neg$，因為 $neg \notin Y_i$，故條件成立，此 $Error_t$ 加 1。

---

**Algorithm 3-2** Error Count Calculation Algorithm

---

Given: $h_1(x_i, l), ..., h_T(x_i, l)$ and $Y_i$ where $i = 1, ..., m$ and $Y_i \subseteq \mathcal{Y}$, $\mathcal{Y}$

**for** $t = 1, ..., T$ **do**

$$Error_t = \sum_{i=1}^{m} [\![ \mathcal{Y}[\arg\max[h_t(x_i, l)]] \notin Y_i ]\!]$$

**end for**

Output: $E_1, ..., E_T$

---

### 3.3.2 Accuracy Ratio Calculation

本小節所介紹的演算法，是根據方程式(3.1)的結果$E_1,...,E_T$來做後續的運算；*Max*和*Min*分別為所有 hypothesis 中最大及最小 Error Count，*Avg*為所有 Hypotheses 的 Error Count 之平均，$a,b,c$為三個參數。此演算法可將原本每一個 Hypothesis 的 Error Count 轉變為 Accuracy Ratio。在方程式(3.2)中的分母$(Max-Min)/10$為代表算出最大與最小的距離，除以 10 個間隔，以計算出每一個間隔的大小，在本系統是將設定值的範圍定於+5~-5 之間，代表分成 10 等份。如圖 3-2 可以很清楚了解範圍大致上是在+6~-6 之間。分子$Avg-E_t$代表此 Hypothesis 與平均值的差距，若$Avg-E_t$越大代表此 Hypothesis 的錯誤次數越小，則$A_t$越大，代表 Accuracy Ratio 越大。最後將方程式(3.2)代入到方程式(3.3)中，就可以計算出每一個 hypothesis 的 Accuracy Ratio。

$$\tau = \frac{Avg - E_t}{(Max - Min)/10} \tag{3.2}$$

$$A_t = \frac{1}{1 + a \times e^{-b\tau}} \tag{3.3}$$

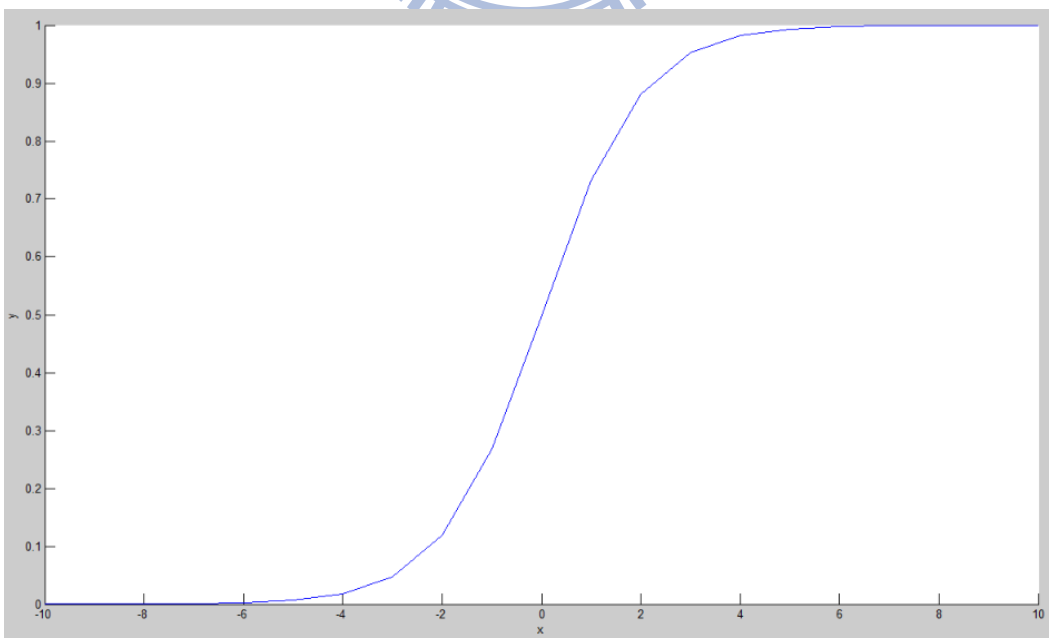


圖3-2. Logistic Function演算法圖示

**Algorithm 3-3** Accuracy Ratio Calculation Algorithm

Given: $E_1,...,E_T$

Initialize: $Max = max[E_1,...,E_T]$, $Min = \min[E_1,...,E_T]$, $Avg = \frac{1}{T}\sum_{t=1}^{T} E_t$

$a = \text{rounding off}\left\{1+\left[\left(Max/Min\right)-c\right]\times 10\right\}$, b

**for** $t = 1,...,T$ **do**

$$\tau = \frac{Avg - E_t}{\left(Max - Min\right)/10} \quad , \quad A_t = \frac{1}{1+a\times e^{-b\tau}}$$

**end for**

Output: $A_1,...,A_T$

### 3.3.3 Fuzzy Confidence Calculation

本小節所介紹的演算法為計算每一篇 Testing Data 文章在每一個類別 $l$ 的面積值，方程式(3.4)中所計算的 $S(x_i,l)$ 為第 $i$ 篇文章在類別 $l$ 的面積值，這代表了假設 $|\mathcal{Y}|=k$，則代表一共有 $k$ 的類別，而每一篇文章也將會算出 $k$ 個面積值。方程式(3.4)為將每一 Round $t$ 的 Hypothesis 的 confidence 當作三角形面積的底寬度，而 Hypothesis 的 Accuracy Ratio 是由方程式(3.3)所計算出來，用來當作三角形切線的高度，其經由 Accuracy Ratio 所切出來的面積為一個梯形，而此梯形面積為此演算法所要計算之值。

$$S(x_i,l) = \frac{1}{2}\sum_{t=1}^{T} h_t(x_i,l)[1-(1-A_t)^2] \qquad (3.4)$$

Ex:

假設我們目前想要計算的面積為等號左邊的藍色梯形面積(圖 3-3)，以 $|h_t(x_i,l)|$ 當作三角形面積的底，但是由於我們需要知道此面積算出來為正還是負，所以在本演算法中並沒有加上絕對值，保持原先的正負號。故我們可以先計算出等號右邊藍色三角形的面積 $\frac{1}{2}h_t(x_i,l)$。

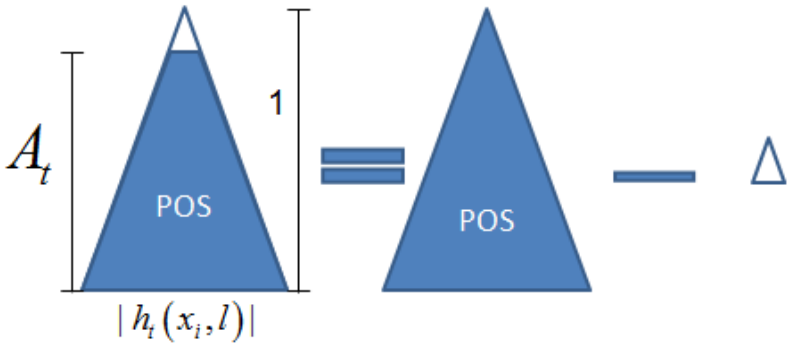


圖3-3. 梯形面積生成

接著算出白色小三角的面積，由圖 3-4 可知其步驟。最後將兩三角形相減，就可得到最後藍色梯形面積值。如圖 3-5 為將圖 3-4 算出來的白色小三角形，代入到圖 3-3 中，就可得到其結果。

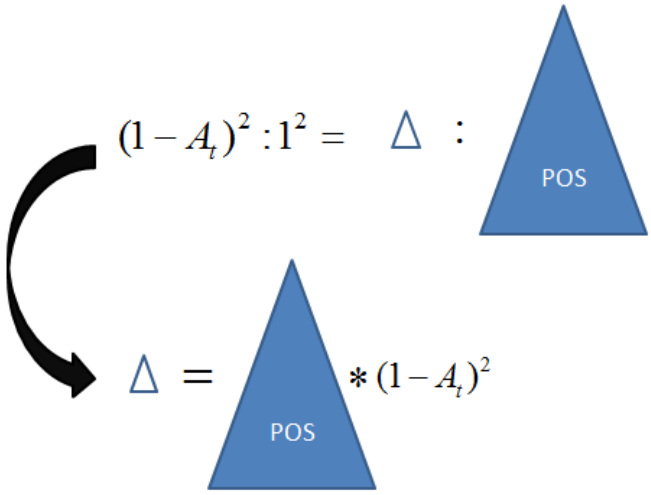


圖3-4. 面積公式推導(1)

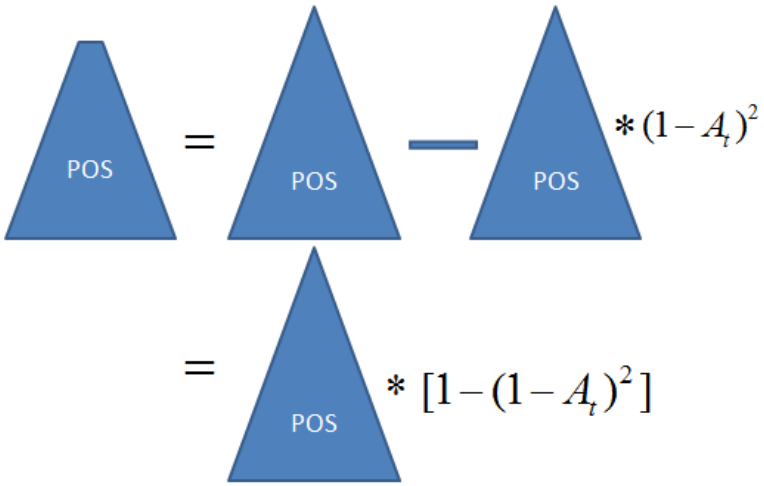


圖3-5. 面積公式推導(2)

經由方程式(3.4)計算出每一篇 Testing Data 在所有類別 $l$ 中各自的面積值之後，將其 output 給下一個 Algorithm 使用。

---

**Algorithm 3-4** Fuzzy Confidence Calculation Algorithm

---

Given: $A_1,...,A_T$ , $h_1(x_i,l),...,h_T(x_i,l)$ where $i=1,...,z$

for $i=1,...,z$ do

$$S(x_i,l)=\frac{1}{2}\sum_{t=1}^{T}h_t(x_i,l)[1-(1-A_t)^2]$$

end for

Output: $S(x_i,l)$ where $i=1,...,z$

---

### 3.3.4 Degree Transformation

本小節的演算法是對所有 Testing Data 在所有類別 $l$ 的面積值，進行正規化處理，形成每一篇 Testing Data 皆會有一個在各類別 $l$ 的 Degree。首先在方程式(3.4)中計算出的所有 $S(x_i,l)$ where $i=1,...,z$ 中找出最小值 $S_{\min}$ 與最大值 $S_{\max}$ ，若是找出的最小值 $S_{\min}$ 小於 0，則執行方程式(3.5)對所有的 $S(x_i,l)$ 加上所找出的最小值 $S_{\min}$ 乘上-1 之後再加上 1，加 1 是使用 smoothing 技巧以避免有 0 的情況發生，之後得到出新的 $S(x_i,l)$ 。

$$S(x_i,l)=S(x_i,l)+(-1)\times S_{\min}+1 \qquad (3.5)$$

接著方程式(3.6)為算出所有 $S(x_i,l)$ 的平均 $S_{average}$ 。

$$S_{average}=\frac{1}{Z\times k}\sum_{i=1}^{z}S(x_i,l) \qquad (3.6)$$

最後再經由方程式(3.7)代入到方程式(3.8)去求出每一篇 Testing Data 在各個類別中的 Degree，且範圍為 0~1，其演算法如 Algorithm 3-5 所示。其中

$\arg\max\{Degree(x_i,l)\}$ 代表先找出此篇 Testing 文章 $i$ 在各類別中最大 Degree 類別 $l$ 的 index，接著再到 $\mathcal{Y}$ 中找出類別的名稱，並且回傳。

$$\tau = \frac{S(x_i,l) - S_{average}}{(S_{\max} - S_{\min})/10} \tag{3.7}$$

$$Degree(x_i,l) = \frac{1}{1+e^{-\tau}} \tag{3.8}$$

**Algorithm 3-5** Degree Transformation Algorithm

Given: $S(x_i,l)$ where $i = 1,...,z$

Initialize: $S_{\max} = Max[S(x_1,l),...,S(x_z,l)]$, $S_{\min} = \min[S(x_1,l),...,S(x_z,l)]$

if $(S_{\min} < 0)$ then

for $i = 1,...,z$ do

$$S(x_i,l) = S(x_i,l) + (-1) \times S_{\min} + 1$$

end for

end if

$$S_{average} = \frac{1}{Z \times k} \sum_{i=1}^{z} S(x_i,l)$$

for $i = 1,...,z$ do

$$Degree(x_i,l) = \frac{1}{1+e^{-\tau}} \quad , \quad \tau = \frac{S(x_i,l) - S_{average}}{(S_{\max} - S_{\min})/10}$$

end for

Output: $\mathcal{Y}\left[\arg\max\{Degree(x_i,l)\}\right]$ where $i = 1,...,z$

### 3.3.5 Semi-Boosting

本小節為 Semi-Boosting 的演算法，在此我們設 $z$ 為 Testing Data 的文章數目，$Y_i$ 為第 $i$ 篇 Training Data 的類別。演算法中間的 if 條件式是為了要辨別此篇 Testing Data 所被分類到的類別其信心度是否大於等於門檻值 $C$，若條件成立，則將此 Testing Data 加入到 Training Set 中，並且從 Testing Set 中刪除此篇文章。而演算法最後的 if 條件式則是判斷剩餘的 Testing Set 中的文章數目是否大於所設定的門檻值，如果條件成立，則會再繼續進行下一回合的 Semi-Boosting 演算法，直到不符合條件式的條件才會停止。

---

**Algorithm 3-6** Semi-Boosting Algorithm

---

Given labeled example pairs $(x_1,Y_1),...,(x_m,Y_m)$, unlabeled examples $(x_1,...,x_Z)$, where $x_i \in \mathcal{X}, Y_i \subset \mathcal{Y}$, a confidence threshold value $C$ and a stopping threshold value $N$

Run AdaBoost.MH algorithm using labeled examples $(x_1,Y_1),...,(x_m,Y_m)$

for $i=1,...,z$ do

$x_i.data = x_i$

$x_i.label = \arg\max_{1\le l\le K} \sum_{t=1}^{T} h_t(x_i,l)$

$x_i.confidence = \max_{1\le l\le K} \sum_{t=1}^{T} h_t(x_i,l)$

end for

for $i=1,...,z$ do

if $x_i.confidence \ge C$ then

Remove $x_i.data$ from unlabeled set

Insert $(x_i.data, x_i.label)$ into labeled set

end if

---

end for

if number of unlabeled examples $> N$ then

Run Semi-Boosting with new labeled examples and unlabeled examples

else

Insert the rest of unlabeled examples with their labels into labeled set

end if

## 3.4 系統概念

系統流程解說:

以下為詳細的系統概念例子，分別介紹各 Step 的詳細步驟。

### Step 1: Hypothesis Value

假設我們由 Adaboost.MH 跑出來的 Hypothesis 如圖 3-6 右方，Round 數為 5，所以共有 5 個 Hypotheses。我們由第一個 Round 取出來的字 bad 來解釋，圖 3-6 左方表格的 POS、NEG 為要分類的兩個類別名稱，$c_0$、$c_1$ 分別代表 bad 這個字不出現以及出現在文章中的情況，而 0.344、-0.344、0.218、-0.218 為經由 AdaBoost.MH 所計算出來的數值。例如表中-0.344 為文章中出現 bad 且同時被分到 POS 這個類別得到的值，這也代表說 0.344 為文章中出現 bad 且同時被分到 NEG 這個類別得到的值。反之 0.218 為文章中未出現 bad 且同時被分到 POS 這個類別得到的值，由此類推。

| bad | POS | NEG |
|---|---|---|
| $c_0$ | 0.218 | -0.218 |
| $c_1$ | -0.344 | 0.344 |

bad
0.218 -0.218
-0.344 0.344
Run : 1

wast
0.082 -0.082
-0.804 0.804
Run : 2

worst
0.069 -0.069
-0.601 0.601
Run : 3

the#best
-0.109 0.109
0.334 -0.334
Run : 4

bore
0.076 -0.076
-0.504 0.504
Run : 5

圖3-6. AdaBoost.HM之Hypothesis分析

Step 2: Error Count Calculation

假設我們一篇為 POS 類別的 Training Data 在 5 個 Hypotheses 的預測後，出現情況如表 3-1，其中 $c_0$ 代表此 Hypothesis 不出現在這篇文章當中，$c_1$ 為此篇文章有出現，POS、NEG 欄位為取出相對應 $c_0$ 或是 $c_1$ 的值。由於此篇 Training Data 文章是屬於 POS 的類別，所以可以假設說不管 Hypothesis 的出現情況是 $c_0$ 還是 $c_1$，其 Hypothesis 的值在 POS 的類別中都會是最大的，因為這樣的結果代表此 Hypothesis 偏向 POS 這個類別。以 bad 為例，在 $c_0$ 的情況下，POS 的類別的值 0.218 是最高的，所以我們可以認為說 bad 不出現於文章中時，此文章應該是偏向 POS 的類別，而此假設也符合此篇 Training Data 的類別 POS，所以預測正確，Error Count 並未增加。由此類推，如同表 3-1 中 The#best，在 $c_0$ 的情況下，NEG 的類別中的值 0.109 是最高的，所以可以認為說 The#best 在文章中未出現時，此文章應該是偏向 NEG 的類別，但是此篇文章屬於 POS 類別，故此假設為錯誤，所以 The#best 的錯誤次數+1，代表實際情況與預測的結果不同。而每一個 Hypothesis 在一篇文章中最多只會增加一次 Error Count。

| Hypothesis | $c_0 / c_1$ | POS | NEG |
|---|---|---|---|
| Bad | $c_0$ | 0.218 | -0.218 |
| Wast | $c_0$ | 0.082 | -0.082 |
| Worst | $c_0$ | 0.069 | -0.069 |
| The#best | $c_0$ | -0.109 | 0.109 |
| Bore | $c_0$ | 0.076 | -0.076 |

表 3-1. 舉例的 Training Data 文章中，Hypotheses 出現的情況與其值

此篇文章所計算出來每一個 Hypothesis 的錯誤次數為:

Bad:0

Wast:0

Worst:0

The#best:1

Bore:0

圖 3-7. 是在 Pang 的影評文章中算出來的實際例子，我們也以這個例子作為後面的計算數值:

```
bad 591
wast 674
worst 687
the#best 699
bore 679
```

圖3-7. 計算錯誤次數之實際結果

**Step 3: Accuracy Ratio Calculation**

首先在所有的 Hypotheses 中取出 MAX 以及 MIN 的 Error Count，並且計算 Average 的 Error Count，接著我們將每一個 Hypothesis 的錯誤次數，使用 Logistic Function 調整成正確的程度。其計算流程之前處理如圖 3-8。以下為根據圖 3-7 所舉的例子。

將 Error Count 代入到公式中

EX: bad = 591

$$\tau = \frac{(Avg - 591)}{(Max - Min)/10}$$

$$P(t) = \frac{1}{1 + a * e^{-b\tau}}$$

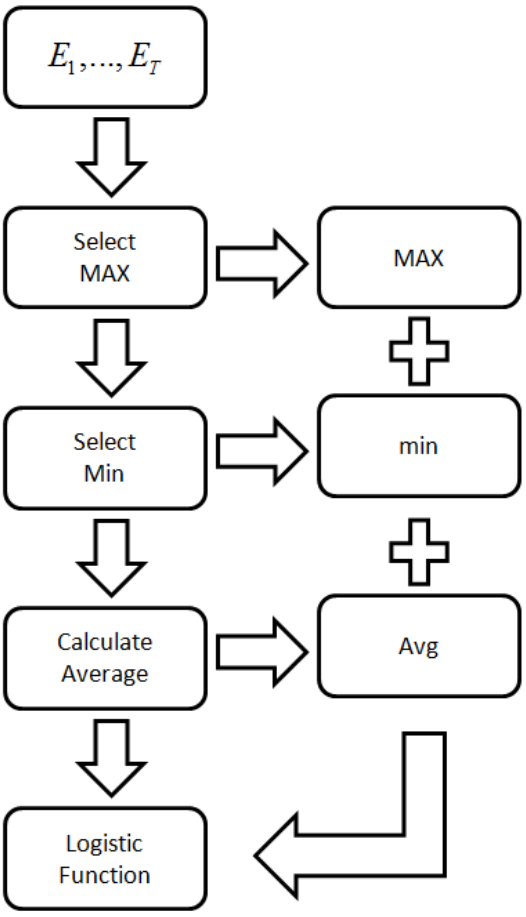


圖3-8.Logistic Function之前置處理

Step 4: Fuzzy Confidence Calculation

如表 3-2 為 5 個 Hypotheses 的 Error Count 經由 Accuracy Ratio Calculation 出來後的 Accuracy Ratio，以及在此篇 Testing Data 文章出現的情況與其相對應的值。

| Hypothesis | Accuracy | $c_0 / c_1$ | POS | NEG |
|---|---|---|---|---|
| Bad | 0.9 | $c_1$ | -0.344 | 0.344 |
| Wast | 0.7 | $c_0$ | 0.082 | -0.082 |
| Worst | 0.7 | $c_0$ | 0.069 | -0.069 |
| The#best | 0.5 | $c_0$ | -0.109 | 0.109 |
| Bore | 0.8 | $c_0$ | 0.076 | -0.076 |

表 3-2.舉例的 Testing Data 文章中，Hypotheses 出現的情況與其值

接著需要計算 POS 類別的面積值，如圖 3-9，以 bad 為例，以-0.344 為三角形面積底部的中心座標，因為其絕對值為此 Hypothesis 在 POS 這個類別的信心度，所以用絕對值來當作三角型底部的寬度，而三角形外框的高設為 1，但由於 bad 在 Training Data 中所算出來的正確率值(Accuracy Ratio)為 0.9，代表除了信心度之外，實際上的正確程度為 0.9，並沒有百分之百的正確，因此將 0.9 設為高度，代表經由 Training Data 測試之後影響了信心度的程度。最後計算高度為 0.9 的梯形面積，由於 bad 在 POS 類別值為負，故計算出來的面積我們給他標記為負號。使用面積法來當作 fuzzy number 符合 fuzzy number 的定義，包含了(1)convex (2)normal fuzzy set 兩種的特性。

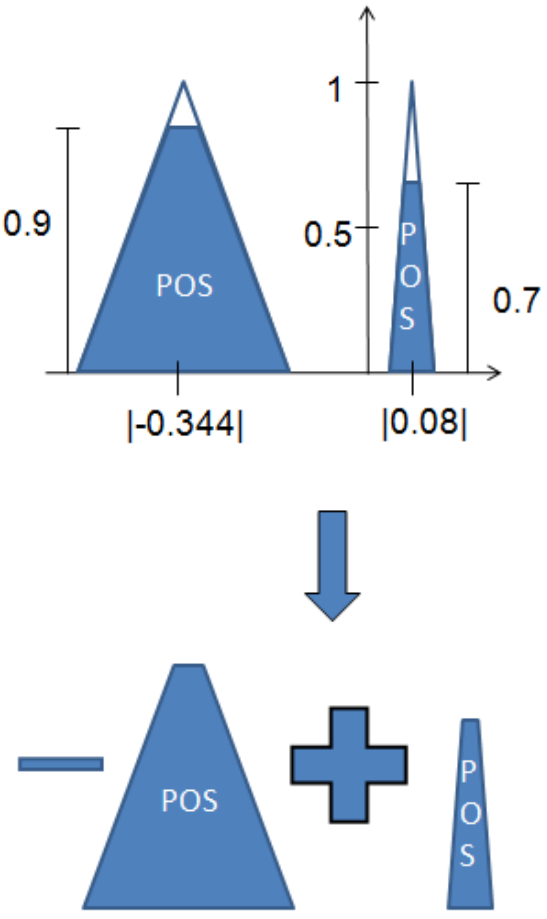


圖3-9. Bad、Wast之計算面積方法

因此類推，每一篇 Testing 文章在每一個類別底下都會有一個總和的面積值，我們將會取最大面積的類別當作此篇文章所預測的類別。如圖 3-10 所示，假設我們一篇文章在POS以及NEG兩個類別算出來的面積值總和在圖3-10的最下方，我們可以看到在 NEG 類別的面積大於在 POS 類別的面積，故我們可以知道這篇文章將會被本系統分類到 NEG 的類別中。

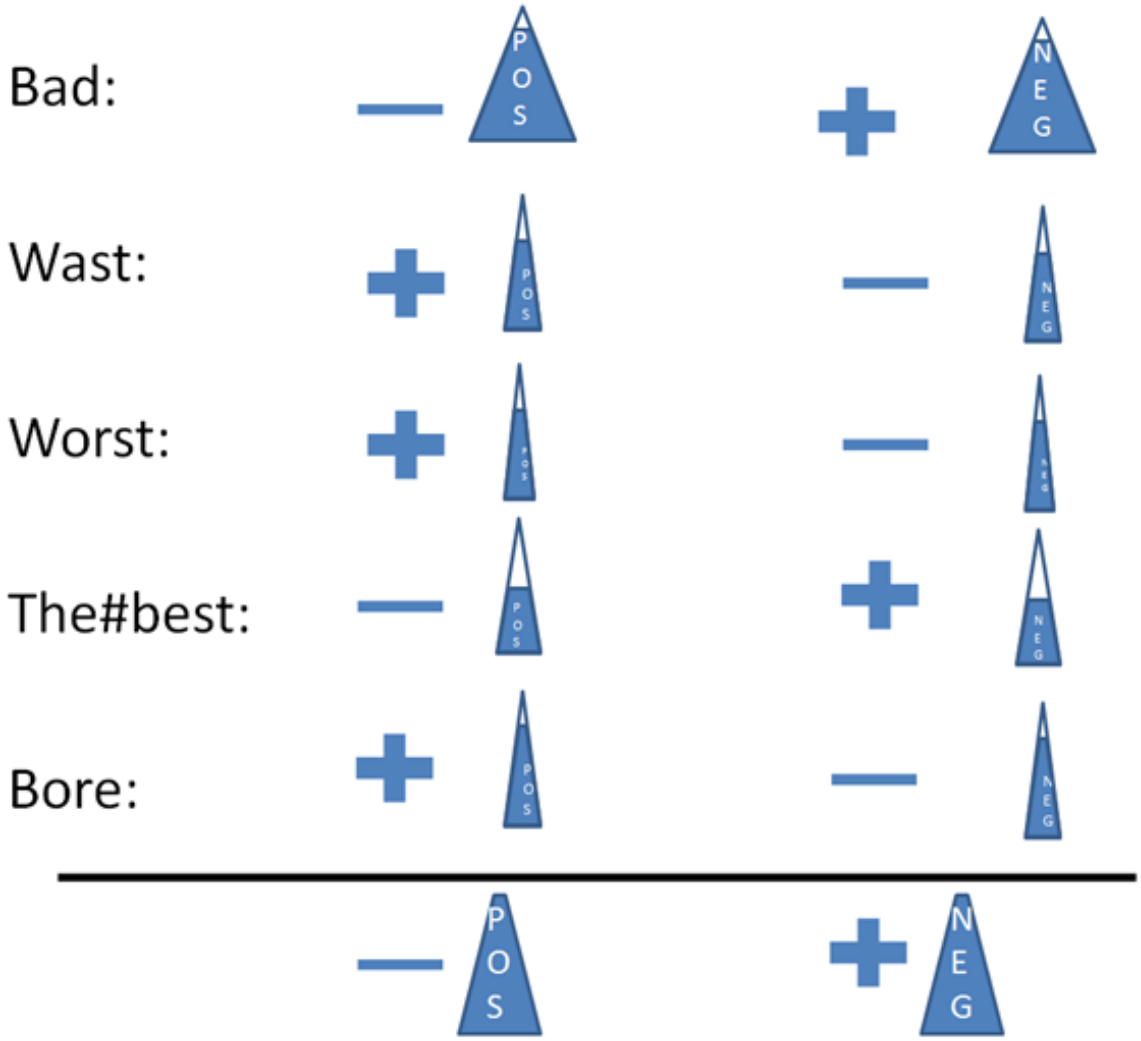


圖3-10.舉例的Testing Data文章所計算出POS、NEG類別的總值

最後我們要將文章在不同的類別中具有 Degree 的概念，所以我們必須對所有的 Testing Data 進行正規化的處理。由 Degree Transformation 的演算法求出每一篇 Testing Data 在各個類別中的 Degree。以下為計算 Degree 之步驟的例子，詳細步驟如下:

EX:假設下表為 Testing Data 中的其中兩篇文章在 POS 以及 NEG 兩個類別底下經由 Fuzzy Confidence Calculation 步驟所計算出來的面積值。

| | POS | NEG |
|---|---|---|
| Testing Data 1 | 15 | -15 |
| Testing Data 2 | -5 | 5 |

假設此 Testing Data 中的 $S_{\max}=15, S_{\min}=-15$，接著將所有的值加上 $(-1)\times S_{\min}+1$，因為 $S_{\min}<0$，其所給的兩篇例子結果如下表。

| | POS | NEG |
|---|---|---|
| Testing Data 1 | 31 | 1 |
| Testing Data 2 | 11 | 21 |

可計算出 $S_{average}=16$，假設經由 Degree Transformation 的步驟之後，兩篇 Testing Data 的結果如下表所示。

| | POS | NEG |
|---|---|---|
| Testing Data 1 | 0.89 | 0.16 |
| Testing Data 2 | 0.46 | 0.76 |

圖 3-11、3-12 分別為正規化形成 Degree 的結果的兩篇 Testing Data 1 和 Testing Data 2，可清楚呈現出兩篇 Testing Data 在兩種類別的 Degree，圖中的例子為在兩種類別 POS 以及 NEG 下的 Degree 情形，可以很明顯的看出在圖 3-11 的文章偏向 POS 類別的程度比圖 3-12 的文章偏向 POS 類別的程度來的高，而且圖 3-11 的文章偏向 POS 類別的程度比偏向 NEG 類別的程度高，所以可以知道圖 3-11(Testing Data 1)的文章將會被本系統分類到 POS 的類別中，而圖 3-12(Testing Data 2)的文章將會被本系統分類到 NEG 的類別中。

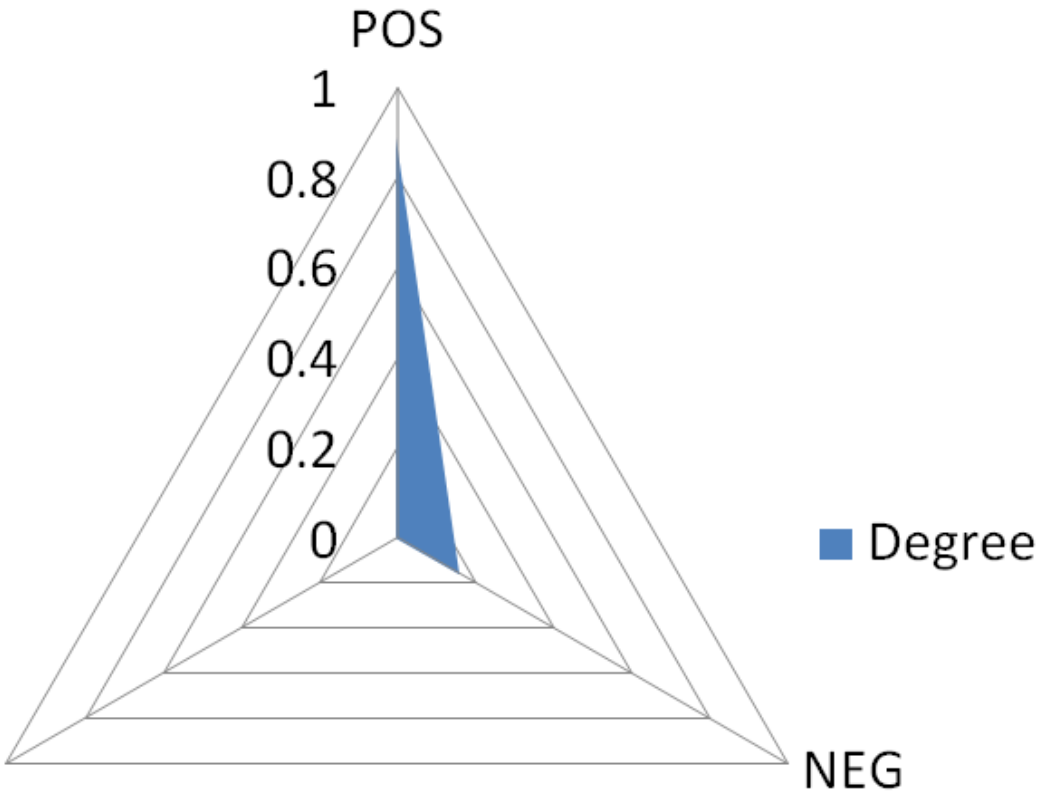


圖3-11正規化形成Degree(Testing Data 1)

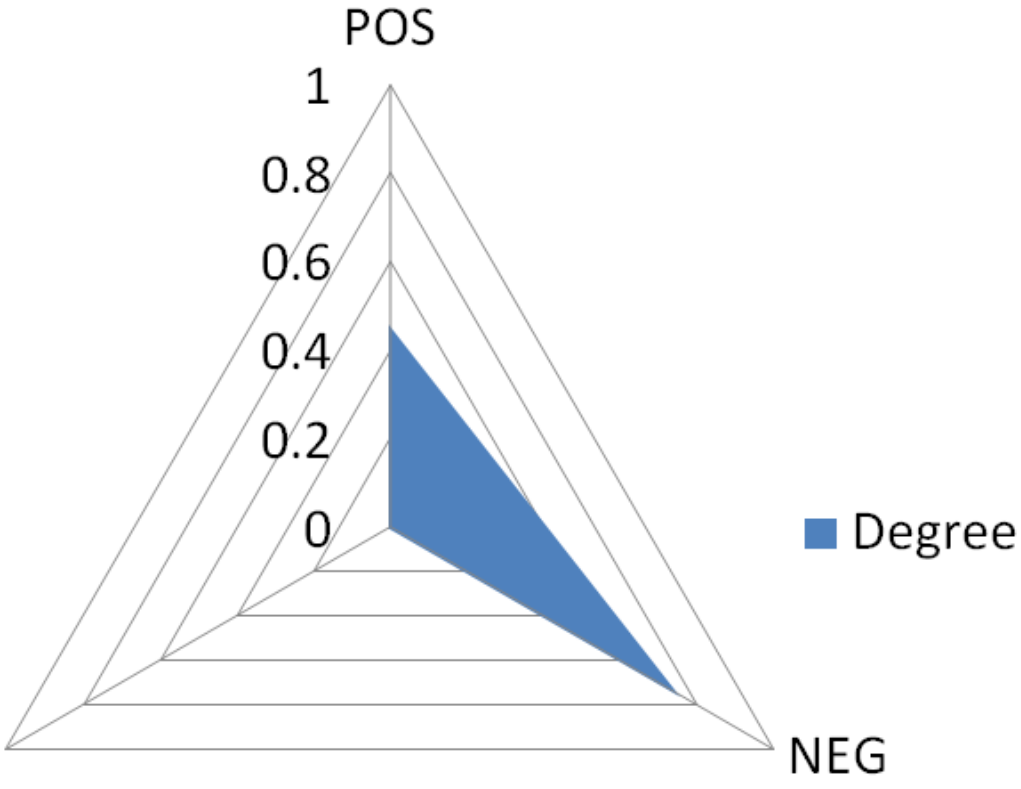


圖3-12正規化形成Degree(Testing Data 2)

## 3.5 Semi-Boosting

本小節將介紹由本系統的 Supervised 演變出的新想法 Semi-Boosting，也就是使用 Semi-Supervised 的方法對文章集進行處理。本系統 Semi-Boosting 目前只能對 Single-Label 進行處理，所以以下流程將以 Single-Label 的方式呈現。本系統 Semi-Boosting 的詳細流程可由圖 3-13 來一一作解說。

(1)Training:

假設我們現在使用 20%的 Training Data 來進行 Training，Training 的方法一樣是使用本系統的 Training 方法，使用 AdaBoost.MH 進行這個步驟，並且可以得到 Training 出來的 Hypotheses 進行下一個步驟。

(2)Testing:

第二步驟為使用 Hypotheses 對剩下的 80% Testing Data 進行預測的步驟，並且可以得到每一篇 Testing Data 在每一個不同類別底下的預測信心值，此步驟與原先本系統的 Test 步驟相同。

(3)Select:

圖 3-13 的下方有綠色以及紅色的點，分別代表 Testing Data 被分到 POS 以及 NEG 兩種類別的文章，綠色以及紅色的點中間有黑色直條線將其兩種類別區隔開來，由此黑色直條線為基準線，越往左右兩邊的點代表差異性越大，也可以說這些越外圍的點，我們可以很容易看得出來是屬於哪一個類別，代表文章趨向那一個類別的程度越大。在圖 3-13 的下方綠色以及紅色的點中，以綠色的點為例子，在所有綠色的點中，我們將其群中心的部分切一半，如 POS 類別中淺藍色的直條線，淺藍色直條線的左方與右方各有 50%的綠色點，我們可以說在淺藍色直條線左方的點，被分到 POS 類別有較大的信心度，而在淺藍色直條縣右方的點，對於被分到 POS 類別的信心度並不足夠。

(4)Recall:

本系統將圖 3-13 中被分到 POS 以及 NEG 類別信心度較大的 data，加入到

Training Data 中，剩下比較靠近中心的點則繼續放回 Testing Data 中，並且準備進行下一回合的 Training，本系統是取各類別中 50%的 data 為實驗。

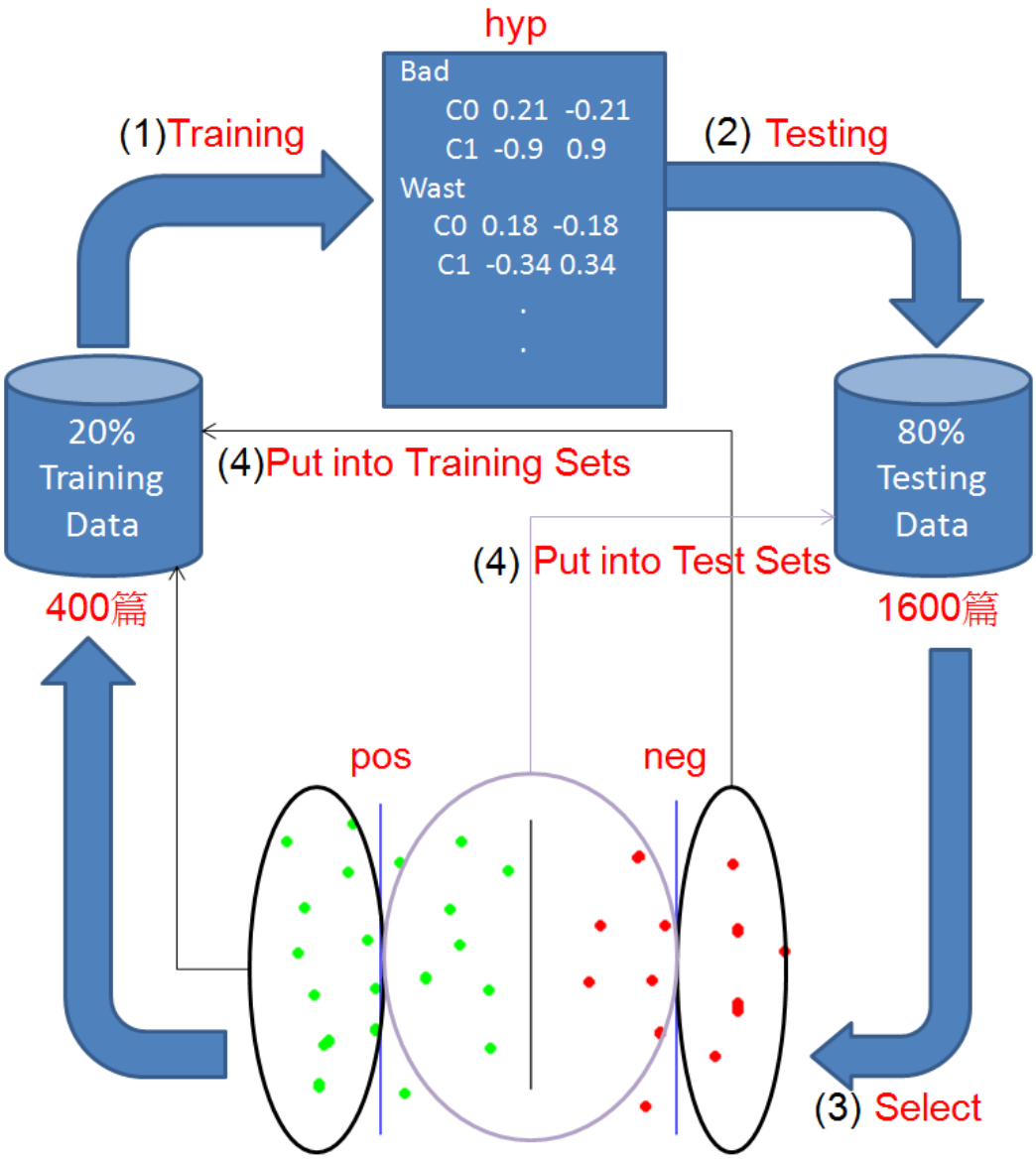


圖3-13. Semi-Boosting系統流程圖

# 第四章、實驗過程與結果討論

## 4.1 實驗資料集

本論文使用了三種不同的文章集作為估計本系統的效能，分別為 Bo Pang[1]的電影影評文章集、20 Newsgroups[2]、Reuters-21578[3]，以上三種文章集包含了 Multi-Class 以及 Multi-Label 的問題。使用三種不同的文章集，為了就是要評估本系統在不同領域的文章集中是否皆有較好的準確度。本章節將使用本系統對於以上三種文章集進行實驗，實驗方法步驟與實驗的結果將在以下章節作詳細解說，最後也會對於實驗結果進行分析與討論。

### Movie Review Data

電影的影評文章集為從網路電影的資料庫中搜集出使用者對於電影評論的文章，出自於 IMDB[4]。此文章集包含了 2000 篇的電影影評文章，其中的 1000 篇文章被標記為 POS，代表是屬於正向的影評文章，而另外的 1000 篇文章被標記為 NEG，代表是屬於負向的影評文章。本論文所用 Bo Pang 的電影影評文章版本為 polarity dataset v2.0。

### 20 Newsgroups

20 Newsgroups 是目前相當受歡迎用來評估分類方法的文章集，擁有 Single-Label 以及 Multi-Class 的特性，其文章集包含了 7 個大的主要類別，以及 20 個的子類別，每一個子類別有 1000 篇文章，總共有 20000 篇文章，本論文也將此 20 Newsgroups 文章集的標頭過濾掉，因為文章的標頭會有類別的資訊，故本論文將其刪除掉。本論文的實驗數據，將對此文章集選出幾種組合來進行實驗，包含了主要類別的分類以及子類別的分類，本論文所用的 20 Newsgroups

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[2] http://people.csail.mit.edu/jrennie/20Newsgroups/
[3] http://www.daviddlewis.com/resources/testcollections/
[4] http://www.imdb.com/

版本為 20news-19997。

Reuters-21578

本文章集包含 21578 篇文章，一共擁有 123 個類別，是屬於 Multi-Class 且 Multi-Label 的文章集，由於類別數目眾多，故本論文取文章篇數前 10 大的類別，來進行本論文的實驗，如表 4-1 為本論文實驗所取的類別之文章篇數，9108 為所取的文章總數目。

| | |
|---|---|
| acq | 2131 |
| corn | 207 |
| crude | 510 |
| earn | 3753 |
| grain | 528 |
| interest | 389 |
| money-fx | 601 |
| ship | 276 |
| trade | 449 |
| wheat | 264 |
| | 9108 |

表 4-1.Reuters-21578 文章集之各類別文章篇數

## 4.2 實驗設計

### 4.2.1 文章集的前置處理

各種文章集的內容繁雜，除了英文單字以及片語之外，包含許多數字和各種標點符號。而每一個單字在詞性上也有很多種變化，不同的詞性也有可能有不同的拼法。本論文將其文章集做統一的處理，包含以下兩個部分:

(1)只保留英文字母

文章中包含許多的標點符號以及數字，為了不讓這些不必要的內文影響了分

類的結果，故本論文將只會對文章集保留大小寫的英文字母 A 到 Z，這樣也同時會減少取 feature 時的數目，降低取到不必要資訊的可能性。

(2)進行 Stemming 處理

英文中每一個單字的將會因為不同詞性而有可能有不同種的拼法，而造成分類上的混亂，例如:play 和 plays，都是屬於同樣意思的單字，差別只在單複數主詞。故本論文為了將其統一，而對於只剩下英文字母的文章集進行 stemming 的處理。

### 4.2.2 實驗方法與參數

(1)1. Accuracy 的評估方法:

$$\text{accuracy}: \frac{\text{Test Data 中分類正確的文章篇數}}{\text{所有的 Test Data 的文章篇數}}$$

2. 多類別情況下 F-Value 之計算方法:

本論文是使用 Macro-average F-measure 來評估分類結果。假設第一個 fold 中有 A、B、C 三個類別，其分類之結果如圖 4-1 所示。計算方法為先計算出各類別之 precision、recall、F-Value，之後再計算出 F-Value 之平均值，由於本論文使用 5-fold 形式，故最後需再將每一個 fold 所計算出的 F-Value 加總，再除以 5 取平均值。

| | | Predicted Class | | |
|---|---|---|---|---|
| | | **A** | **B** | **C** |
| Known Class | **A** | 25 | 5 | 2 |
| | **B** | 3 | 32 | 4 |
| | **C** | 1 | 0 | 15 |

圖4-1. F-Value計算例圖(一)

以下所計算出的 F-Value 為文章集中第一個 fold 的 F-Value 值。

$$Precision_A = 25/(25+3+1)$$

$$Recall_A = 25/(25+5+2)$$

$$F-Value_A = \frac{2\times Precision_A \times Recall_A}{Precision_A + Recall_A}$$

$$F-Value_{first-flod} = \frac{1}{3}\left(F-Value_A + F-Value_B + F-Value_C\right)$$

之後必須對 5-fold 的 F-Value 進行平均，就可得到此文章集之 F-Value 的值。

$$F-Value = \frac{1}{5}\sum_{i=1}^{5} F-Value_i$$

3. Multi-Label 情況下 F-Value 之計算方法：

假設目前文章分類情況如圖 4-1。若現在有一篇新的文章其類別為{A, B}，但是卻被系統分到類別{C}中，則圖 4-1 更新之後如圖 4-2。

| | | Predicted Class | | |
|---|---|---|---|---|
| | | **A** | **B** | **C** |
| Known Class | **A** | 25 | 5 | 2+1 |
| | **B** | 3 | 32 | 4+1 |
| | **C** | 1 | 0 | 15 |

圖4-2. F-Value計算例圖(二)

假設目前文章分類情況如圖 4-1。若現在有一篇新的文章其類別為{A, B}，而系統分到類別{A}中，則圖 4-2 更新之後如圖 4-3。

| | | Predicted Class | | |
|---|---|---|---|---|
| | | A | B | C |
| Known Class | A | 25+1 | 5 | 2 |
| | B | 3 | 32+1 | 4 |
| | C | 1 | 0 | 15 |

圖4-3. F-Value計算例圖(三)

4. 時間之計算單位:

[時:分:秒]

(2)N-Gram Model 取 Feature 方法解說

本論文選取 feature 時，將有可能使用不只是 unigram model 甚至使用 bigram model、trigram model 來取 feature，本系統並無限制只能對文章選取到多少 N-Gram，而本論文實驗數據最多使用到 trigram model 的選取 feature 方法，因為如果再繼續增加取 feature 的 N-Gram 並沒有增加太多的準確度，可能導致準確度下降，也會耗費太多記憶體，增加處理時間等壞處。圖 4-4 為本論文的 N-Gram Model 取 feature 方法解說圖。本系統將只對於相鄰的字做 N-Gram，因為若是要考慮到各種文字的排列組合，則總體的數量會以指數倍率增加，如此龐大的數量將會造成系統的整理效能下降，故本系統採取相鄰的方法，減少資源的使用量。

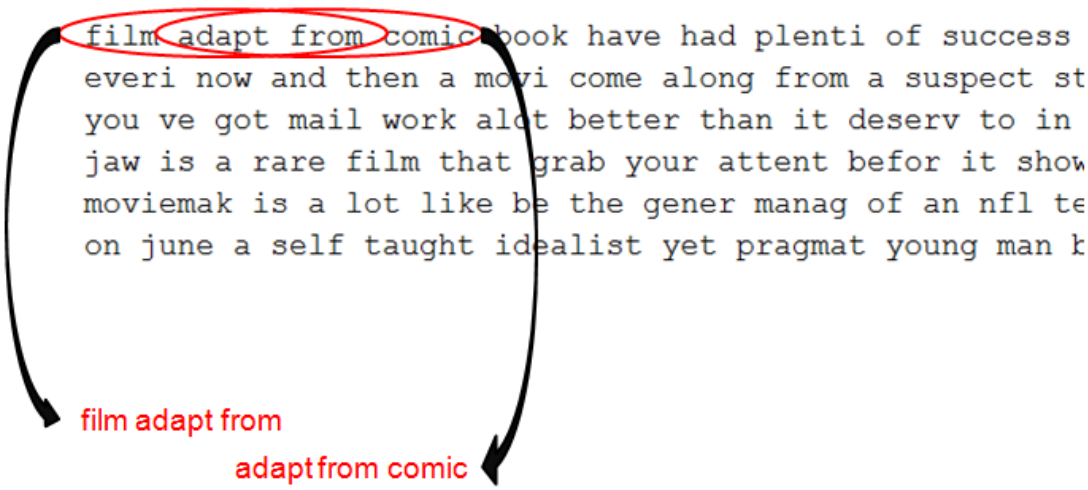


圖4-4. 本系統N-Gram Model解說

(3)篩選 Feature 的方法

即使本系統在 N-Gram Model 的取字方法下，所選取的 feature 依然很多，若是要將所有的 feature 進行 training，將會耗費相當大的時間。本論文在所有經過 N-Gram Model 建立的 feature 中，計算每一個 feature 在所有 training data 的出現次數或是出現篇數。若此文章集為兩群，則本系統將會將每一個 feature 在兩個不同的類別的出現次數相減，若大於所設定的門檻，則將此 feature 加入到系統的 feature List 中。若此文章集為多群的文章集，則將會檢查此 feature 在所有 training data 中的出現篇數是否達到門檻，若有，則將此 feature 加入到系統的 feature List 中。在兩群的文章中，若同一 feature 在兩個不同的類別中出現次數有一定的差距時，可以代表此 feature 在此兩類別中有可能有較大的辨別度。而在多群的文章集中，則無法使用兩群的文章集的取法，故使用出現篇數大於一定以上；因為若出現篇數太小，代表此 feature 重點度不大，故將此 feature 排除在外。

## 4.3 實驗結果

### 4.3.1 本系統之實驗結果

本論文之系統的實驗皆使用 5-fold 的形式，且每一個 fold 皆分成 80%之 Training Data 以及 20%之 Testing Data，本系統所 training 之 Round 數目、N-Gram Model 以及使用在篩選 feature 之門檻皆為本系統之參數，根據不同文章集將可能會有不同的參數。本論文之實驗除了比較準確度之外，也將實驗的時間納入考量，為的就是要在準確度以及時間上做比較。

表 4-2 為 Bo Pang 電影影評文章集之數據，本實驗將會把文章中內文的't 改成 not，以突顯否定詞於影評中的影響力。本系統所 training 之 Round 數目為 3000 次，且使用最多到 trigram Model 的取 feature 方式，而篩選 feature 的門檻在本文章集設定為 10 次，代表 feature 需出現在 POS 以及 NEG 兩個類別的出現次數差別大於 10 次才會被選取。Accuracy Ratio Calculation Algorithm 中的參數 a 由參數 c 所計算出，參數 c 設為 1.4，參數 b 設為 0.675。表中 Fuzzy AdaBoost.MH 為本論文的模糊化系統，Naïve Bayes[11]使用所有 unigram 的 feature 進行實驗，而 Support Vector Machine(SVM)[9][10]為使用所有 unigram 且 feature 在兩類別中出現次數需大於 3 次的 feature 進行實驗。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | Naïve Bayes | SVM |
|---|---|---|---|---|
| Pang | 84.2% | 84.9% | 82.65% | **85.55%** |

表 4-2. 電影影評文章集之實驗數據

表 4-3 為本系統與 Support Vector Machine(SVM) 在電影影評文章集之實驗時間表。

本章節中所有的時間表形式皆為[時:分:秒]。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | SVM |
|---|---|---|---|
| Pang | 3:48:51 | 3.58.05 | 5:49:45 |

表 4-3. 電影影評文章集實驗時間比較表[時:分:秒]

表 4-4 為本系統與 Support Vector Machine(SVM)以及 Naïve Bayes(NB)在電影影評文章集之 F-Value 比較表。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | Naïve Bayes | SVM |
|---|---|---|---|---|
| Pang | 0.8420 | 0.8489 | 0.8264 | **0.8554** |

表 4-4. 電影影評文章集之 F-Value

表 4-5 為 20 Newsgroups 文章集之實驗數據。表中的 Dataset 欄位為 20 Newsgroups 文章集中所挑選出的實驗組合。本系統所 training 之 Round 數目的範圍為 500~1300 次，且使用最多到 bigram model 的取 feature 形式。Accuracy Ratio Calculation Algorithm 中的參數 c 設為 *Max* / *Min* 的值，故參數 a 為 1，參數 b 設為 1。表中 Fuzzy AdaBoost.MH 為本論文的模糊化系統，表中 Naïve Bayes 以及 Support Vector Machine(SVM)所實驗之 feature 與本系統挑選之 feature 相同，使用相同的最多到 bigram Model 的取 feature 形式以及設定 feature 出現篇數的門檻來篩選，篩選門檻值為 15~20。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | Naïve Bayes | SVM |
|---|---|---|---|---|
| comp.graphics, rec.autos, sci.crypt, talk.politics.guns | 92.18% | **92.28%** | 92.13% | 90.58% |
| comp.graphics, comp.os.ms-windows.misc, comp.windows.x, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware | 76.60% | **77.08%** | 73.88% | 74.34% |
| rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey | 92.58% | **92.73%** | 91.95% | 90.53% |
| sci.crypt, sci.electronics, sci.med, sci.space | 90.85% | **91.23%** | 90.58% | 88.28% |
| talk.politics.guns, talk.politics.mideast, talk.politics.misc | 81.23% | 81.97% | 81.90% | **82.80%** |

表 4-5. 20-Newsgroups 之多種組合實驗數據

表 4-6 為 20 Newsgroups 文章集在本系統以及 SVM 實驗時間比較表。不同種組合的實驗時間會因為 training 的 Round 數目不同以及 feature 所取的數目不同而有不同的實驗時間，所以即使分類的群數目相同，也有可能有時間上的差距。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | SVM |
|---|---|---|---|
| comp.graphics, rec.autos, sci.crypt, talk.politics.guns | 9:44:01 | 9:47:0 | 13.42.59 |
| comp.graphics, comp.os.ms-windows.misc, comp.windows.x, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware | 12:0:40 | 12:3:10 | 17:28:29 |
| rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey | 3:27:21 | 3:28:10 | 10:27:08 |
| sci.crypt, sci.electronics, sci.med, sci.space | 8:19:06 | 8:21:03 | 12:43:57 |
| talk.politics.guns, talk.politics.mideast, talk.politics.misc | 4:58:13 | 5:01:0 | 9:08:58 |

表 4-6. 20-Newsgroups 之多種組合實驗時間比較表[時:分:秒]

表 4-7 為本系統與 Support Vector Machine(SVM)以及 Naïve Bayes(NB)在 20 Newsgroups 文章集之 F-Value 比較表。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | Naïve Bayes | SVM |
|---|---|---|---|---|
| comp.graphics, rec.autos, sci.crypt, talk.politics.guns | 0.9217 | **0.9227** | 0.9216 | 0.9060 |
| comp.graphics, comp.os.ms-windows.misc, comp.windows.x, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware | 0.7660 | **0.7707** | 0.7335 | 0.7442 |
| rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey | 0.9259 | **0.9274** | 0.9197 | 0.9056 |
| sci.crypt, sci.electronics, sci.med, sci.space | 0.9089 | **0.9126** | 0.9066 | 0.8836 |
| talk.politics.guns, talk.politics.mideast, talk.politics.misc | 0.8131 | 0.8202 | 0.8195 | **0.8283** |

表 4-7. 20-Newsgroups 之多種組合之 F-Value

表 4-8 為 Reuters 文章集之實驗數據。本系統所 training 之 Round 數目為 500 次，且使用最多到 bigram model 的取 feature 形式以及設定 feature 出現篇數的門檻來篩選，篩選門檻值為 20。Accuracy Ratio Calculation Algorithm 中的參數 c 設為 *Max* / *Min* 的值，故參數 a 為 1，參數 b 設為 0.1。表中 Fuzzy AdaBoost.MH 為本論文的模糊化系統，表中 Naïve Bayes 以及 Support Vector Machine(SVM)所實驗之 feature 也是與本系統相同。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | Naïve Bayes | SVM |
|---|---|---|---|---|
| Reuters | 96.17% | **96.20%** | 90.87% | 94.09% |

表 4-8. Reuters 文章集之實驗數據

表 4-9 為本系統與 Support Vector Machine(SVM)在 Reuters 文章集之實驗時間表。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | SVM |
|---|---|---|---|
| Reuters | 14:10:43 | 14:11:52 | 37:46:46 |

表 4-9.Reuters 文章集實驗時間比較表[時:分:秒]

表 4-10 為本系統與 Support Vector Machine(SVM)以及 Naïve Bayes(NB)在 Reuters 文章集之 F-Value 比較表。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | Naïve Bayes | SVM |
|---|---|---|---|---|
| Reuters | **0.9511** | 0.9510 | 0.8965 | 0.9413 |

表 4-10.Reuters 文章集之 F-Value

### 4.3.2 本系統在小資料量 Training Data 之實驗結果

此小節將對本系統在小資料量 Training Data 的情況下進行實驗，本實驗也皆使用 5-fold 的形式，且每一個 fold 皆分成 20%之 Training Data 以及 80%之 Testing Data，本系統所 training 之 Round 數目、N-Gram Model 以及使用在篩選 feature 之門檻皆為本系統之參數，根據不同文章集將可能會有不同的參數。主要目的是要評估模糊化系統是否在比較小量的 Training Data 中，比原先 AdaBoost.MH 有更高的準確度。

表 4-11 為 20 Newsgroups 文章集之實驗數據。表中的 Dataset 欄位為 20 Newsgroups 文章集中所挑選出的實驗組合。本系統所 training 之 Round 數目皆為 1500 次，且使用最多到 bigram model 的取 feature 形式。Accuracy Ratio Calculation Algorithm 中的參數 c 設為 *Max* / *Min* 的值，故參數 a 為 1，參數 b 設為 1。表中 Fuzzy AdaBoost.MH 為本論文的模糊化系統，表中 Naïve Bayes 以

及 Support Vector Machine(SVM)所實驗之 feature 與本系統挑選之 feature 相同，使用相同的最多到 bigram Model 的取 feature 形式以及設定 feature 出現篇數的門檻來篩選，篩選門檻為 15~20。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | Naïve Bayes | SVM |
|---|---|---|---|---|
| comp.graphics, rec.autos, sci.crypt, talk.politics.guns | 83.26% | **84.88%** | 84.01% | 80.39% |
| comp.graphics, comp.os.ms-windows.misc, comp.windows.x, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware | 63.47% | **65.90%** | 61.54% | 61.05% |
| rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey | 83.44% | **85.84%** | 82.92% | 79.17% |
| sci.crypt, sci.electronics, sci.med, sci.space | 78.42% | **80.11%** | 79.91% | 73.31% |
| talk.politics.guns, talk.politics.mideast, talk.politics.misc | 68.35% | **70.80%** | 67.93% | 67.57% |

表 4-11. 20-Newsgroups 之小資料量多種組合實驗數據

表 4-12 為本系統與 Support Vector Machine(SVM)以及 Naïve Bayes(NB)在 20 Newsgroups 文章集之 F-Value 比較表。

| Dataset | AdaBoost.MH | Fuzzy AdaBoost.MH | Naïve Bayes | SVM |
|---|---|---|---|---|
| comp.graphics, rec.autos, sci.crypt, talk.politics.guns | 0.8328 | **0.8490** | 0.8401 | 0.8040 |
| comp.graphics, comp.os.ms-windows.misc, comp.windows.x, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware | 0.6358 | **0.6601** | 0.6115 | 0.6130 |
| rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey | 0.8347 | **0.8588** | 0.8293 | 0.7925 |
| sci.crypt, sci.electronics, sci.med, sci.space | 0.7856 | **0.8026** | 0.8002 | 0.7345 |
| talk.politics.guns, talk.politics.mideast, talk.politics.misc | 0.6848 | **0.7097** | 0.6809 | 0.6770 |

表 4-12. 20-Newsgroups 之小資料量多種組合之 F-Value

### 4.3.3 Semi-Boosting 之實驗結果

Semi-Boosting 的實驗結果以 20 Newsgroups 以及 Reuters 的文章集來做實驗評估，而 Training Data 從 5%、4%、3%、2%、1%皆有進行實驗，此系統的實驗在每一個循環皆會有約 50%的 Testing Data 加入到 Training Data 中，所以每一次循環皆會少掉約 50%的 Testing Data，而本實驗設定當文章數目小於 4 篇時，則系統將會停止。Semi-Boosting 的實驗皆使用 unigram Model 來取 feature，並且設定 feature 出現篇數的門檻來篩選所取的 feature，篩選門檻值為 4~6。每一循環所 Training 的 Round 數目為 500。

表 4-13、4-14 為 Semi-Boosting 在 20 Newsgroups 文章集的實驗數據，表中使用 Graph-based Semi-supervised[24]來進行實驗數據比較。表中為 20 Newsgroups 文章集中所選出實驗的組合，欄位%為使用多少%的文章篇數來當作 Training Data。

表 4-13 為 20 Newsgroups 文章集中"talk.politics.guns"以及"talk.politics.mideast"的組合

| % | Semi-Boosting | Graph-based Semi-supervised |
|---|---|---|
| 5% | **84.99%** | 77.35% |
| 4% | **82.90%** | 74.875% |
| 3% | **80.58%** | 72.55% |
| 2% | **71.82%** | 63.45% |
| 1% | 59.23% | **61.20%** |

表 4-13. Semi-Boosting 之 20-Newsgroups 實驗數據(一)

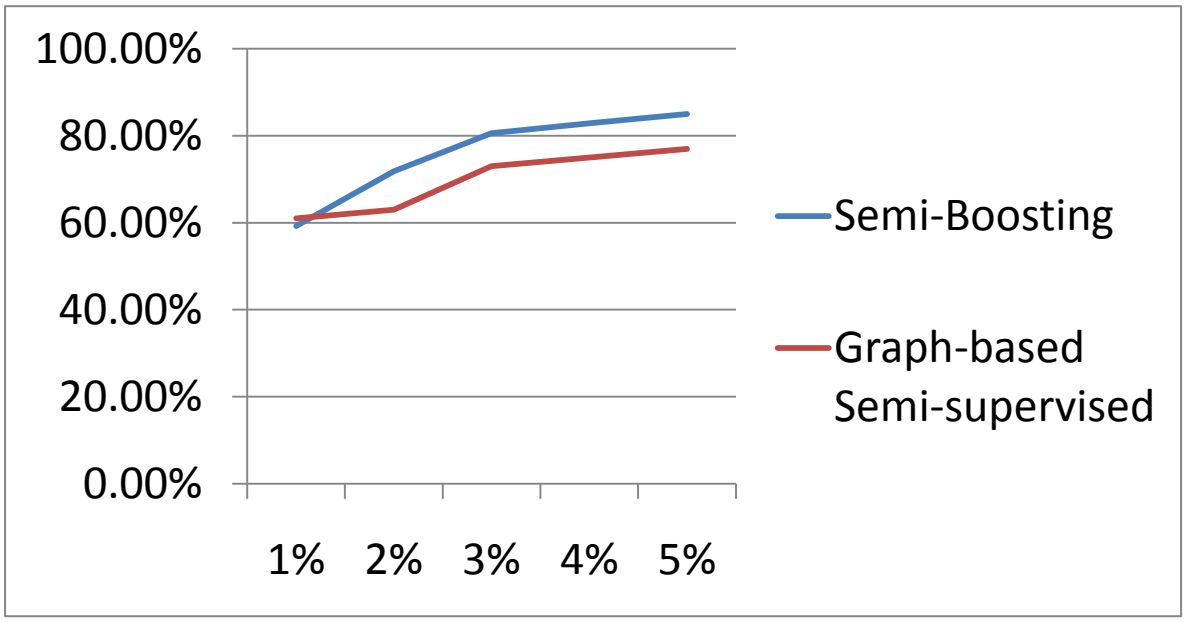


圖4-5. Semi-Boosting之20-Newsgroups成長圖表(一)

表 4-14 為 20 Newsgroups 文章集中"talk.politics.guns", "talk.politics.mideast","talk.politics.misc","talk.religion.misc"的組合

| % | Semi-Boosting | Graph-based Semi-supervised |
|---|---|---|
| 5% | **59.21%** | 41.83% |
| 4% | **59.45%** | 41.93% |
| 3% | **54.18%** | 37.90% |
| 2% | **50.42%** | 35.00% |
| 1% | **35.04%** | 30.20% |

表 4-14. Semi-Boosting 之 20-Newsgroups 實驗數據(二)

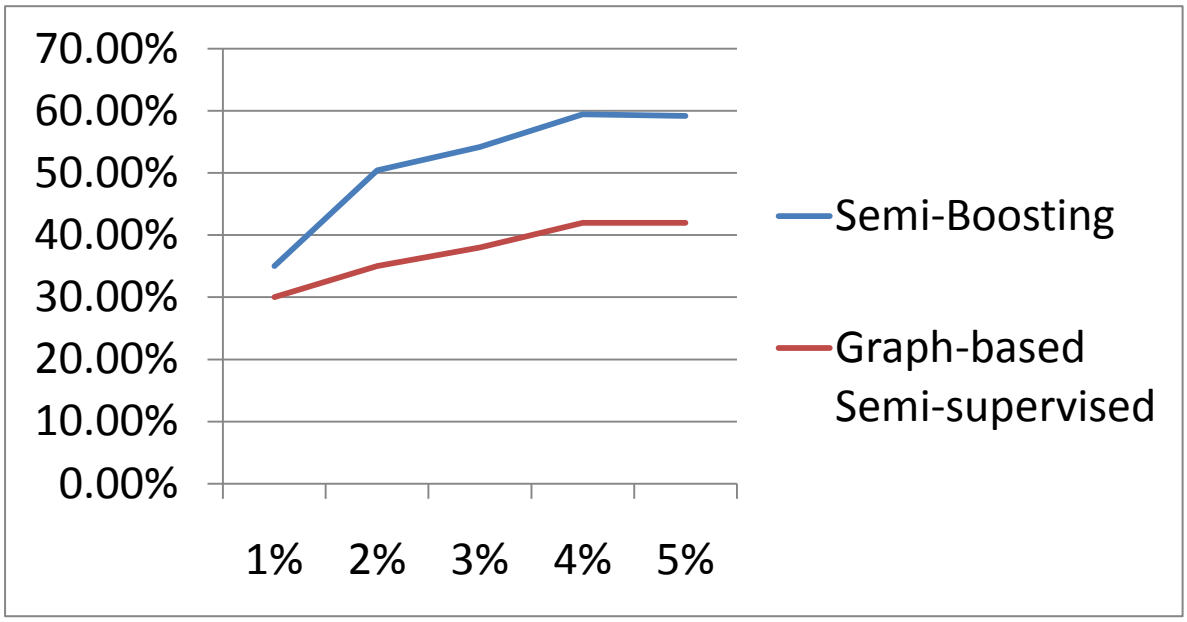


圖4-6. Semi-Boosting之20-Newsgroups成長圖表(二)

表 4-15 為 Semi-Boosting 在 Reuters 文章集的實驗數據，表中使用 Graph-based Semi-supervised[24]來進行實驗數據比較。欄位%為使用多少%的文章篇數來當作 Training Data。

| % | Semi-Boosting | Graph-based Semi-supervised |
|---|---|---|
| 5% | **90.14%** | 34.84% |
| 4% | **88.91%** | 33.49% |
| 3% | **87.55%** | 31.04% |
| 2% | **83.00%** | 29.09% |
| 1% | **72.76%** | 14.47% |

表 4-15. Semi-Boosting 之 Reuters 文章集實驗數據

表 4-16 為本系統與 Graph-based Semi-supervised 在 Reuters 文章集之 F-Value 比較表。

| % | Semi-Boosting | Graph-based Semi-supervised |
|---|---|---|
| 5% | **0.8744** | 0.1587 |
| 4% | **0.8528** | 0.1545 |
| 3% | **0.8234** | 0.1492 |
| 2% | **0.7544** | 0.1331 |
| 1% | **0.5522** | 0.1177 |

表 4-16. Semi-Boosting 之 Reuters 文章集 F-Value

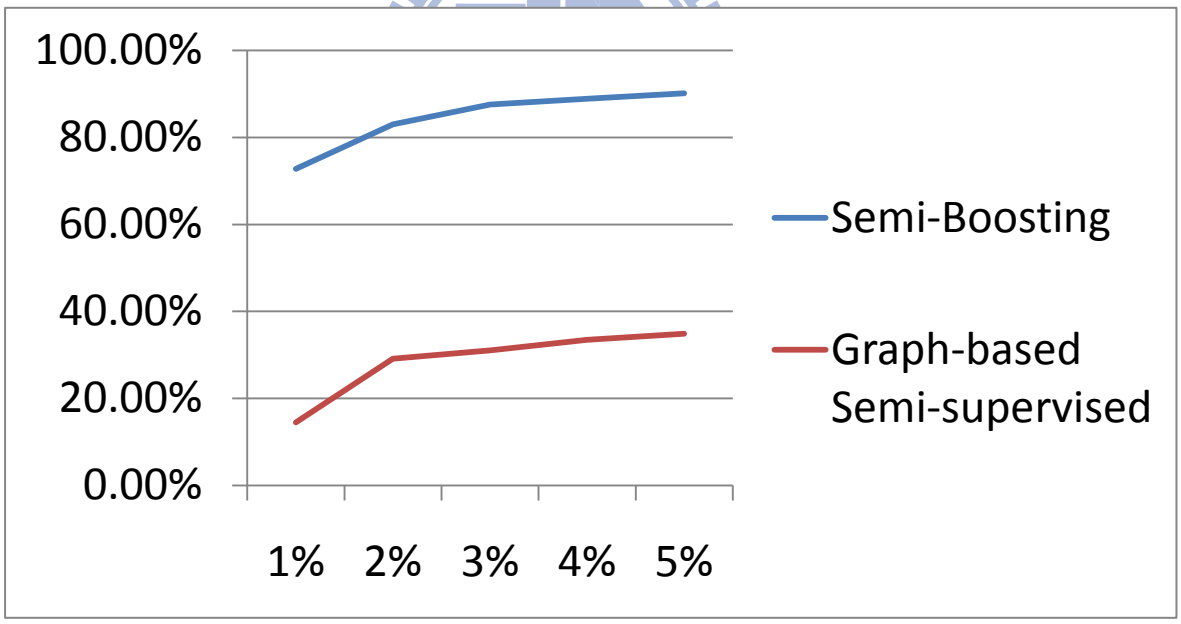


圖4-7. Semi-Boosting之Reuters成長圖表

## 4.4 實驗討論

### 4.4.1 本系統之實驗數據討論

本系統對三種文章集進行實驗，在 Bo Pang 的文章集中，由表 4-2 可以知道，本系統的準確度方面比原先的 AdaBoost.MH 增加了 0.7%，也比 Naïve Bayes 分類法好上 2.25%，雖然準確度與 SVM 分類法相較之下差了一些，不過我們由表 4-3 可以很清楚的看到，SVM 分類法估計參數的時間比本系統的處理時間長，且本系統處理時間比原先 AdaBoost.MH 的處理時間只有微量增加，卻能有效提升準確度。

而從表 4-5 中的各種 20 Newsgroups 的文章集之組合實驗數據可以看到，本系統之準確度皆比 AdaBoost.MH 分類法還要好，且各種組合皆優於 Naïve Bayes 分類法，而在 SVM 分類法中，也只有一種組合優於本系統。處理時間方面，由表 4-6 可以很清楚的看到，SVM 分類法估計參數的時間遠大於本系統之處理時間，這代表說如果要使用 SVM 分類法，或許在某些情況下會有較高的準確度，但是卻需要花費大量的處理時間才能完成，而在準確度與處理時間兩者方面若是要取得最大效益，還是本系統擁有優勢。

對於 Multi-Label 之文章集，由表 4-8 的 Reuters 文章集可以看出，對於 Multi-Label 的情況下，本系統之準確度還是比 Naïve Bayes 分類法和 SVM 分類法來的好，且比 Naïve Bayes 分類法高上 5.33%的準確度。再表 4-9 的處理時間中，雖然本系統花上 14 個小時，但是 SVM 分類法卻要花上 37 個小時來估計參數，整整快上 2 倍以上的時間，這代表說 SVM 分類法若是要對多群以及多 Label 之文章集進行處理時，需要花上的時間將會很大。

本論文也對小資料量進行了實驗，從先前的表中我們可以看到，在 Training Data 為 80%的情況下，本系統之實驗準確度比 AdaBoost.MH 的準確度沒有大量的增加。但我們由表 4-11 可以看到，在 5 種的文章集組合情況下，小資料量的情況下平均準確度比 AdaBoost.MH 分類法高於 2%以上，此現象可以說若是在 Training Data 數量越小時，本系統的準確度將有可能會越高，也具有較高的優

化效果。

### 4.4.2 Semi-Boosting 之實驗數據討論

對於 Semi-Boosting 的系統，由表 4-13、4-14 可以看出此系統的分類方法比 Graph-based Semi-supervised 分類法要來的好，除了在表 4-13 中 1%的 Training Data 情況下略低於 Graph-based Semi-supervised 分類法外，其餘情況下，準確度皆遠高於 Graph-based Semi-supervised 分類法。而在表 4-15 可以很明顯的看出來，本系統在 Reuters 文章集的準確度遠比 Graph-based Semi-supervised 分類法來的好。Semi-Boosting 系統目前還是屬於實驗性的階段，因為在處理時間上需要花上較常的時間，如何減少處理的時間，以及每一次循環是否一定要 Remove 掉 50%的 Testing Data，或是在篩選 feature 時的門檻該設定多少等等，都是在未來可以有更多的想法和嘗試，除此之外，其新穎的想法也可以在往後有更多的運用和提升效能。

# 第五章、結論與未來展望

## 5.1 研究總結

本篇論文主要介紹模糊化分類系統，目的是能夠對於文章的分類有更高的準確度以及節省分類的處理時間。從實驗的數據結果顯示，本系統在不同的文章集中，在準確度上皆擁有不錯的分類結果，且分類結果也都比原先 AdaBoost.MH 分類法來的好，雖然在有些實驗的準確度上 SVM 分類法些微高於本系統分類法，但是在分類的處理時間方面，本系統的模糊化分類法速度遠遠快於 SVM 分類法，如果使用者需要較高的準確度和節省更多的時間，相較之下明顯還是本系統具有較大的優勢。使用模糊化技術不僅僅增加了的準確度，在時間方面也只有些微的增加，突顯了模糊化技術使用在分類法上的好處與優點。而在 Semi-Boosting 的部份，由於是屬於一個新穎的想法，在實驗數據上也有不錯的結果，目前還是屬於實驗性的階段，在未來可以有更多的想法和嘗試。

## 5.2 未來展望

由於本系統是以 AdaBoost.MH 演算法為基礎結合模糊化方法的分類法，在本論文中本系統的實驗結果皆比原先 AdaBoost.MH 好，但是其效果卻有限，而在未來中，或許有更多的空間能夠因為加了模糊化而有比原先的 AdaBoost.MH 有更高的準確度，而在處理時間方面也能夠更為減少。另外，在 Semi-Boosting 的部份，雖然準確度有不錯的結果，但是在處理時間方面還有改善的空間，未來除了在時間方面，準確度方面或許也有進展的可能性。此外，Semi-Boosting 未來也可以與模糊化方法做結合，延伸出新的分類法。

# 參考文獻


[1] R. Polikar, "Ensemble Based Systems in Decision Making", IEEE Circuits and Systems Magazine, vol.6, no.3, pp. 21-45, 2006.

[2] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119 - 139, 1997.

[3] R. E. Schapire, "A brief introduction to boosting," in *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 1401 - 1406.

[4] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39,no. 2/3, pp. 135 - 168, 2000.

[5] X. Carreras, L. S. Marquez, and J. G. Salgado, "Boosting trees for anti-spam email filtering," in *In Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing, Tzigov Chark, BG*, 2001,pp. 58 - 64.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 1, p. 511, 2001.

[7] X. Carreras, L. M'arquez, and L. Padr´o, "Named entity extraction using adaboost," in *COLING-02: proceedings of the 6th conference on Natural language learning*. Morristown, NJ, USA: Association for Computational Linguistics, 2002, pp.1 - 4.

[8] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Mach. Learn.*, vol. 37, pp. 297 - 336, December 1999. [Online]. Available: http://portal.acm.org/citation.cfm?id=337859.337870

[9] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European Conference on Machine Learning (ECML)*. Berlin: Springer, 1998, pp. 137 - 142.

[10] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[11] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*. AAAI Press, 1998, pp. 41 - 48.

[12] L. A. Zadeh, "Outline of a new approach to the analysis of complex

systems and decision processes," *IEEE Trans. On Sys., Man and Cybern.*, vol. SMC-3, pp. 28–44, 1973.

[13] ——, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965. [Online]. Available: http://www-bisc.cs.berkeley.edu/Zadeh-1965.pdf

[14] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997.

[15] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*. New York, NY, USA: John Wiley & Sons, Inc., 1997.

[16] M. J. del Jes´us, F. Hoffmann, L. J. Navascu´es, and L. S´anchez, "Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms," *IEEE T. Fuzzy Systems*, vol. 12, no. 3, pp. 296–308, 2004.

[17] Y. Yuan and H. Zhuang, "A genetic algorithm for generating fuzzy classification rules," *Fuzzy Sets Syst.*, vol. 84, pp.1–19, November 1996. [Online]. Available: http://dx.doi.org/10.1016/0165-0114(95)00302-9

[18] J. A. Roubos, M. Setnes, and J. Abonyi, "Learning fuzzy classification rules from labeled data," *Inf. Sci. Inf. Comput. Sci.*, vol. 150, pp. 77–93, March 2003. [Online]. Available: http://portal.acm.org/citation.cfm?id=763284.763290

[19] F. Hoffmann, "Combining boosting and evolutionary algorithms for learning of fuzzy classification rules," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 47–58, 2004.

[20] J. Otero and L. Sanchez, "Induction of descriptive fuzzy classifiers with the logitboost algorithm," *Soft Comput.*, vol. 10,pp. 825–835, May 2006. [Online]. Available: http://portal.acm.org/citation.cfm?id=1127062.1127065

[21] R. Scherer, "Designing boosting ensemble of relational fuzzy systems," *Int. J. Neural Syst.*, vol. 20, no. 5, pp. 381–388,2010.

[22] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting," *The Annals of Statistics*, vol. 38, no. 2, 2000.

[23] *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[24] A. B. Goldberg and X. Zhu, "Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment

categorization," in Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, ser. TextGraphs-1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 45–52. [Online]. Available: http://portal.acm.org/citation.cfm?id=1654