國立臺灣大學電機資訊學院

資訊網路與多媒體研究所

碩士論文

Graduate Institute of Networking and Multimedia

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

對不平衡的資料有效率的訓練和自我訓練的門檻分析

Efficient Training for Imbalance Data and Threshold Analysis for Self-training

張峻銘

Chang Chun Min

指導教授：林守德 博士

Advisor: Shou-De Lin, Ph.D.

中華民國九十八年七月

July, 2009

摘要

本篇論文提出兩個方法解決在不平衡資料下的分類問題。首先提出利用自我訓練時，比之前自我訓練的方法少的參數維度以及提升效能的方法。透過在已標記的訓練資料上，對每個分類器單獨訓練出預測信心值門檻，用來分辨高信心的未標記資料，並將結果作聯集給它們虛擬類別加入已標記資料中重新訓練。藉此不但降低了選參數的時間，效能也跟複雜的參數差不多。再者我們提出有效率地訓練不平衡資料的方法，從速度快的 down-sampling 開始透過類似 booststrap 的方法，將模型逼近得與 up-sampling 一樣，由於使用的資料量少，速度獲得了提升。我們在 KDD cup 2008 的極端不平衡資料中爲它們實驗，實驗結果顯示在自我訓練中我們的方法選擇參數表現較之前方法稍好；而在效率上提出的方法是直接使用 up-sampling 的 1.3 倍快，而且在 AUC 上的表現差距不多。

Abstract

There are two methods proposed to address classification problems of imbalanced data. First, we propose a method that has smaller parameter space and more performance when using self-training. We train confidence thresholds for each classifier using labeled data to identify high confident data, and label them pseudo labels for re-train. Through this scheme we get less training time for parameters and get better performance. Second, we proposed an efficient training method for imbalanced data. We start with down-sampling and using a method like bootstrap. The model will approximate the model of up-sampling. Using less training data leads to less training time. We do experiments on KDDCUP 2008 data. The result shows that our threshold-based self-training has better performance and the approximated model has the same performance as up-sampling but cost only 0.75 times training time of up-sampling.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 背景及動機

這篇論文主要解決的問題有兩個，一是在自我訓練(self-training)中設定門檻(threshold)的分析，利用它得到更多的改善；二是在不影響效能的情況下，降低處理資料不平衡(imbalance data)問題的訓練時間(training time)。

自我訓練是 semi-supervised learning 的一種方法。semi-supervised learning 是同時利用到了未標記的(unlabeled)和已標記的(labeled)資料來分類的學習方法。較 supervised learning 多利用了為標記資料的資訊，直覺上 semi-supervised learning 的表現會比較好，事實上相配的模型假設和問題結構也是必須的。

假設現在有一個二維分類問題，我們要分類紅色和藍色如 figure 1.。

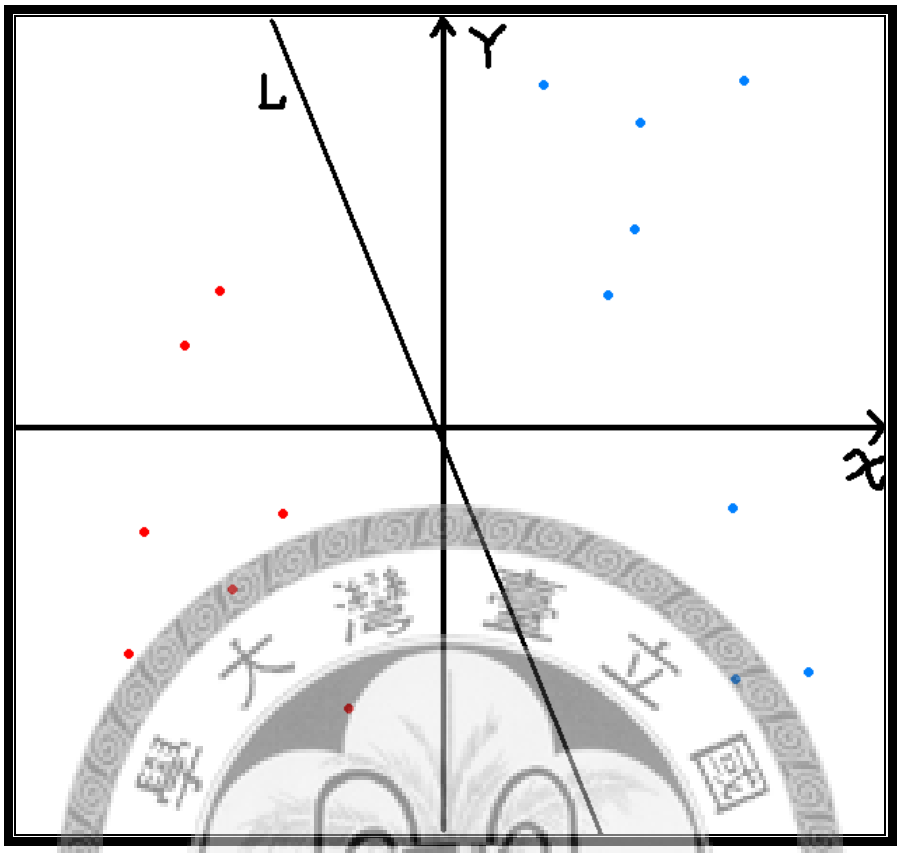

Figure 1 supervised learning 的一個例子，L 為用來決定類別的邊界

以標記為藍色和紅色的資料大概可以直線 L 來分開兩類，不過在 figure 2.考慮了未標記資料後，將判斷邊界設在 Y 軸上更為合理。

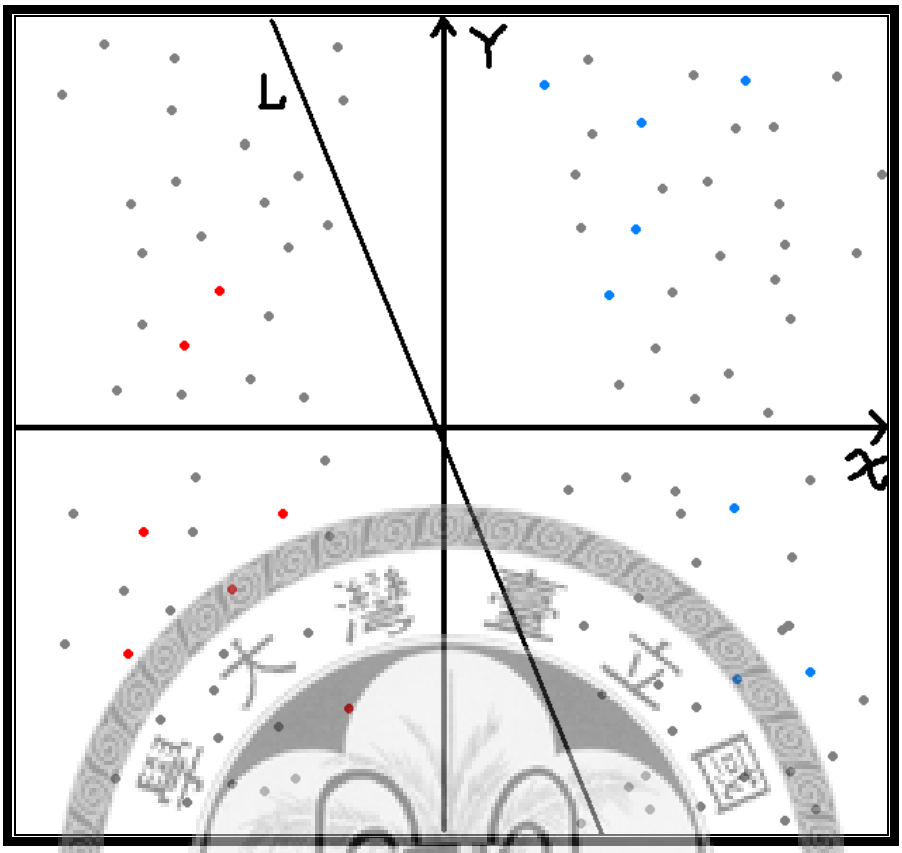

Figure 2 semi-supervised learning 的一個例子，考慮未標記資料後，判斷邊界設在 Y 軸上更為合理

自我訓練可能是最早被提出的 semi-supervised learning 方法，簡單地說就是一個監督式分類器(supervised classifier)不斷地教自己。大部份的形式如，分類器 C 在分類未標記資料 U 時，發現 U 中有一些範例可以用來使自己分類得更好，這些範例便被加上虛擬標記(pseudo-label)，然後跟已標記資料一起作為 C 的訓練資料。

後來發展而成的 ensemble-driven self-training of multiple classifiers[1]裡為基本自我訓練的骨架加上了類似 bootstrap 的方法，從未標記資料中隨機選取的子集合 U'，用以代替 U 在每個迴圈中被分類，然後選出範例加入已標記資料，並再從 U 和 U'的差集合中選取資料填滿 U'使之大小固定。除此之外選出範例的準則則是分

類信心最高的前 n_j 個未標記資料，n_j 則和在已標記資料中觀察到的 j 類與其他比例成正比—在標記資料中出現越多的類別，其 n_j 也越大。這個方法也將停止條件設為經過 k 個迴圈。至此已經有三個互相有關係的參數需要調整，分別為 U'的大小 |U'|、n_j 和 k，在調參數時必須在一個三維空間搜尋，難以最佳化；同時它以 multiple classifier system(MCS)合併多個分類器後輸出的結果來做自我訓練，這可能導致無法使得其中各個分類器的增益彼此。

由於這個方法的三個參數同時都會影響到最後會標記多少虛擬標記資料，舉例來說，如果在|U'|=1~100，設類別 j 和其他類的比例是 1:4，那 nj = 1~1/5*|U'|，k 則是自然數，這三維的空間複雜度為 100/5*k+99/5*k+98/5*k+……+1/5*k = (|U'|+1)*|U'|/2*k/5，搜尋好的參數需要耗費很多時間在測試它們。因為 MCS 的合併方式不同，有時候少數分類器分得好的地方會被其他多數分類器所影響，最後輸出的結果反而分不好，當每個分類器表現越好越明顯。在處理不平衡資料時，由於合併的方法是將各分類器輸出的預測信心值轉換成排名後平均，再計算 Area Under FROC Curve(AUC)。這時可能會有一筆資料兩個分類器將它排名得低，但另一個排得高的情形。轉換為排名後平均很可能使它因為大部份分類器的排名而無法使其他分類器得到這個資訊。

我們認為與其要決定做幾次迴圈、每次迴圈要標注多少虛擬標記，不如直接以分類器輸出的預測信心值做為門檻，在門檻以上的未標記的資料標注上虛擬標記。假設門檻以上的資料都是足以標注虛擬標記，並加入已標記資料重新訓練。這時迴圈的停止條件就是當未標記資料中沒有任何一筆預測信心值大於門檻時。以此為假設則搜尋參數的空間只剩一維，比較容易選擇。同時之前的方法因為只

對整個 MCS 做自我訓練，可能無法得到 MCS 中各分類器的長處。我們提出對 MCS 中每個分類器都調整一個預測信心值門檻，並使它們認為的高信心資料都標記虛擬標記加入訓練資料。

這篇論文要解決的第二個問題是在不影響效能的情況下，降低處理資料不平衡問題的訓練時間。所謂不平衡資料就是各類別的數量不平均的資料。一般評估分類問題最常用的是準確率，然而對一個資料中有兩類比例為 1:9 來說，因為只要猜測數量多的那個類別就有 0.9 的機率答對，分類器會偏向猜測數量多的類別， 對稀有的類別無法分得很好。這通常有兩種取樣方法去處理，up-sampling 和 down-sampling。這兩種取樣都是為了平衡資料中類別的稀有性。Up-sampling 是將稀少類別的資料複製得跟大量存在的類別一樣，相對地 down-sampling 是從具眾多資料的類別中隨機取出符合稀有類別數量的子集合，資料經過取樣後，類別之間的平衡就消除了，這時再開始分類的工作。

Down-sampling 雖然因為使用的資料少而能訓練得很快，但也因此而表現不如 up-sampling，而 up-sampling 則是因為使用的資料多，訓練得比較慢。這篇論文提出一個方法使用在不平衡資料的分類時，同時具有 down-sampling 使用資料少而快，以及 up-sampling 的良好表現兩種優點。主要的想法是從 down-sampling 開始，用類似 boosting 的方法使分類模型逼近為 up-sampling。一開始使用 down-sampling 來分類已標記資料，並不斷將分類錯誤的部份加入 down-sampling 裡來使得模型逼近 up-sampling，直到分類錯誤小於某一個既定的值（如 1%）為止。

我們以 KDDCUP08 的資料來做實驗。以門檻為基礎的自我訓練，我們發現在自

我訓練中使用預測信心值門檻，AUC 比 supervised learning 高 2.4%，比 ensemble-driven self-training of multiple classifiers 好 0.5%。

而以 down-sampling 逼近 up-sampling，可以擁有 up-sampling 1.3 倍的訓練速度，同時幾乎與 up-sampling 的效能一樣。

Chapter 2

Related Works

2.1 Semi-supervised learning

Semi-supervised learning and classification 就是指利用了未標記資料的學習方法，其中基本的做法大致可以分成 generative model、self-training、multiview learning、avoiding changes in dense regions 和 graph based model[2]。

Generative model 假設 $p(x,y) = \sum p(y)p(x|y)$，利用各種方法去估計 p(y)和 p(x|y)，然後就可以得到 joint probability p(x,y)。self-training 是將一個分類器輸出的結果加上了虛擬標記，再加入已標記資料重新訓練，直到所有未標記資料都有標記。Multiview learning 是以不同的觀點來看資料，以這些觀點分別建立分類器後再互相教另一個，例如分類網頁時可以將網頁的文字和圖片分開建立分類器，然後將高信心的未標記資料加上虛擬標記加入另一個分類器的已標記資料。avoiding changes in dense regions，顧名思義是假設資料密集的區域，標記變化會不大，其中一個方法是 transductive SVM，除了跟傳統的 SVM 一樣要分開已標記資料之外，它還尋找可以把未標記資料分得最開的 hyper plane。Graph based model 是將資料

以圖來表示，節點是資料，邊是資料之間的相似度，最基本的分類法是取出類別間的最小切集，然後就可以將未標記資料分類成跟它連在一起的已標記資料一樣。

本篇處理的範圍屬於 self-training 裡利用了 multiple classifier system 的地方。在此已經有人提出一個以 multiple classifier system 為基礎自我訓練的方法 ensemble-driven self-training of multiple classifiers [1]。在前面已經提過，因為需要調整過多參數，使用在大資料時由於訓練時間過長，導致十分沒有效率。

在自我訓練中標注虛擬標記時，不參照已標記資料中的類別分布的作法，在之前也有被提出。作者在《A New Data Selection Principle for Semi-Supervised Incremental Learning》中指出，基於各種引入未標記資料反而降低效能的實驗，對分類有幫助的未標記資料可能不是預測信心值最高的那群[5]。作者提出的方法是先將所有已標記和未標記資料一起分群，然後對各群中的未標記資料依預測信心值排序成數區段。在自我訓練時，就以這些區段分別加上虛擬標記，測試加入哪些區段到已標記資料中對效能比較好。他估計自我訓練中的效能，是以他自己提出的 pseudo-accuracy 和 energy regularization 兩者加上比重相加而成。由於 pseudo-accuracy 是以準確率為基礎的估計，因此明顯不適合本篇論文的主題不平衡資料。

Up-sampling 和 down-sampling 主要是用來解決 imbalance data 中各 class 數量分布不平均的問題，一般來說，up-sampling 的效果會比 down-sampling 好，因為它使用了 data 中所有的資訊，但它的訓練時間也會比較長因為使用了所有的資料。我們折衷使用的方法介於兩者之間，希望可以訓練時間又短，效能又好。

2.2 不平衡資料的訓練

不平衡資料就是各類別的數量不平均的資料。類別不平衡會造成以下幾種問題：(一)不適合的評估標準，(二) 絕對稀有造成的資料短少，(三)相對的資料短少造成的相對稀有，(四)資料破碎，(五)不適合的歸納性偏見，(六)雜訊[2]。

以下分別簡單說明上述幾種問題：

(一) 不適合的評估標準：一般評估分類問題最常用的量度是準確率，然而對一個類別比例為 class 1:class 2 = 1:9 的資料來說，分類器只需要猜測所有資料都是 class 2，而不需要做任何事就可以得到 90%的準確率。另一方面，改進預測 class 1 的能力，反映在準確率上也只得改進預測 class 2 的九分之一效率。為了使評估不管對稀少或眾多的類別都公平，一般使用的量度可能是接收器運作指標曲線(ROC curve, Receiver Operating Characteristic Curve)或是其下的面積(AUC, Area Under ROC Curve)[3][4]。因為它 X,Y 軸的單位是假陽性比率和真陽性比率，不會偏向數量眾多的類別。

(二) 絕對稀有造成的資料短少：因為有些資料取得不易，不管是稀少或是眾多的類別絕對上都很稀少，造成機器難以發現資料分布的模式。由於資料的稀少，原本一種概念可能會因此分裂為幾種包含很少資料的概念，而且對它訓練發現訓練出來的分類器有很高的錯誤率。這方面已經有一個專門研究這種被稱為 small disjuncts 問題的領域。

(三) 相對的資料短少造成的相對稀有：與(二)不同的是這裡的稀少指相對而言，某

種類別的資料變得像大海撈針，難以搜尋。貪婪經驗法則的搜尋，會因為稀有物件需要綜合許多情況才能偵測而難處理。許多研究用基因演算法或是決策樹之類的非貪婪經驗法則的方法去解決這個問題。

(四) 資料破碎：許多資料探勘方法採取各個擊破(divide-and-conquer)。但在不平衡資料中資料分布的常規通常只存在其中某個資料量稀少的分割部份裡。這裡造成的問題即跟(二)一樣。

(五) 不適合的歸納性偏見：一般化(generalizing)或是歸納需要額外事實的傾向。一般化也會被稀少類別所影響。考慮使用 maximum-generality bias[6]時，由於它會最大地去一般化訓練資料，所以資料量少的稀少類別會被一般化得太過火導致效果很差。簡單的改善方法就是偵測資料是否存在一個小的可分類區域內(disjunct)，如果是表示它是稀有的類別，就以 instance-based classifier 分類它，反之則用普通的一般化分類器即可。

(六) 雜訊：因為稀有類別的資料已經夠少了，如果它們被雜訊影響的話，分類結果可能相差很大。一種 down-sampling 方式被提出[7]，它將一些多餘的資料，或是厚重區域中的資料視為雜訊來處理。

文獻中大部份的方法針對舊的處理不平衡資料方法加以改進表現，對於加快訓練速度的部份較少人提出。本篇論文提出的方法主要關於使用比較少資料達到跟 up-sampling 一樣的效果，同時也加快了訓練速度。

Chapter 3

Methodology

3.1 自我訓練的門檻分析

3.1.1. 問題定義

給定一個不平衡資料 D，以及 supervised classifier $C_1, C_2, \ldots, C_N$，為每個分類器分別訓練門檻，以解決先前方案的兩個問題：難以調整彼此相關的參數，以及無法利用各分類器改進彼此。

3.1.2. 先前的解決方案

在 ensemble-driven self-training of multiple classifiers [3]中，作者使用一個 multiple classifier system(MCS)讓它 self-training k 個迴圈。在自我訓練的每個迴圈中，從未標記資料裡隨機取出一個子集合 U'，用 MCS 分類 U'，然後根據已標記資料中各類別的比例來標記這子集合的虛擬類別。將標記了虛擬類別的未標記資料加入已標記資料後，再從子集合以外的未標記資料隨機選取跟虛擬標記資料等量的資料

來填補 U’，使得它維持一定的大小。

這個方法需要調整的參數包括：U’的大小|U’|、迴圈數 k 以及每次需要標記虛擬標記的數量 nj。其實以上三個參數之間都有關係，k 以及 nj 互相影響了標記虛擬標記的數量，k 越大而 nj 越小代表了對分類結果比較不信任，因此每次只取預測信心值最前面的幾筆資料；nj 越大而 k 越小則反之。|U’|則對分類結果有所影響，小的|U’|過了幾次的迴圈，運氣不好時可能只剩一類資料。由於這三個資料會彼此影響，因此在調整參數時等於在一個很大的三維空間搜尋，以 KDD cup 08 的資料為例，未標記資料有 94730 筆，|U’|的範圍就落在 1 到 94730 筆；已標記資料中顯示 positive：negative = 1:164，與類別比例成正比的 nj 從 1 到 1/164*|U’|都有可能；而迴圈數 k 則是從 1 開始的自然數。|U’|從 94730 到 1，對應到 nj 的範圍是 1~|U’|/165，再乘上 k 就得到下式。參數的空間是 94730/165*k+94729/165*k+94728/165*k+……+1/165*k = k/165*sum(1~94730) = k/165*94730*94729/2。

此外這個方法是對整個 MCS 做自我訓練，也導致無法使各個分類器改進彼此。舉例在 KDD cup 08 中，adaboost 表示某點高度可能患病，然而 linear SVM 和 RBF SVM 不這麼認為，在處理不平衡資料時通常使用的評估是 AUC，經過 MCS 合併排名後，它可能因為兩個分類器不表贊同而落居人後，但這不表示它不該被當作患病部位而被加入訓練資料中，尤其是以 KDD cup 08 中 supervised learning 即可以有好表現時。

除了需要調整的參數太多以及無法使各個分類器改進彼此的問題，在類別不平衡時，一個類別佔了大部分而另一個類別的資料很少，量少的類別比較能影響分類的結果，但是測試時選出的子集合中包含的 class 比例其實是不穩定的，這導致

結果不太理想。在我們的實驗中，當子集合等於整個未標記資料時，表現會比較好。

3.1.3. 提出的解決方案

針對參數搜尋空間過大的問題。我們提出為分類器選擇一個預測信心值門檻，用來區別它分類的資料是否應該加入訓練資料，當沒有資料在此門檻之上時，就輸出現有的模型來分類所有的未標記資料。這同時整合了之前需要調整自我訓練停止條件的 k，以及每個迴圈需要加入多少資料的 nj，至於|U'|之前已經提過由於不平衡的資料，設為所有未標記資料的大小比較適當。以上的三個參數合成了一個預測信心值門檻。

而 MCS 中的分類器無法增益彼此的問題。我們合併前一個問題的解決方法為 MCS 中的每個分類器調整預測信心值門檻，並將它們選出的高信心的資料以聯集方式加入訓練資料。使用聯集方式來合併比投票或是交集合理，因為聯集同時可以得到各分類器的長處。由於這些長處已在已標記資料中用 cross-validation 來驗證是否對 MCS 的效能有增益，可以相信聯集所有分類器的意見是最好的。如 figure 3. 就是整體的過程。

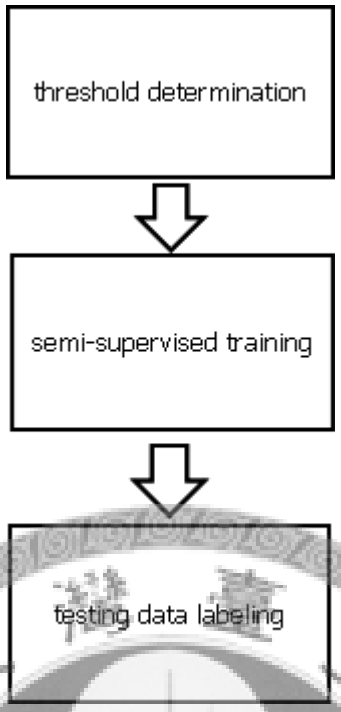

Figure 3 overall flowchart

合理地決定預測信心值門檻是必要的，因為它們是增進效能的關鍵。基本的想法是透過 cross-validation 觀察各個預測信心值門檻在已標記資料上的效能之後，選出表現好的應用在未標記資料上。實際的作法是利用 greedy 演算法分別訓練 C_{TH1} 至 C_{THn}。如 figure 2.訓練 C_{TH1} 時，先在 C_{TH1} 可能出現的範圍取樣一些值當作候選人，這範圍可由 cross-validation 得到。然後對每個候選人都做 cross-validation 如 figure 2.來測試它對 MCS 是否有幫助，比較各候選人的表現後，擇優選取用在 semi-supervised training 的階段。

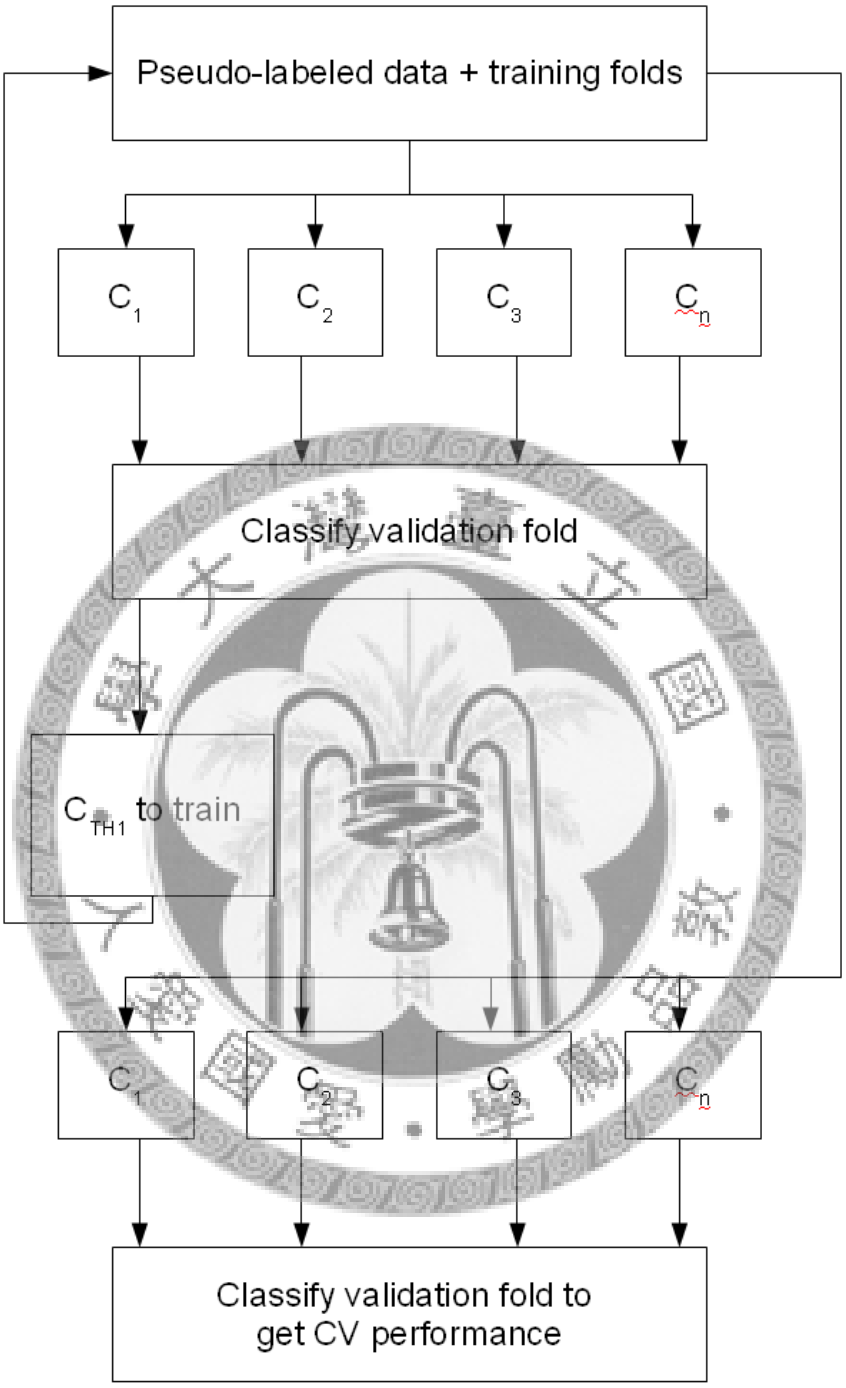

Figure 4 an example of threshold determination, training C_{TH1}

Training confidence threshold C_{THn} algorithm
Given:
Training folds L
Validation fold U
Supervised classifier C_1, C_2, …...C_n
Confidence threshold for training C_{THn}

Loop
Train C_1, C_2, …… C_n with L.
Classify U with C_1, C_2, …...C_n to get confidence cf_1, cf_2,……cf_n.
Take highly confident unlabeled data H_n from cfn according to C_{THn}
Move H_n from U to L.
Until H_n is null
Use C_1,C_2……C_n to get result.

Figure 4.中詳細的演算法如以上的 Training confidence threshold C_{THn} algorithm，對 cross-validation 中的一個測試來說，有一個 fold 用來 validation，其他 fold 用來訓練， C_n 使用 C_{THn} 為高信心的資料加上虛擬標記加入訓練資料 L，直到 U 中沒有資料的預測信心值高於 C_{THn} 為止。這時再使用已用虛擬標記資料和 L 訓練好的 MCS 來得到這個 fold 的表現。

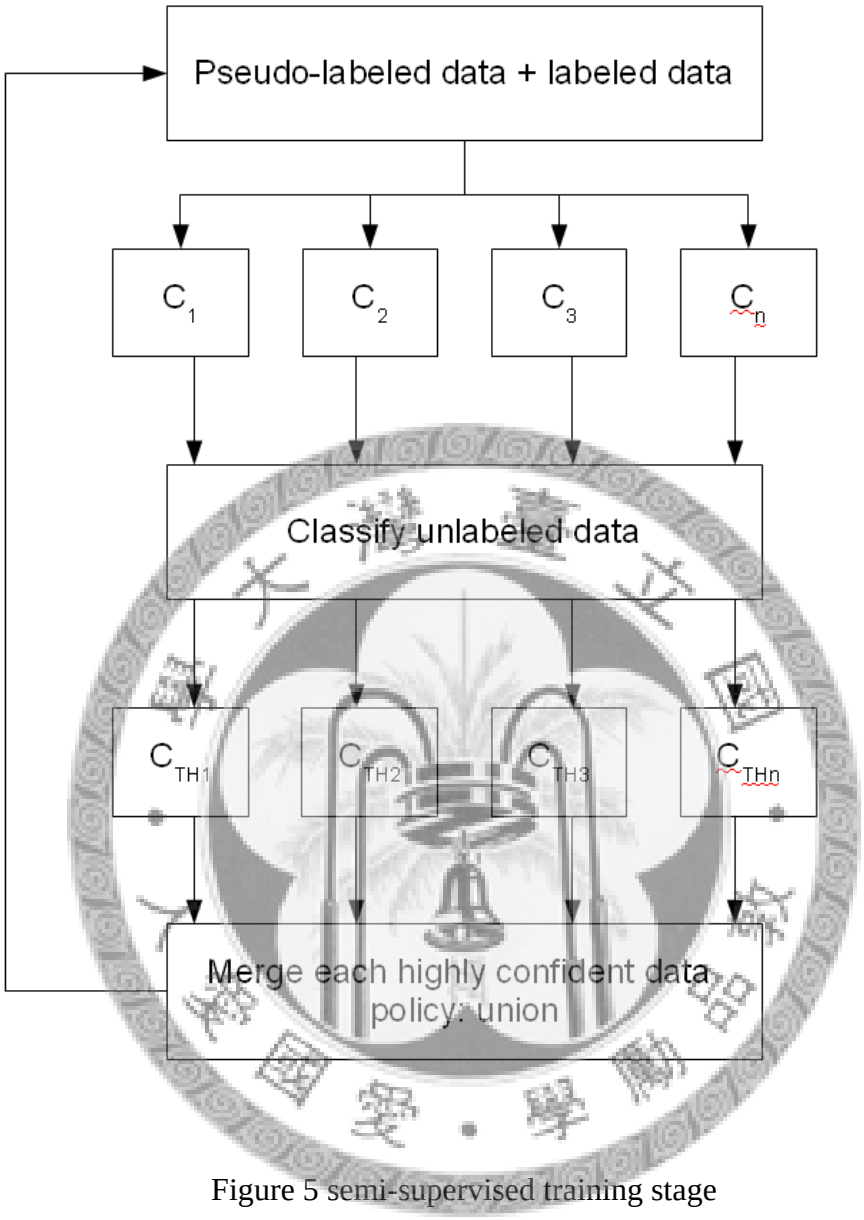

Figure 5 semi-supervised training stage

使用 cross-validation 決定好各分類器的預測信心值門檻後，將它們使用在 semi-supervised training stage，以得到自我訓練的模型，如 figure 5。詳細的過程請見 Threshold-based self-training algorithm。

Threshold-based self-training algorithm
Given:
labeled data L
Unlabeled data U
Supervised classifier C_1, C_2, …...C_n
Confidence threshold C_{TH1}, C_{TH2}, …...C_{THn}

Loop
Train C_1, C_2, …… C_n with L.
Classify U with C_1, C_2, …...C_n to get confidence cf_1, cf_2,……cf_n.
Take highly confident unlabeled data H_1,H_2… from cf_1, cf_2…. according to each C_{TH}
Merge H_1, H_2,…… to produce P.
Move P from U to L.
Until P is null

較 ensemble-driven self-training of multiple classifiers 細膩的地方在於，標記虛擬標記的準則變成為每個分類器量身設計，並且經過 cross-validation 的驗證。每個迴圈都把各分類器經預測信心門檻的高信心資料放入已標記資料中，直到沒有資料可放為止，停止時的 C_1,C_2……C_n 就是組成最後 MCS 的分類器。在 testing data labeling 使用此 MCS 就可以得到自我訓練的結果。

3.2 對不平衡的資料有效率地訓練

3.2.1. 問題定義

給定不平衡資料 D，以及 supervised classifier C，C 的訓練時間太長時，針對它縮短訓練時間，但效能不能下降太多。

3.2.2. 提出的解決方案

在實驗中發現，當資料很大時，自我訓練門檻分析中的 supervised classifier 可能會有執行時間太長的問題。在此提出了一個可以用來提升處理不平衡資料的分類器速度的演算法。

處理不平衡資料一般有 down-sampling，又稱 under-sampling 和 up-sampling，又稱 over-sampling 兩種方法來平衡各類別的數量。為了分類效能，分類器通常會選擇使用 up-sampling，但是如此會產生效率不彰的問題。在此提升訓練速度的基本想法是，盡量減少使用的資料數量，就可以達到加速的目的。不過同時間必須要滿足有限的效能下降。我們提出一種類似簡化的 boosting，以 down-sampling 為基礎的方法。

Algorithm. Approximate up-sampling from down-sampling
Given:
    Positive class P
    Negative class N
    K

Take N' randomly from N, such that |N'|=|P|.
Train classifier C with P and N'
For k iteration
    Classify N-N' with C.
    If false positive < 1%
        Break;
    end
    Move false positive to N'.
    Use P and N' to build classifier C.
End for
Output C as final classifier for testing data.

Approximate up-sampling from down-sampling algorithm.描述了整個方法。一開始 down-sampling 利用平衡的資料建立一個分類器 C，接下來的迴圈裡，以 C 來分類 N 和 N'的差集合，並將分錯的部分加入訓練資料再建立分類器 C。經過迴圈之後還沒被選進訓練資料裡的已標記資料就被當成是多餘的資料，最後就以 P 和 N'所訓練的分類器來分類未標記資料。例如現在資料中有 positive(P)和 negative(N) 兩種 class，P 類的資料比 N 類少很多。一開始先用 down sampling，也就是從 N 類中隨機選出跟 P 類一樣多的資料，稱為 N'，然後用 P 加上 N'建立分類器。為了了解隨機選出的這些資料是否足以解決問題，也就等於在問：是否還有必要的資訊存在剩下沒挑到的資料(N-N')中?我們提出將 N-N'當作 held out data 來測試由 P 和 N'建立的 classifier。如果 N-N'中有資料被分錯為 P 類，那麼就將它們加入 N'中，以加強分類的效能。重覆數次之後，這時 N 和 N'的差集合加上 P 類建成的 classifier 就足以跟使用全部資料的分類器有一樣的效能。

Chapter 4

Experiment and Result

4.1 實驗資料

在實驗中面對的是 Data Mining and Knowledge Discovery competition(KDD cup) 2008 的 task 1，它是一個不平衡分類問題。它也和 multi-instance learning(MIL)問題很類似。MIL 的訓練資料由 labeled bags 組成，每個 bag 裡有許多 unlabeled instance。只要一個 bag 裡有一個 positive instance，那它被標記為 positive；反之 bag 裡都是 negative 資料時，它就是 negative。在 KDD cup 2008 中，每個病人就像上述的 bag，bag 中有許多從 X 光片取得的特徵向量。不過不同的地方在於 bag 中的 instance 是有 labeled 的。

目標是建立 computer-aided detection system 分辨每個病人是否罹患乳癌。特徵向量由許多乳癌病患的 X 光片上取樣的點組成，共有 117 維，不過我們無法知道 117 維各自代表的意義。每個病患擁有四種 X 光片：左乳和右乳，以及不同的照片拍攝方式，分別叫 MLO 和 CC。資料中還提供了樣本在 X 光片中的位置。不平衡的地方在於患病的病患和健康的人的比例是 118:1712，以特徵向量的比例來看更為

不均，623:101662。

4.2 評估方式

分類問題一般量測效能的方式是準確率。但對不平衡資料分類時，只要光猜測全部的測試資料為 negative 類，就可以得到 95%以上的準確率。這顯然不合理，因為以醫生的立場來說，找出患病的病人比健康的病人重要多了。

用 free-receiver operating characteristic(FROC) curve 來測量 computer-aided detection(CAD) system 的效能已行之多年[8]。它反應了分類結果中 false alarm 和 sensitivity 之間的取捨。這是一個 2 維的曲線，當我們把 decision boundary 從 confidence value 最高移到最低，以 X 軸代表特徵向量的 false alarm rate，Y 軸是找出患病患者的比例。我們就可以畫出如 figure 10.這個曲線。

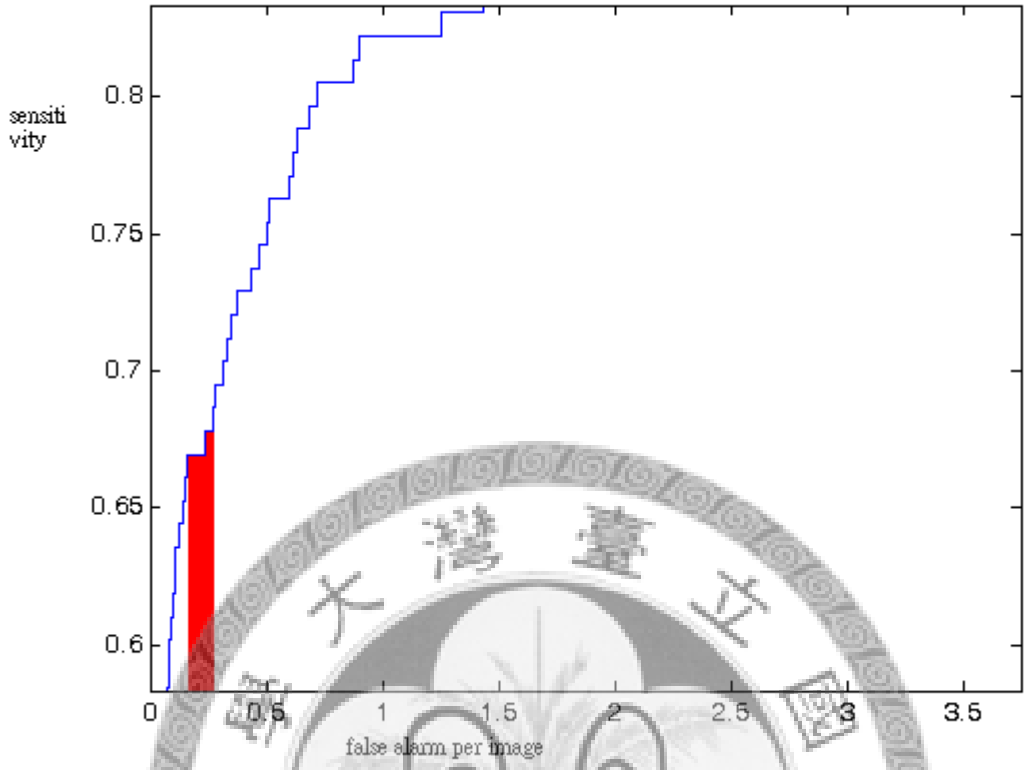

Figure 6 an example of area under FROC curve

為了不要有太多 false alarm，我們跟 KDD cup 一樣取 0.2 到 0.3 之間的 area under FROC curve(AUC)來當作評估標準如 figure 6.所示的紅色部份，最高是 0.01。為了容易了解，以下我們將會用百分比來表示 AUC。

4.3 Confidence threshold exploitation in self-training of MCS

4.3.1. Baseline 我們使用 KDD cup 2008 winner 的三種 classifier 組合而成的結果做為 baseline。[11]三種 classifier 是 adaboost, RBF SVM 和 linear SVM，使用的工具

分別是 MDV adaboot, weighted libsvm[9]和 liblinear[10]。一般的 up-sampling 是將 positive 類(患病)複製得跟 negative 類(健康)一樣多，不過 KDD cup 2008 winner 將每個患病病人中的比重加重，並把每個患者得到的比重平均分給他的 positive feature。舉例來說，KDD cup 2008 的訓練資料有 118 個 positive 病人和 101662 個 negative feature，為了平衡 class，他們把 101662 平均分給 118 個病人，也就是每個患者可以得到約 860 的比重,這 860 的比重再平均分給每個病患中的 positive 特徵向量。舉例來說，患者 A 中有兩個 positive 特徵向量，那麼這兩個 positive 特徵向量在訓練 classifier 時設的比重就是 860/2 = 430。

將三種 classifier 輸出的 confidence value 轉換成降序排名，也就是將最有可能是 negative class 的資料排第一，最有可能是 positive class 的資料排到最後，再將三個排名平均，就得到 baseline 如 table 1.。

classifier	adaboost	linear svm	RBF svm	merge
AUC	86.70%	87.50%	87.40%	89%

Table 1 performance of 3 base classifiers and merged baseline

4.3.2.實驗　因為 KDD cup 2008 的資料極度不平衡且數量很大,我們的實驗專注在對 positive 類標記虛擬標記。藉著增加稀少的 positive 類的數量，使結果變好。

在自我訓練的門檻分析中需要傳入的參數取決於使用的 supervised classifier 個數。每個 classifier 需要一個 confidence threshold，大於預測信心值門檻的資料我們就將它標記為 positive。為了決定在未標記資料上各個分類器的預測信心值門檻設定哪些值，我們首先在預測信心值大於 0 以上的範圍取樣出幾個點，並在訓練資

料上 cross-validation 來測試預測信心值門檻設在哪裡對結果最好。

4.3.3.結果　ensemble-driven self training of multiple classifiers 結果如 table 3.：

|U'|	30000	60000	94730(all)
AUC	90.40%	90.60%	90.99%

Table 2 performance of ensemble-driven self training of multiple classifiers for each |U'|

對每種|U'|我們都在已標記資料上 cross-validation，用以選出較好的 k，至於 nj 則是等於已標記資料中的比例乘上未標記資料的數目 1*623/102294*|U'|，至於 MCS 中的合併方式，跟 baseline 一樣是轉為排名後再平均。

U'是每個迴圈中要標記虛擬標記的部份未標記資料。由於是隨機選取，稀少的 positive 類不保證在 U'中會出現多少。我們直接將 U'設為全部的未標記資料的確得到了比較好的結果。

lsvm threshold	1.4114	2.4114	3.4114	4.4114	5.4114	6.4114	7.4114	8.4114
AUC	79%	78.80%	78.10%	78.10%	78.10%	78.10%	78%	78%

Table 3 cross-validation over labeled data　for choosing confidence threshold of linear SVM

wsvm threshold	1.7458	2.7458	3.7458	4.7458	5.7458	6.7458	7.7458
AUC	78.30%	78.10%	77.70%	78.10%	78.70%	78.70%	78.80%

Table 4 cross-validation over labeled data for choosing confidence threshold of RBF SVM

mdv ada threshold	2.1943	3.1943	4.1943	5.1943	6.1943	7.1943	8.1943
AUC	78.40%	78.70%	78.70%	78.70%	78.90%	78.90%	78.80%

Table 5 cross-validation over labeled data for choosing confidence threshold of mdv ada boost

自我訓練的門檻分析實驗中，各個 classifier 在 training data 上做 cross-validation，然後根據 cross-validation 結果的預測信心值來取樣出數個大於 0 的點來分別測試。以上的 table 3,4,5 分別代表三個 classifier 在選取預測信心值門檻時 cross-validation 的實驗。最後我們選擇 AUC 分別定出 linear svm、weighted svm 和 mdv adaboost 的 confidence threshold 分別為 1.4114, 7.7458, 6.1943。

決定了預測信心值門檻之後，把它們套用進測試資料，可得到最終的結果如 table 6。

merge policy	inter	union	vote
AUC	89%	91.40%	88.40%

Table 6 performance of algorithm 1 using 3 merge policies

Merge policy 有取交集(inter)、聯集(union)和投票(vote)來做比較。較 ensemble-driven self training of multiple classifiers 進步了 0.5%左右，比 baseline 進步了 2.4%。因為交集方式是三個分類器都同意後才能標注虛擬標記，等於學到原本 ensemble 就能學到的東西，因此預期是表現最好的是聯集的合併方式，可以使 MCS 同時學到三種分類器的長處。

4.4 Approximate up-sampling from down-sampling

4.4.1. Baseline table 7.記錄了三種分類器用已標記資料訓練所花的時間。執行最慢的是 RBF SVM。相對 linear SVM 的數秒鐘、adaboost 的十數分鐘，RBF SVM 每次訓練需要花上一小時以上顯然成為需要改進的瓶頸。改進 RBF SVM 實驗的 baseline 就是 table 7.中的 7508.8 秒。

classifier	RBF SVM	linear SVM	adaboost
training time(sec)	7508.8	3	845.9

Table 7 training time of 3 base classifiers

4.4.2.實驗 由 table 7.可知，花費時間最長的是 RBF SVM，每次訓練要花一小時以上。所以 algorithm Approximate up-sampling from down-sampling.的實驗主要測試可以改進它多快的速度，以及降低了多少分類效能。為 algorithm Approximate up-sampling from down-sampling.做的實驗，我們必須決定在 testing data 上，加入 false positive 的次數，也就是 iteration k。跟之前一樣，我們在 training data 上做 cross-validation，測試 k 設為各種值時，RBF SVM 的分類效能和執行時間表現如何。再將決定好的 k 用在 testing data 上。

4.4.2. 結果 對 RBF SVM 的加速實驗，為了決定 iteration k 應該要設為多少，先在十萬筆 labeled data 上用 10 fold cross-validation 測試，用九萬筆資料當作訓練資料，一萬筆當作測試資料。請見 table 8.：

k	1	2	3	4	total data
AUC	69%	73.70%	75.20%	75.80%	76%
avg time(second)	1.4	969.5	4173.8	6969	5364
avg used data	1120.4	29892.7	30971.5	30390.6	92064.6

Table 8 labeled data cross validation for choosing k

可見 k=3 時，AUC 跟使用所有資料時相差不遠，但是每 fold 平均執行時間縮短了。將 k=3 套用進完全十萬筆已標記資料和九萬筆未標記資料的實驗，可得到下列結果：

k	3	total data
AUC	87.10%	87.40%
time(second)	5705.7	7508.8
used data	36930	102294

Table 9 result of RBF SVM using up-sampling and proposed method

k=3 時，跟使用全部資料的表現都相差不到 0.5%，觀察分類器輸出的預測信心值可知，雖然加速後的預測信心值排序會改變，導致表現不會跟使用全部資料一樣，但是以病人來排序的話，排名跟使用全部資料時一樣。這說明了為什麼雖然表現不一樣，但是差異非常地小。加速後的執行速度是使用全部資料的 1.3 倍，而且減少訓練資料對病人排名改變的很小甚至沒有改變，表現不會降低很多。

在實驗中，為了逼近成跟完整資料時一樣，k=1 開始我們就為它加上了跟使用全部資料時一樣的 patient-based up-sampling，negative point:positive patient = 1:860，因此在 down-sampling 時的 AUC 不高，不過後來漸漸逼近使用全部資料的 AUC。

Approximate up-sampling from down-sampling 用在訓練時間最高的 RBF SVM 效果十分好，我們也將它套用在 AdaBoost 上。接下來 table 10,11 是套用進 adaboost

的表現。

k	1	2	3	4	5	use all data
AUC	74.95%	75.80%	76.90%	77%	77%	80%
avg time	8.1	103	198.8	289.6	386.8	757
avg used data	1120.4	14233.6	14235.7	14236	14236	92064.6

Table 10 labeled data trained by adaboost cross validation for choosing k

k	4	all data
auc	79.70%	86.70%
time(sec)	231	845.863
used data	14290	102294

Table 11 result of adaboost using up-sampling and proposed method

Adaoost 不管在 cross-validation 或是測試階段都和使用所有資料時有明顯的誤差，這應該是因為它不像 SVM 系列用許多參數來表達模型，而是使用資料中的錯誤來修正，當訓練資料數目少時就無法得到很好的效果，導致逼近的效果不好。

Chapter 5

Conclusion

我們在論文中提出了兩種方法，其一以在訓練資料上自我訓練的效能為基礎選擇預測信心值門檻，並在 MCS 中使用聯集方式來合併各分類器意見的自我訓練方法，將參數搜尋空間降為一維使得可以平行地為每個分類器選擇門檻，並且得到稍好的 AUC。簡化參數調整使得在處理不同的資料和問題時，花費較少的時間在訓練資料上驗證參數，能快速得到好的表現。

此外我們還提出了在不平衡資料上以 down-sampling 為基礎，將模型逼近 up-sampling 的方法。實驗上得到了 1.3 倍於使用完整資料的速度，並且以病人為主的排序改變得很少甚至沒有，AUC 與它一樣。加速 up-sampling 除了能加快不平衡資料的訓練之外，代入自我訓練這種需要不斷重新訓練的方法，也能加快速度，效能並不會降低。

Bibliography

[1] Luca Didaci, Fabio Roli: Using Co-training and Self-training in Semi-supervised Multiple Classifier Systems. SSPR/SPR 2006: 522-530

[2] G. M. Weiss. Mining with rarity - problems and solutions: A unifying framework. *SIGKDD Explorations*, 6(1):7–19, 2004

[3] A. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7): 1145-1159, 1997.

[4] F. Provost, and T. Fawcett. Robust classification for imprecise environments. Machine Learning, 42: 203-231, 2001.

[5] Rong Zhang, Alexander I. Rudnicky, "A New Data Selection Principle for Semi-Supervised Incremental Learning," *Pattern Recognition, International Conference on*, vol. 2, pp. 780-783, 18th International Conference on Pattern Recognition (ICPR'06) Volume 2, 2006.

[6] R. C. Holte, L. E. Acker, and B. W. Porter. Concept learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 813-818, 1989.

[7] M. Kubat, and S. Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In Proceedings of the Fourteenth International Conference

on Machine Learning, pages 179-186, Morgan Kaufmann, 1997.

[8] R. G. Swensson, “Unified measurement of observer performance in detecting and localizing target objects on images,” Med. Phys. **23**, 1709–1725 s1996d.

[9] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research 9(2008), 1871-1874. Software available at http://www.csie.ntu.edu.tw/~cjlin/liblinear

[11] Hung-Yi Lo, Chun-Min Chang, Tsung-Hsien Chiang, Cho-Yi Hsiao, Anta Huang, Tsung-Ting Kuo, Wei-Chi Lai, Ming-Han Yang, Jung-Jung Yeh, Chun-Chao Yen and Shou-De Lin, Learning to Improve Area-Under-FROC for Imbalanced Medical Data Classification Using an Ensemble Method, SIGKDD Explorations, 10(2), pp.43-46, December 2008.