

國 立 中 央 大 學

資訊工程研究所
碩士論文

動態多模型融合分析研究

Dyanamic Ensemble Learning research



研 究 生：蘇俊儒

指 導 教 授：陳 弘 軒 博 士

中 華 民 國 一 百 零 八 年 六 月



國立中央大學圖書館 碩博士論文電子檔授權書

(104 年 5 月最新修正版)

本授權書授權本人撰寫之碩/博士學位論文全文電子檔(不包含紙本、詳備註 1 說明)，在「國立中央大學圖書館博碩士論文系統」。(以下請擇一勾選)

(√)同意 (立即開放)

()同意 (請於西元 _____ 年 _____ 月 _____ 日開放)

()不同意，原因是：_____

在國家圖書館「臺灣博碩士論文知識加值系統」

(√)同意 (立即開放)

()同意 (請於西元 _____ 年 _____ 月 _____ 日開放)

()不同意，原因是：_____

以非專屬、無償授權國立中央大學、台灣聯合大學系統圖書館與國家圖書館，基於推動「資源共享、互惠合作」之理念，於回饋社會與學術研究之目的，得不限地域、時間與次數，以紙本、微縮、光碟及其它各種方法將上列論文收錄、重製、與利用，並得將數位化之上列論文與論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

研究生簽名: 莊俊儒 學號: 106522069

論文名稱: 動態多模型長效力分析研究

指導教授姓名: 陳弘軒

系所 : 資訊工程 所 博士班 碩士班

填單日期: 108/7/2

備註：

1. 本授權書之授權範圍僅限電子檔，紙本論文部分依著作權法第 15 條第 3 款之規定，採推定原則即預設同意圖書館得公開上架閱覽，如您有申請專利或投稿等考量，不同意紙本上架陳列，須另行加填申請書，詳細說明與紙本申請書下載請至本館數位博碩論文網頁。
2. 本授權書請填寫並親筆簽名後，裝訂於各紙本論文封面後之次頁（全文電子檔內之授權書簽名，可用電腦打字代替）。
3. 讀者基於個人非營利性質之線上檢索、閱覽、下載或列印上列論文，應遵守著作權法規定。

國立中央大學碩士班研究生
論文指導教授推薦書

資訊工程 學系/研究所 蔣俊儒 研究生所提之論
文 動態多模型融合分析研究

係由本人指導撰述，同意提付審查。

指導教授 洪弘志 (簽章)

108 年 7 月 2 日

國立中央大學碩士班研究生
論文口試委員審定書

資訊工程 學系/研究所 蘇俊儒 研究生
所提之論文
動態多模型融合分析研究

經本委員會審議，認定符合碩士資格標準。

學位考試委員會召集人

謝炯宇

委

員

謝炯宇

王家慶

孫弘仰

中華民國 108 年 06 月 24 日

動態多模型融合分析研究

摘要

現今多模型的整合大多採用固定策略，在訓練過後，多個基礎模型將以「靜態」的方式作融合，即：不會因為待測樣本的特徵不同而改變基礎模型的融合方式。但在現實的訓練情境中，單一模型可能只擅長於預測特定特徵分佈的樣本。由於各個樣本的特徵分佈不盡相同，只採用「靜態」融合的策略可能是過於天真的。

主流多模型融合大多假設單一基礎模型對不同數據的預測的能力大致相同，本論文想嘗試設計「動態」的融合學習以彌補這個假設可能造成的缺陷。我們已經嘗試了五種不一樣的方法，分別根據 (1) 基礎模型判斷類別的機率；(2) 將基礎模型判斷轉換成損失；(3) 根據樣本空間的判斷能力；(4) 根據樣本空間的答對個數；及 (5) 加入分類器判斷正確屬性，以這五種不同的方法來實做「動態」融合。

本文將說明我們設計的五種方法，並在人工生成資料集、車險資料集、Fahsion-MNIST 以及 Kuzushiji-MNIST 上的實驗結果。我們設計的融合方法的預測準確度均優於基礎模型，這說明動態的多模型融合是可行的。然而，與理想 Model 相比，結果相差甚遠，在訓練額外屬性學習器上還有加強的空間。

關鍵字：多模型融合, 動態多模型融合, 監督式學習

Dyanamic Ensemble Learning research

Abstract

Nowadays most of the ensemble learning methods apply a static strategy to integrate the base learners. After training, base learners are merged in a “static” manner, that is, the basic models will not adapt the fusion strategy to the different feature distribution of the samples to be tested. However, in a realistic training scenario, a single model may only be good at predicting samples of a particular feature distribution. Since the features of each sample are distributed differently, the strategy of using only “static” fusion may be over-naïve.

The mainstream ensemble models mostly assume that the ability of a single base model to predict different data is roughly the same. This paper attempts to design a “dynamic” ensemble model to compensate for the shortcomings of this hypothesis. We have tried five different methods, based on (1) the category probability predicted by the base learners; (2) the loss of the base learners; (3) the percentage of correctness of the nearby samples predicted by the base learners ; (4) the numbers of correctness of the nearby samples predicted by the base learners ; and (5) adding extra features about which base learner correctly predict the right label. These five methods realize the “dynamic” ensemble.

This article will explain the five methods we designed and the experimental results based on a simulated dataset and three real datasets, includ-

ing the Allstate dataset, the Fashion-MNIST dataset, and the Kuzushiji-MNIST dataset. We found that all five ensemble methods perform better than each of the single base learners. However, if we compare our method with an ideal model, the result is not good enough. Therefore, it may still be possible to improve our methods by training the leaner with extra features.

Keywords: ensemble learning, dynamic ensemble learning, supervised learning

誌謝

我們感謝工業技術研究院巨量資訊科技中心 (ITRI 107-W1-21A2) 及
科技部 (MOST 107-2221-E-008- 077-MY3) 對本計畫的補助。

目錄

	頁次
摘要	iv
Abstract	v
誌謝	vii
目錄	viii
圖目錄	x
表目錄	xi
一、 緒論	1
二、 背景及相關論文	3
三、 動態多模型融合法	10
3.1 根據基礎模型判斷類別機率做融合	10
3.2 將基礎模型判斷損失做融合	11
3.3 根據樣本空間的判斷能力做融合	13
3.4 根據樣本空間的答對個數做融合	14
3.5 加入分類器判斷正確屬性作融合	15
四、 實驗結果	18
4.1 資料集介紹	18
4.1.1 人工生成資料集	18
4.1.2 車險資料集	19

4.1.3	Fashion-MNIST 資料集.....	21
4.1.4	Kuzushiji-MNIST 資料集	23
4.2	實驗設定	24
4.3	人工生成資料集結果	25
4.4	車險料集結果	27
4.5	Fashion-MNIST 結果.....	28
4.6	Kuzushiji-MNIST 結果	31
五、	討論與未來發展	33
參考文獻		35
附錄一		37

圖目錄

	頁次
2.1 最近鄰居法概念圖	3
2.2 支援向量機概念圖	4
2.3 決策樹概念圖	5
2.4 隨機森林概念圖	6
2.5 梯度增強演算法概念圖	7
2.6 XGBoost 概念圖 [1]	8
2.7 META-DES 概念圖 [2]	9
3.1 模型 1 資料集分割設計概念圖	10
3.2 模型 2 資料集分割設計概念圖	12
3.3 模型 3 資料集分割設計概念圖	13
3.4 模型 4 資料集分割設計概念圖	15
3.5 KSD 屬性設計概念圖	16
3.6 模型 5 資料集分割設計概念圖	17
4.1 人工生成資料之正樣本及負樣本分布情況	18
4.2 車險資料集原始資料集擷取	20
4.3 車險資料集 Target 的分布狀況	20
4.4 Fashion-MNIST 範例	22
4.5 Kuzushiji-MNIST 範例	23

表 目 錄

	頁次
4.1 Fashion-MNIST 10 個 class 的種類	21
4.2 Kuzushiji-MNIST 10 個 class 的種類	24
4.3 人工生成資料集在三種基礎學習器 (KNN、SVM、Decision tree)、一種典型的融合模型 (Majority Voting)、及三種我們提出的動態融合模型的準確率比較 (單位：%)	26
4.4 車險資料集在基礎學習器 (KNN、SVM、Decision tree)、一種典型的融合模型 (Majority Voting)、及五種我們提出的動態融合模型的準確率比較 (單位：%)	28
4.5 Fashion-MNIST 資料集在基礎學習器 (KNN、SVM、Decision tree、XGBoost)、兩種典型的融合模型 (Majority Voting、XGBoost)、及五種我們提出的動態融合模型的準確率比較 (單位：%)	30
4.6 Kuzushiji-MNIST 資料集在基礎學習器 (KNN、SVM、Decision tree、XGBoost)、兩種典型的融合模型 (Majority Voting、XGBoost)、及五種我們提出的動態融合模型的準確率比較 (單位：%)	32

一、緒論

現今許多有關機器學習 (Machine Learning)、人工智慧 (Artificial Intelligence) 的比賽，或是 Kaggle 上的成果，甚至是 Benchmark 上面的分數，大多都是由多模型融合取得不錯的成績，在現在這種多模型融合流行的時代中，我們發現一個問題，如果把這個問題解決好，或許多模型融合能有更出色的表現。

目前主流的多模型融合 (ensemble learning)，如：XGBoost、Adaboost 等強大的多模型融合大多以「靜態」的方式做結合。所謂「靜態」指的是訓練結束後各個基礎模型的結合方式即定型，例如：融合模型可能學習各個基礎訓練器的權重，而這組權重在訓練結束後，就代表著每個基礎學習器的強弱或重要程度。在最後融合的過程中，僅僅以加權後的結果，代表著所有基礎學習器對這一項任務或問題整合後的預測。但在真實的訓練過程中，有較多的情況顯示，不同的基礎學習器對不一樣的資料樣本 (data sample) 有著不同的判斷準確度。在這樣的狀況下，「靜態」的融合策略似乎過於天真。所以我們認為：採用「動態」的融合方式 (例如：給予各個基礎學習器動態的權重)，可能可以更好地將基礎學習器對不同資料樣本的優劣勢表現出來，例如：對於樣本 1，融合模型根據該樣本的特徵分佈認為應給予基礎模型 a 較高的權重，但對於樣本 2，融合模型卻給予基礎模型 b 較高的權重。這種動態的融合方式與目前常見的多模型融合模型有相當的差異。

在我們的想像中，從學好的基礎分類器中，根據不同的理念多提取資訊，來設計與發想「動態」這個概念是非常重要的，而我們也提出了

五種不一樣的多模型融合法，來改善上述所提及的問題，也希望多模型融合能更出彩、更完美。

接下來在第二章會提及背景與相關論文，第三、四章分別講述實驗模型的原理以及實驗結果，第五章總結目前為止的發現、討論實驗的問題以及未來可能可以繼續專研的方向。

二、背景及相關論文

近期機器學習、人工智慧等名詞，因為電腦的運算速度、記憶體大小等電腦性能優化後，重新引起學者的關注與持續的發展，這兩個名詞已經變成一種口號現在我們生活中的每個角落，而這樣的東西又是甚麼呢？機器學習就是利用統計學模型或是科學研究的演算法，讓電腦可以有效率地完成特定的任務。我們不需要給予電腦明確的指示，而是讓電腦根據資料的樣子去推理。以下會介紹幾個常見且簡單的機器學習演算法，而這幾個演算法也會在實驗中當成基礎學習器。

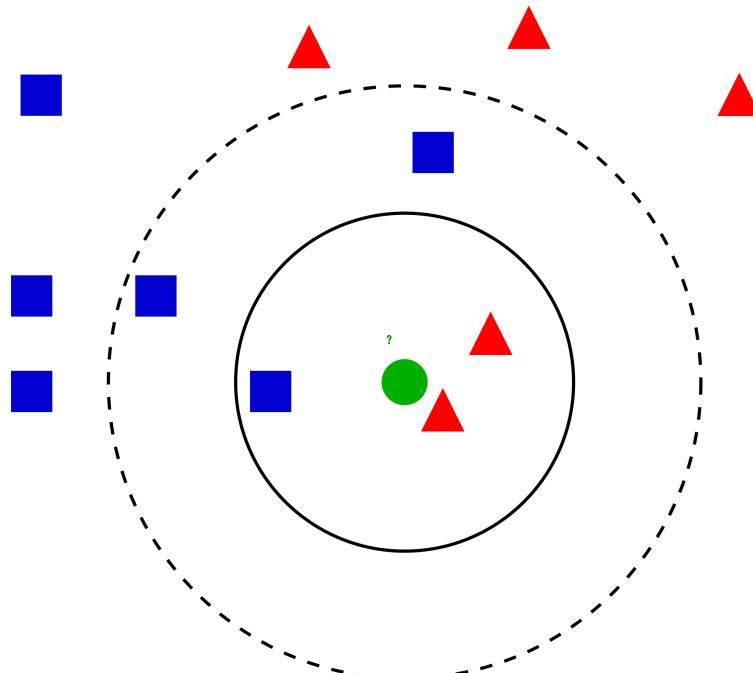


圖 2.1: 最近鄰居法概念圖

最近鄰居法 (K-nearest neighbors algorithm，簡稱 KNN) 是一種可用於分類 (Classification) 與回歸 (Regression) 的監督式學習演算法

(supervised learning)。在分類問題中，最近鄰居法從特徵空間 (Feature Space) 找到最近的 k 個訓練樣本，根據這 k 個鄰居的類別多寡決定目前測定的樣本類別。其概念如圖 2.1 所示，圖中的綠色樣本為待測定樣本，藍色與紅色為已觀察到的樣本，分別為類別藍與類別紅。根據選定的 k 不同，綠色待測樣本會被分類為不同類別。例如：選定 $k = 3$ 則綠色待測樣本會因為 3 個最近鄰居中，紅色類別較多被分類為紅色，但選定 $k = 5$ 則綠色待測樣本會因為 5 個最近鄰居中，藍色類別較多而被分成藍色。而在回歸問題中，最近鄰居法則是將周圍 k 個資料的平均值當成待測資料之結果。

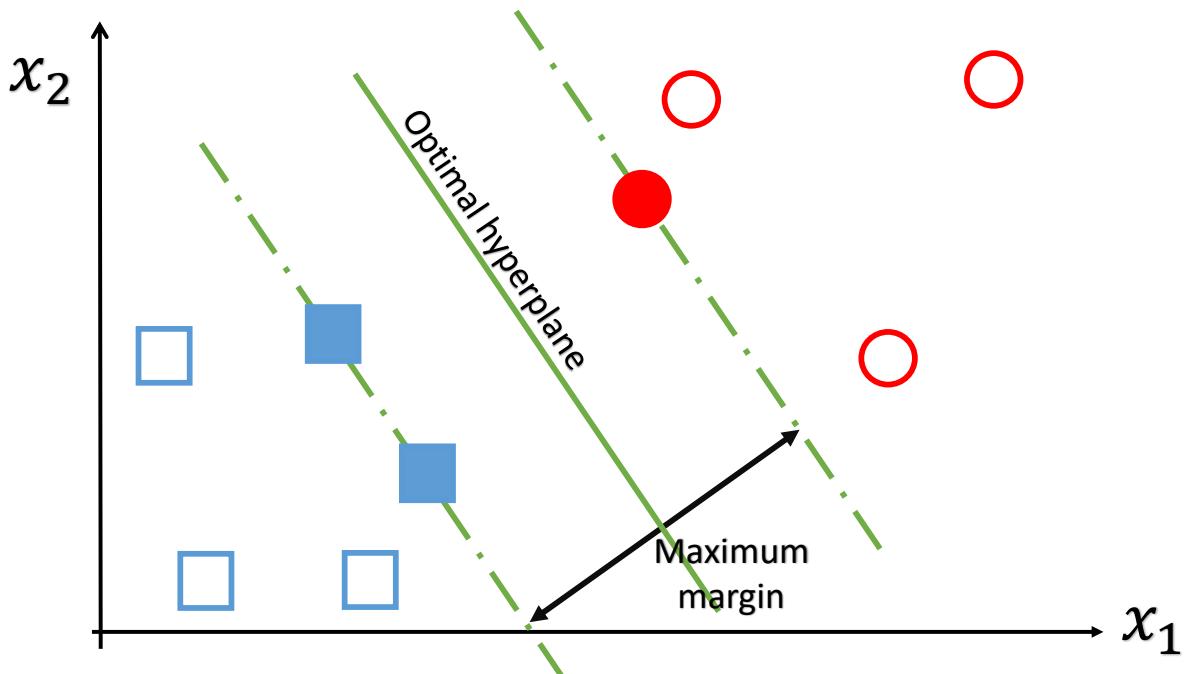


圖 2.2: 支援向量機概念圖

支援向量機 (Support vector machine, 簡稱 SVM)[3] 也是一種可用於分類與回歸的監督式學習演算法。其概念如圖 2.2 所示，簡單的概念為根據已知的資料找到一條分割線，使得這條線到不同類別的垂直距離最大。待測資料的分類就會依據其落在分割線的哪一側決定其類別。

決策樹 (Decision tree)[4] 也是一種可用於分類與回歸的監督式學習演算法。其概念如圖 2.3 所示，在訓練時，從根節點 (Root) 出發，根據訓練資料可以獲得最小熵值 (Entropy) 的特徵 (Feature) 分割點，分割出

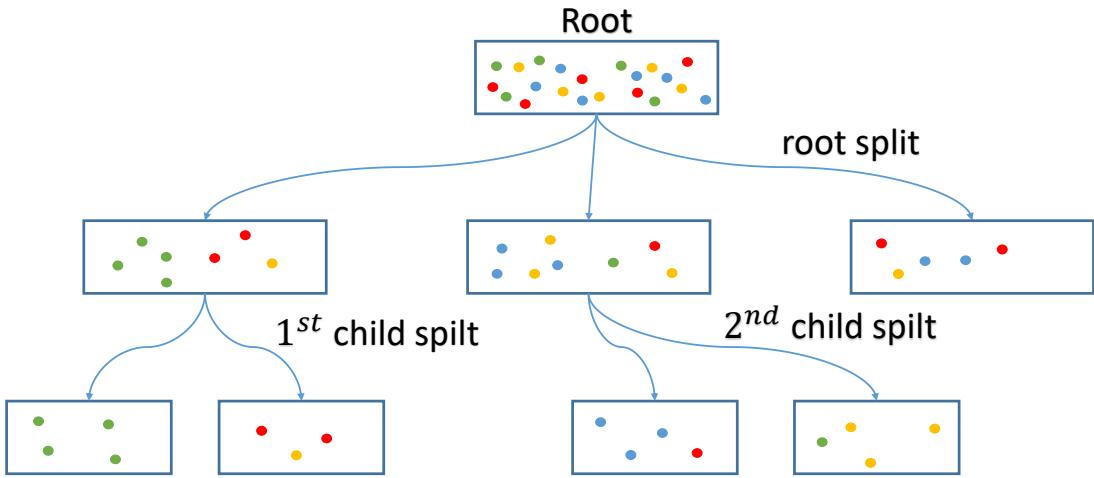


圖 2.3: 決策樹概念圖

目前節點 (Node) 的子節點 (child)，用相同的做法一直往下延伸，直到到達設定的最大分割次數或不能再分割為止。而在測定分類問題的未知資料或回歸問題的未知資料時，都是將未知資料餵進建好的樹中，從根節點開始一路往下爬，到最底部的節點就會顯示此資料的預測結果。

近期應用融合學習的機器學習已經越來越普及。系統化的實驗比較顯示，大部分的融合學跟能夠贏過單一模型 [5]。融合學習也在不同的領域大放異彩 [6, 7, 8]。融合學習建構在監督式學習 (supervised learning) 之上，監督式學習是機器學習演算法中最廣泛被應用及研究的分支，在這樣的基礎上，融合學習藉由融合多個簡單的監督式學習對某項任務的預測，藉此來增加假說空間 (hypothesis space)，並避免單一個模型的過適 (overfitting) 現象。在融合學習的領域，通常把被融合的模型稱為基礎學習器 (base learner) 或弱學習器 (weak learner)，以下將混用這些名詞。

接著介紹幾個較為有名的融合學習演算法，最常見也較直觀的可能是隨機森林 (random forest) [9]。其概念如圖 2.4 所示，隨機森林針對同一個資料集，由隨機的方法選取部份特徵或部份資料實例來產生多棵不

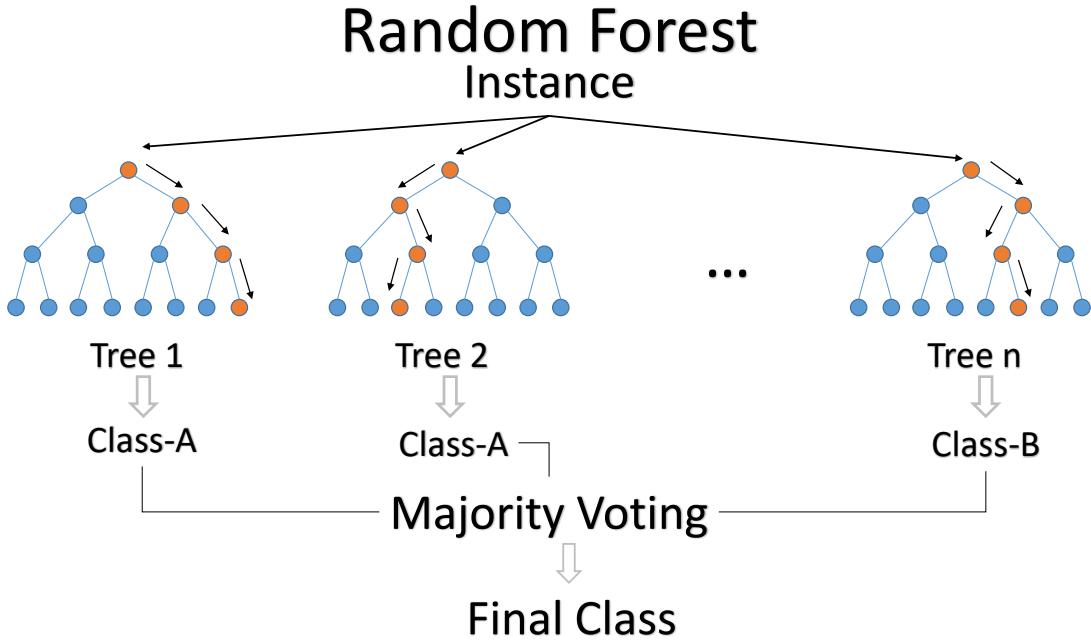


圖 2.4: 隨機森林概念圖

同的決策樹 (decision tree)，再將這些決策樹的結果經由簡單的投票表決或是取平均值來獲得最後的結果。如果將隨機森林中的決策樹換成一般的監督式模型，像是最近鄰居法或是支援向量機，並且每個模型都必須採用所有的特徵，這樣就變成經典的拔靴聚合演算法 (Bootstrap Aggregating，簡稱 Bagging) [10]。理論及實證上都顯示：若兩兩模型之間有足夠的差異，那融合的結果通常比起單一模型有較好的預測。

當然也可以用機器學習的方法，讓機器自動化的決定每個弱學習器的權重。這也就是統計學中著名的一般化附加模型 (Generative additive model，簡稱 GAM) [11]。最有名的一般化附加模型可能是自適應增強演算法 (Adaptive boosting，簡稱 Adaboost) [12]，此方法讓前一個基礎學習器分錯的樣本在下個回合有較高的權重，也就是在下一個回合這個樣本的對錯會優先被考慮，因此新的基礎學習器就有較大的可能把這個樣本分類正確。但也由於這樣的作法會使得異常值 (outlier) 被看得太重，容易導致過適現象 (overfitting)。後來的研究發現自適應增強演算法其實與梯度增強演算法 (gradient boosting) 採用指數損失函數作最佳化的結果等價 [13, 14]，故自適應增強演算法可被視為一種梯度增強演算法。

Gradient Boosting

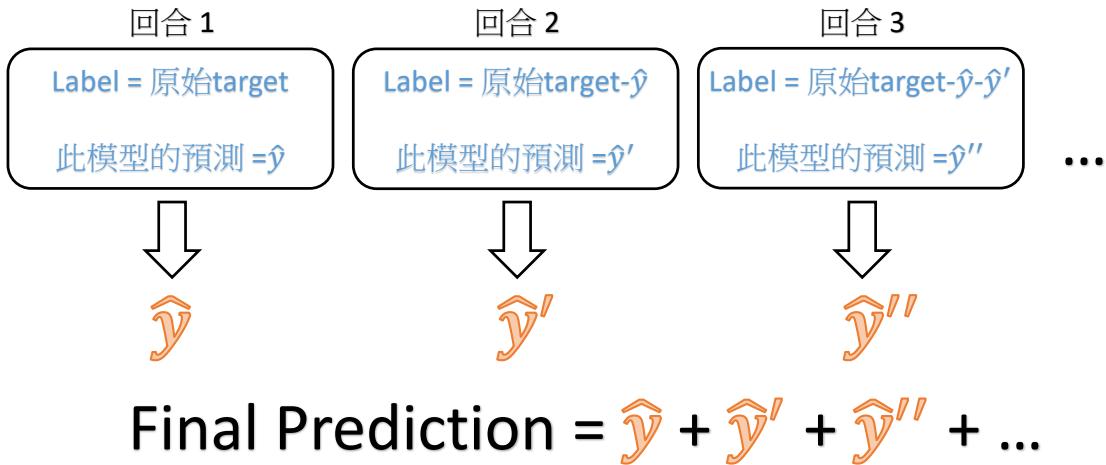


圖 2.5: 梯度增強演算法概念圖

那梯度增強演算法到底是甚麼呢？梯度增強演算法是一種靜態的多模型融合演算法，其概念如圖 2.5，在每個回合 (iteration) 訓練一個監督式模型，並根據上一個回合所產生的預測與實際值的誤差 (也稱為殘差 (residuals)) 當成這個回合的目標，最後將所有產生的模型預測相加做為最後的預測。實際面上，並不一定會以殘差來訓練模型，而是直接定義損失函式，再經由梯度下降法 (Gradient descent) 來尋找最佳解。

梯度增強演算法最有名的 XGBoost，在近期的許多資料競賽中得到優越的成績 [1, 15]，因此被工業界及學術界廣泛的採用。它的核心原理依舊是先前提到的梯度增強演算法。其概念如圖 ?? 所示，它從原先的方法中改進了損失函數 (Loss Function)，讓整體的效果變得更好。也由於它融合效果非常好，本篇論文也是以它做為最佳的比較目標。

Microsoft 提出的 LightGBM [16]，是一種靜態的多模型融合方法，其目的是為了在梯度增強決策樹 (gradient boosting decision tree) 的效能上下功夫，藉由梯度單邊採樣 (Gradient one-side sampling) 與稀疏特徵包裹 (Exclusive Feature Bundling)，來解決運用在資料集數量過多或特徵維度過大時，訓練時間過長與記憶體使用過多的情況。

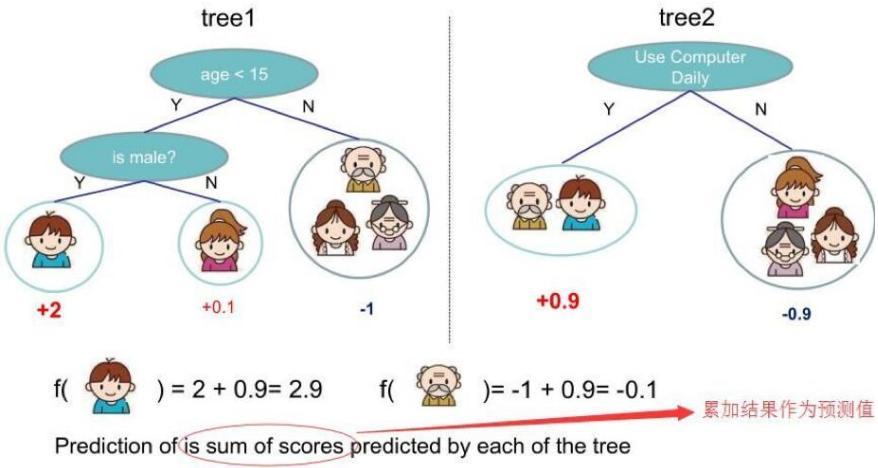


圖 2.6: XGBoost 概念圖 [1]

Catboost(categorical boosting) [17]，是一種能夠很好的處理類別特徵 (categorical feature) 的梯度增強演算法，它將類別特徵做特殊處理，使其有兩個優點：1. 在訓練過程中才處理類別特徵。2. 在選擇決策樹結構時，計算結點的方式不會太相似，可以避免過適現象。

無論是上述的哪種方法，皆是以「靜態」的方式做為融合的策略，然而這樣的融合方法是無法考慮到不同的基礎學習器是否有對特定資料的預測強弱。本論文想藉著設計「動態」的融合方式來增進多模型融合的效果。

在之後的專研中，也有發現類似想法的論文，META-DES [2]，其概念如圖 2.7 所示，它使用了動態的方式，從訓練好的基礎模擬器中，選取最優秀的幾個來做多數決。上述的方法與我們實驗的方法有相似與相異的地方，相似的部分在於都用了不同的方式實作了「動態」的概念，而相異的地方在於 META-DES 雖然「動態」挑選了不一樣的基礎學習器，但是在這些被挑出來的基礎學習器中依舊是用「靜態」的融合方法 (多數決)，而我們的方法則是將「動態」的概念放進權重的設計，讓融合的部分也是用「動態」的方式進行的。儘管有相異的地方，但他的實驗

也證明了與許多靜態方法相比，其效果比靜態融合方法優秀不少，這給我們在研究「動態」的權重上，有更大的信心。

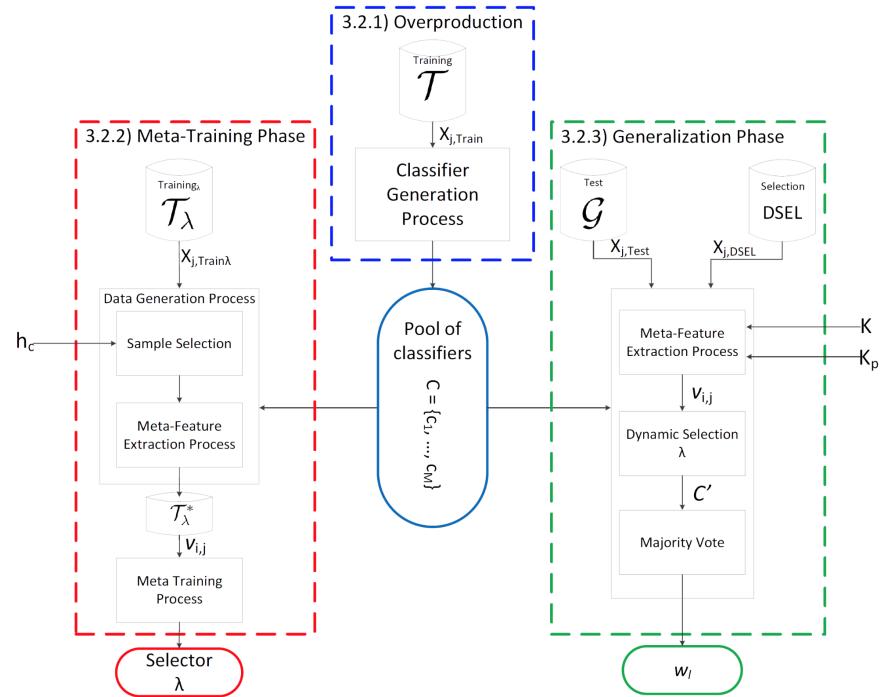


圖 2.7: META-DES 概念圖 [2]

三、動態多模型融合法

此篇論文提出並試驗了五種不同的想法並依次開發出五個動態模型，以下分述五種融合方法的原理。

3.1 根據基礎模型判斷類別機率做融合

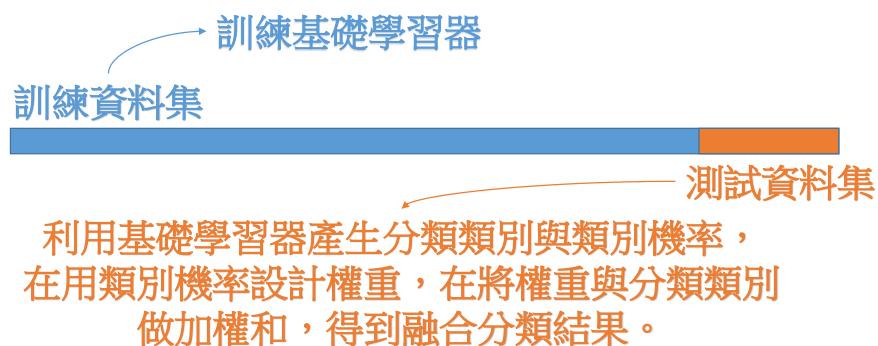


圖 3.1: 模型 1 資料集分割設計概念圖

為了動態的根據基礎學習器的猜測給予不同的權重，我們先從簡單的想法下手，對於不同的樣本實例讓每個基礎學習器輸出其判斷類別的機率。以二元分類問題為例，各個基礎學習器便會分別給予每個待測之樣本實例為正樣本及負樣本的機率。資料集分割設計概念如圖 3.1 所示，

在此方法中，我們將實驗資料分成：訓練資料集 (training set) 和測試資料集 (testing set)，並將權重的設計與加權後的輸出以 3.1 式、 3.2 式表示：

$$w_{ij} = \frac{P_i(X_j)}{\sum_{i=1}^n P_i(X_j)} \quad (3.1)$$

$$E_j = \sum_{i=1}^n w_{ij} \times C_i(X_j) \quad (3.2)$$

其中， w_{ij} 表示第 i 個學習器對第 j 個樣本的權重， $C_i(X_j)$ 表示第 i 個學習器對第 j 個樣本的預測， $P_i(X_j)$ 表示第 i 個學習器將第 j 個樣本預測成 $C_i(X_j)$ 的機率， E_j 表示模型融合後對第 j 個樣本的預測。由於同一個基礎學習器 i 對於不同的待測樣本 j 與 k 將產生 $P_i(X_j)$ 及 $P_i(X_k)$ 且通常 $P_i(X_j) \neq P_i(X_k)$ ，故對於不同的待測樣本，本模型將給予各個基礎模型不同的權重，以達到「動態」融合的效果。

3.2 將基礎模型判斷損失做融合

上個動態融合模型根據各個基礎學習器的預測結果做動態融合。然而，各個基礎學習器的各別預測能力好壞並未被列入考量。第二個動態融合模型將加入這點為考量：我們利用損失 (Loss) 來做為各個基礎學習器的加權權重。

資料集分割設計概念如圖 3.2 所示，這個模型將實驗資料集分成訓練分類器資料集 (training classifier set) 、測試資料集以及訓練融合屬性學習器資料集 (training Ensemble feature set)，會將資料集分成這樣的主要原因是在訓練時，我們可以根據目標變數 (target variable) 的真實值及預測值定義損失，但在測試階段時，我們並不知道待測樣本之目標變數的真實值，故無法計算損失來動態地決定各個基礎學習器的融合方式。為了估計損失，我們運用了訓練融合屬性學習器資料集來訓練一個預測

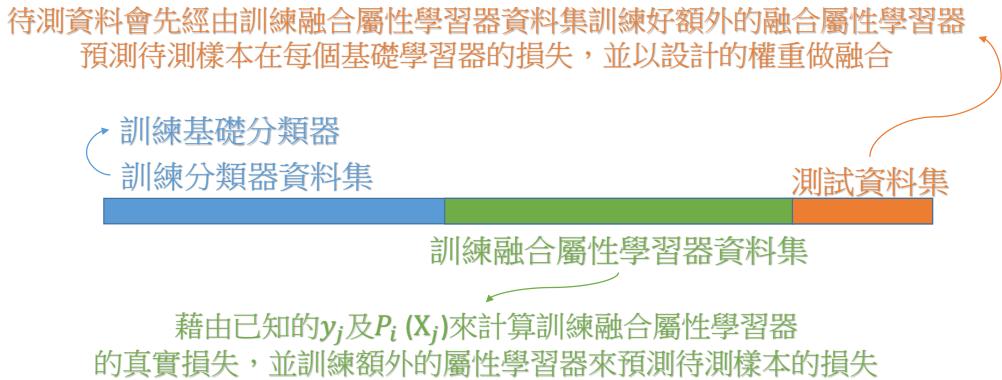


圖 3.2: 模型 2 資料集分割設計概念圖

損失的融合屬性學習器，這樣就解決在測試時，無法求得損失的問題。
損失的定義方式如 3.3 式：

$$Loss_j = |y_j - P_i(X_j)| \quad (3.3)$$

其中， $Loss_{ij}$ 代表第 i 個學習器對第 j 個樣本預測的損失， y_j 代表第 j 個樣本的真實類別。

定義損失後，權重則依損失值來計算。我們期望損失較小者之權重較大，故權重的設計與加權後的輸出以 3.4 式、3.5 式表示：

$$w_{ij} = \frac{\frac{1}{Loss_{ij}}}{\sum_{i=1}^n \frac{1}{Loss_{ij}}} \quad (3.4)$$

$$E_j = \sum_{i=1}^n w_{ij} \times C_i(X_j) \quad (3.5)$$

由於同一個基礎學習器 i 對於不同的待測樣本 j 與 k 將產生 $Loss_{ij}$

及 $Loss_{ik}$ 且通常 $Loss_{ij} \neq Loss_{ik}$ ，故對於不同的待測樣本，本模型將給予各個基礎模型不同的權重達到「動態」融合的效果。

然而，實驗後本模型的實際成效普通，仔細推敲之後，我們猜測只依賴某待測樣本單獨決定各個基礎學習器的損失值可能過度武斷，故我們又進一步提出第三個動態融合模型。

3.3 根據樣本空間的判斷能力做融合



圖 3.3: 模型 3 資料集分割設計概念圖

上節介紹的方法中各個學習器的損失是由訓練樣本決定的，為了估計各個基礎學習器在其他未見過的樣本上的損失值，我們改從樣本的特徵空間中抓取 k 筆與目前待估樣本相近的資料，藉由這些資料來判斷每個分類器在目前的樣本空間中的答對率，並依據答對率高低，來設計融合的方法。正確率的設計以 3.6 式表示：

$$Acc_{ij} = \frac{c_{ij}}{k_j} \quad (3.6)$$

其中， Acc_{ij} 代表第 i 個學習器在分類第 j 個樣本附近資料的答對率， k_j 代表在第 j 個樣本附近的 k 筆資料， c_{ij} 代表第 i 個學習器在這 k 筆資料中判斷正確的筆數。我們以 $k = 30$ 進行之後的實驗。

資料集分割設計概念如圖 4.1 所示，這個模型將實驗資料集分成如同模型 2 的分割法，訓練分類器資料集、測試資料集以及訓練融合屬性學習器資料集。但在使用上有些微不同。在這個模型中訓練融合屬性資料集是用來預測待測樣本的特徵空間中的鄰居在指定的基礎學習器上的答對率，這樣做的原因主要是因為當資料集變大時，利用類似 KNN 的方法必須看過所有的資料才能選出最近的 k 筆，這樣的情況下，勢必要花上許多時間，所以決定訓練一個額外屬性學習器來減少之後時間上的花費。權重的設計與加權後的輸出以 3.7 式、3.8 式表示：

$$w_{ij} = \frac{Acc_{ij}}{\sum_{i=1}^n Acc_{ij}} \quad (3.7)$$

$$E_j = \sum_{i=1}^n w_{ij} \times C_i(X_j) \quad (3.8)$$

由於同一個基礎學習器 i 對於不同的待測樣本 j 與 k 將產生 Acc_{ij} 及 Acc_{ik} 且通常 $Acc_{ij} \neq Acc_{ik}$ ，故對於不同的待測樣本，本模型將給予各個基礎模型不同的權重達到「動態」融合的效果。

實際上，第二個模型可以視為是第三個模型的一種特例。當第三個模型在訓練時將 KNN 的參數 k 設為 1，則與第二個模型等價。然而，將 k 設為大於 1 的值可以防止模型過適 (overfitting)。

3.4 根據樣本空間的答對個數做融合

在實驗過程中發現，章節 3.3 的權重是依據答對率產生，所以時常出現權重非常接近的情況，我們將答對率換成答對個數，資料集分割設



圖 3.4: 模型 4 資料集分割設計概念圖

計概念如圖 4.2 所示。資料切割與使用方式與章節 3.3 相同。權重的設計與加權後的輸出以 3.9 式、3.10 式表示：

$$w_{ij} = \frac{c_{ij}}{\sum_{i=1}^n c_{ij}} \quad (3.9)$$

$$E_j = \sum_{i=1}^n w_{ij} \times C_i(X_j) \quad (3.10)$$

其中， c_{ij} 代表第 i 個學習器在這 k 筆資料中判斷正確的筆數。

由於同一個基礎學習器 i 對於不同的待測樣本 j 與 k 將產生 c_{ij} 及 c_{ik} 且通常 $c_{ij} \neq c_{ik}$ ，故對於不同的待測樣本，本模型將給予各個基礎模型不同的權重達到「動態」融合的效果。

3.5 加入分類器判斷正確屬性作融合

經過發想，我們猜測或許每種不一樣的基礎分類器，有某一種隱藏在資料中的屬性，這種屬性會影響這筆資料會不會被判斷正確，所以我

們藉由訓練融合屬性學習器資料集中的資料再多做一層萃取，以獲得上述屬性。這些屬性我們稱之為 KSD 屬性，其設計概念如圖 4.3，我們假設被同一種基礎分類器答對的樣本都有一個相同的屬性，例如：被 KNN 答對的所有樣本我們都給他一個屬性 K 且 $K = 1$ ，來表示這些樣本都被 KNN 答對。再藉由這樣的資料經由訓練來猜測未知的樣本被 KNN 猜對的屬性 K 值為多少，若發現權重過近可嘗試以 softmax 函數將權重做歸一化，來拉開不同權重的距離。



圖 3.5: KSD 屬性設計概念圖

我們將此概念加入模型 3 模型 4，資料集分割設計概念如圖 4.4 所示，其權重表示以 3.11 式、3.12 式表示：

$$w_{ij} = \frac{Acc_{ij} \times KSD_{ij}}{\sum_{i=1}^n Acc_{ij} \times KSD_{ij}} \quad (3.11)$$

$$w_{ij} = \frac{c_{ij} \times KSD_{ij}}{\sum_{i=1}^n c_{ij} \times KSD_{ij}} \quad (3.12)$$

其中 KSD_{ij} 代表第 i 個分類器對第 j 個樣本的分類器判斷正確屬性。



圖 3.6: 模型 5 資料集分割設計概念圖

由於同一個基礎學習器 i 對於不同的待測樣本 j 與 k 將產生 KSD_{ij} 及 KDS_{ik} 且通常 $KSD_{ij} \neq KSD_{ik}$ ，故對於不同的待測樣本，本模型將給予各個基礎模型不同的權重達到「動態」融合的效果。

四、 實驗結果

4.1 資料集介紹

4.1.1 人工生成資料集

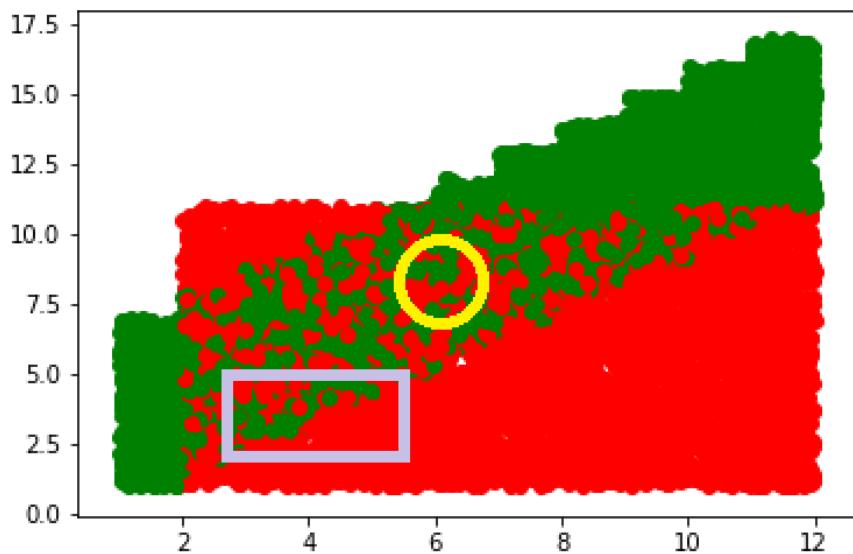


圖 4.1: 人工生成資料之正樣本及負樣本分布情況

為了讓不同的分類器有自己的優劣勢，所以我們用特別的方法合成了一組資料。我們實驗規劃中想使用的基礎分類器為 KNN、SVM、決策樹，所以在生成人工資料集時需要讓各個基礎分類器擅長或拙於不同的特徵組合。由於 KNN 是用周圍最近的 k 筆資料來決定目前這筆資料的類別，我們生成的資料中刻意讓樣本空間中的某一塊大量不同類別的資料堆疊在一起，這個區域中 KNN 預期將表現得特別差。決策樹會將

資料根據其值做分割，所以在資料空間能夠水平切割或垂直切割就可以分類的不錯，我們刻意讓某一區塊的資料類別是斜線分割的，這樣決策樹在這部分就有劣勢。SVM 在這樣的資料設計中已經很有劣勢，就沒有特別在為他設計劣勢的地方。資料的分布如圖 4.1所示，其中黃色圓圈會是 KNN 有劣勢的地方，淡紫色方框處會是決策樹有劣勢的地方。實驗結果如章節 4.3。在模型一、模型二與模型三的實驗過程中，發現了我們人工生成資料集的一些問題，像是我們所產出的資料集可能有稍微地與我們的初衷不相符。我們期待的資料是要讓不同的基礎學習器，其判斷能力會隨著不同的樣本而變強或變弱，但我們生成的資料集，讓每個基礎學習器判斷的能力都下降了，沒有我們所期待的樣子，這也可能是使得我們的實驗結果不盡理想的原因。我們接著用現實的資料集，來更明確的確定與了解「動態」多模型融合法是不是真的有它厲害的地方。

4.1.2 車險資料集

車險資料集使用 Kaggle 上 Allstate Claims Severity 的資料集有 150654 筆訓練資料 (training data) 與 37664 筆測試資料 (testing data)，每筆資料有 116 個類別特徵 (categorical feature) 與 14 個數值特徵 (numerical feature))，其目標變數為預測使用者投保後保險公司的損失，圖 4.2為原始資料的擷取。由於目前所嘗試的模型皆為分類的問題，所以將原本的回歸問題轉化成分類問題，我們利用圖 4.3的峰值，切割成損失大於 2500 與損失小於 2500 兩類，將原本的回歸問題變成分類問題。在前處理的部分，我們將原本 116 個類別特徵，用 one-hot encode 的方式編碼，由原本 116 個特徵變成 1140 個特徵再加上 14 個數值特徵，將 130 維的資料變成 1154 維較為稀疏 (sparse) 的資料。其實驗結果如章節 4.4

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	id	cat1	cat2	cat3	cat4	cat5	cat6	cat7	cat8	cat9	cat10	cat11	cat12	cat13	cat14
2	1	A	B	A	B	A	A	A	A	B	A	B	A	A	A
3	2	A	B	A	A	A	A	A	A	B	B	A	A	A	A
4	5	A	B	A	A	B	A	A	A	B	B	B	B	B	A
5	10	B	B	A	B	A	A	A	A	B	A	A	A	A	A
6	11	A	B	A	B	A	A	A	A	B	B	A	B	A	A
	DO	DP	DQ	DR	DS	DT	DU	DV	DW	DX	DY	DZ	EA	EB	
1	cont2	cont3	cont4	cont5	cont6	cont7	cont8	cont9	cont10	cont11	cont12	cont13	cont14	loss	
2	0.245921	0.187583	0.789639	0.310061	0.718367	0.33506	0.3026	0.67135	0.8351	0.569745	0.594646	0.822493	0.714843	2213.18	
3	0.737068	0.592681	0.614134	0.885834	0.438917	0.436585	0.60087	0.35127	0.43919	0.338312	0.366307	0.611431	0.304496	1283.6	
4	0.358319	0.484196	0.236924	0.397069	0.289648	0.315545	0.2732	0.26076	0.32446	0.381398	0.373424	0.195709	0.774425	3005.09	
5	0.555782	0.527991	0.373816	0.422268	0.440945	0.391128	0.31796	0.32128	0.44467	0.327915	0.32157	0.605077	0.602642	939.85	
6	0.15999	0.527991	0.473202	0.704268	0.178193	0.247408	0.24564	0.22089	0.2123	0.204687	0.202213	0.246011	0.432606	2763.85	

圖 4.2: 車險資料集原始資料集擷取

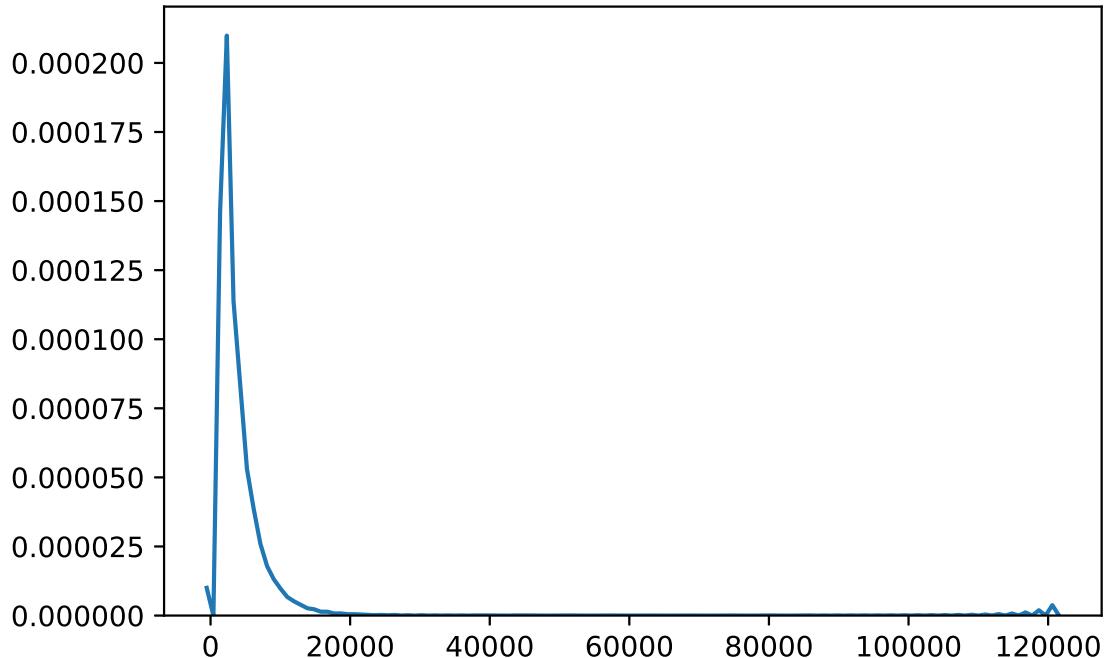


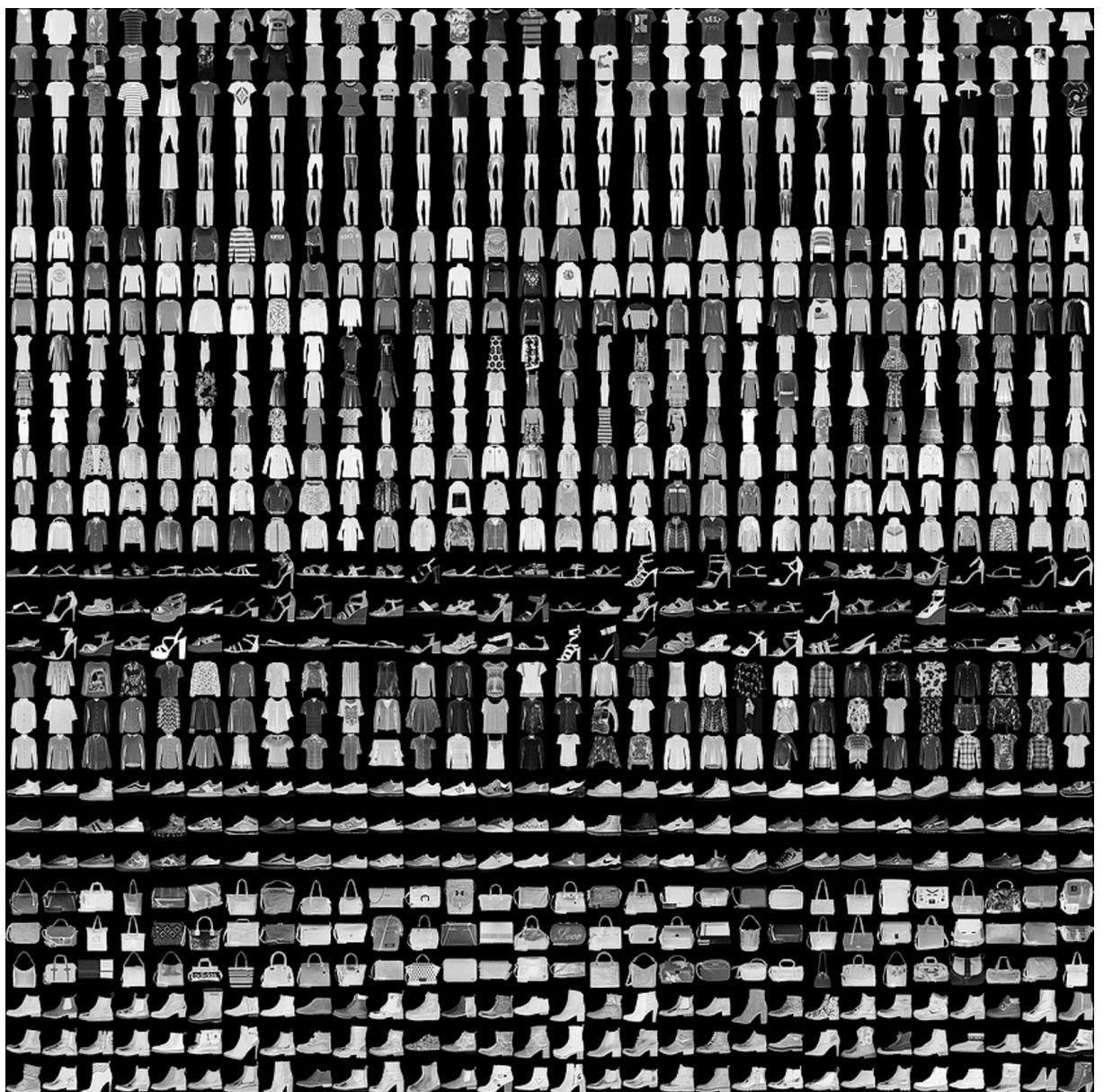
圖 4.3: 車險資料集 Target 的分布狀況

4.1.3 Fashion-MNIST 資料集

Fashion-MNIST 是 Zalando 的論文的衣服圖片資料集 [18]，其中包含 60000 筆訓練資料與 10000 筆測試資料，每筆資料由 28×28 像素組成的灰階圖片，對應到 10 個不同的類別，每個類別的數量是一致的。Zalando 打算將 Fashion-MNIST 當成 MNIST 的替代品，用來衡量不同的機器學習演算法。圖 4.4為 Fashion-MNIST 範例其，對應的 10 個類別如下表 4.1。其實驗結果如章節 4.5。

表 4.1: Fashion-MNIST 10 個 class 的種類

類別號碼	衣服種類
0	T 恤衫\上衣
1	褲子
2	套頭衫
3	連衣裙
4	外套
5	涼鞋
6	襯衫
7	運動鞋
8	背包\袋子
9	踝靴



4.1.4 Kuzushiji-MNIST 資料集

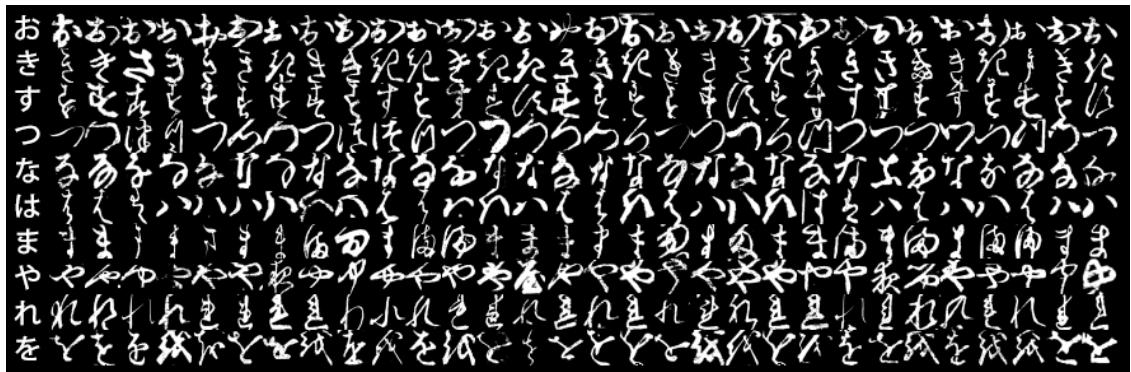


圖 4.5: Kuzushiji-MNIST 範例

Kuzushiji-MNIST 是日文手寫資料集 [19]，其中包含 60000 筆訓練資料與 10000 筆測試資料，每筆資料由 28×28 像素組成的灰階圖片，對應到 10 個不同的類別，每個類別的數量是一致的。Kuzushiji-MNIST 也是一個 MNIST 的替代品，用來衡量不同的機器學習演算法。其對應的 10 個類別如下表 4.2，圖 4.5為 Kuzushiji-MNIST 範例。其實驗結果如章節 4.6。

表 4.2: Kuzushiji-MNIST 10 個 class 的種類

類別號碼	日文種類
0	あ (o)
1	き (ki)
2	す (su)
3	つ (tsu)
4	な (na)
5	は (ha)
6	ま (ma)
7	や (ya)
8	れ (re)
9	を (wo)

4.2 實驗設定

五種動態多模型融合使用的四個基本模型採用：KNN、SVM、決策樹與 XGBoost，KNN 採用 $K = 5$ ，SVM 的 kernel 採用 Poly kernel，決策樹採用 default 設定，XGBoost 可調整參數較多，使用的參數是 Benchmark 上分享的最高分數設定。基本模型與用來融合的模型在使用的資料集上有稍稍的不同，基本模型使用資料集的百分之 80 做為訓練資料集，剩下的百分之 20 做為測試資料集，而用來融合的模型則使用章節 3 的切割方法來訓練與測試模型，在這樣的設定下才會讓模型間公平的競爭。額外的融合屬性（損失、答對率、答對個數、KSD）的學習器在不一樣的資料集使用不一樣的模型。在人工生成資料集、Fashion-MNIST 與 Kuzushiji-MNIST 上，學習答對率與答對個數中，使用簡單的線性回歸（Linear Regression），而在學習 KSD 屬性則是使用 XGBoost。車險資料集維度較大若不使用較複雜的模型，則預測和實際答案不會呈現正相關，所以我們在車險資料集學習答對率與答對個數是使用 4 層的類神經網路（Neutral network），然而學習 KSD 屬性較不會

有上述問題，依舊採用 XGBoost 學習 KSD 屬性。

在實驗過程中，我們發現幾個可以一起比較的方法：

1. Model 1 是使用機率的模型，那這樣的融合法跟一般根據機率和來分類的模型 (例如： $totalP(1) > totalP(0)$ ，則分類成 class1。) 可以相互比較。
2. 在 Model 2 中我們使用損失這個概念，其中損失越大權重越小，但權重可以使用 $\frac{1}{Loss}$ 與 $1 - Loss$ 都同樣符合損失越大權重越小，這也是一個可以比較的部分。
3. 有沒有使用 KSD 屬性及有沒有使用 softmax 函式對權重做歸一化，也分別列出做比較。
4. 由於其中有額外訓練的屬性學習器，我們也與沒有這些預測誤差的模型做比較，我們把這個模型稱為 Model 5 理想值，這個 Model 的融合是根據真實計算測試資料的屬性 (例如：答對個數、答對準確率、KSD 屬性…)，其中有可能包含真實類別的資訊，但回報這個的重點是從理論上出發，如果額外學習屬性的學習器能有更好的表現的話 (已確認目前學習的結果與真實結果為正相關)，是有機會使這整個融合法變得更好的。

經過不同的測試發現，對於多類別的資料集 (Fashion-MNIST、Kuzushiji-MNIST)，設計的方法有可能會因為類別維度的關係而被稀釋，所以我們將大問題拆成小問題，將所有的基礎分類器做成 *one versus all* 的分類器，使得在融合上面，其效果不會稀釋。

4.3 人工生成資料集結果

雖然在實驗過程中，有發現產生人工生成資料集的問題，但在接下來的結果依舊將實驗結果列出比較。下表 4.3為人工生成資料集在五個模型的結果。在這次實驗中看到，除了 Model 2 的動態多模型融合的效果低於多數決之外，其他我們設計的模型皆高於多數決，我們的推測是原本設計用損失的出發點是好的，但可能由於預測損失的學習器有偏差，

導致發生目前的結果。實驗結果最好的是 Model 5-1，也是我們預料之中的結果，從資料集中多提取的資訊，只要預測這些資訊的模型表現不要太差，實驗結果就會不錯。而跟理想 Model 的比較中可以發現，可以進步的空間還很多，只要能將額外的融合屬性學好，實驗結果可以更好。

表 4.3: 人工生成資料集在三種基礎學習器 (KNN、SVM、Decision tree)、一種典型的融合模型 (Majority Voting)、及三種我們提出的動態融合模型的準確率比較 (單位：%)

分類器	準確率 (%)
KNN	78.29
SVM	76.58
Decision tree	77.56
XGBoost	73.12
Majority Voting	77.58
Model 1-1	78.67
Model 機率和	77.78
Model 2-1(1/loss)	76.75
Model 2-2(1-loss)	77.92
Model 3-1(準確率 without softmax)	78.67
Model 3-2(準確率 with softmax)	78.67
Model 4-1(個數 without softmax)	78.67
Model 4-2(個數 with softmax)	78.67
Model 5-1(準確率 with KSD without softmax)	78.92
Model 5-2(個數 with KSD without softmax)	78.83
Model 5-3(準確率 with KSD & softmax)	78.88
Model 5-4(個數 with KSD & softmax)	78.25
Model 5 理想值 (without softmax)	95.17
Model 5 理想值 (with softmax)	94.88

4.4 車險料集結果

下表 4.4為車險資料集在五個模型之結果。從結果來看，除了 Model 1 與 Model 2 輸給多數決模型，其餘的模型皆優於或等於多數決模型，其中 Model 4-2 實驗結果最好，代表著答對個數這個融合屬性有它的影響力，也可以看出 softmax 函數在這種權重較相近的資料集中有它的影響力，但就實驗結果而言這個模型也僅僅是與四個基礎學習器中最厲害的 XGBoost 結果差距較小。令我們意外的是 KSD 屬性，反而在這個資料集沒有表現出它的優勢，雖然 Model 5-4 的實驗結果也不錯，但實際上融合的結果是稍稍地輸給了其中一個厲害的基礎學習器。跟理想 Model 的比較可以發現，如果可以將額外的融合屬性學習器學得更好，我們就可以得到更為接近理想 Model 的成績，也就是使基礎分類器進步約 10% 左右的準確度。

表 4.4: 車險資料集在基礎學習器 (KNN、SVM、Decision tree)、一種典型的融合模型 (Majority Voting)、及五種我們提出的動態融合模型的準確率比較 (單位 : %)

分類器	準確率 (%)
KNN	71.75
SVM	69.50
Decision tree	69.89
XGBoost	79.27
Majority Voting	77.20
Model 1-1	76.73
Model 機率和	75.42
Model 2-1(1/loss)	72.27
Model 2-2(1-loss)	73.05
Model 3-1(準確率 without softmax)	77.74
Model 3-2(準確率 with softmax)	77.74
Model 4-1(個數 without softmax)	77.74
Model 4-2(個數 with softmax)	79.07
Model 5-1(準確率 with KSD without softmax)	78.71
Model 5-2(個數 with KSD without softmax)	77.93
Model 5-3(準確率 with KSD & softmax)	77.73
Model 5-4(個數 with KSD & softmax)	79.04
Model 5 理想值 (without softmax)	92.45
Model 5 理想值 (with softmax)	91.80

4.5 Fashion-MNIST 結果

下表 4.5為 Fashion-MNIST 資料集在五個模型之結果。其中幾個模型想特別提出來說明，像是 Model 機率和、Model 2-1 及 Model 2-2，這三個模型雖然都輸給多數決，但這樣的結果其實證明了一些事情。從 Model 機率和來說，這個模型並沒有用到需要由額外的融合屬性學習器提供的屬性，可是這些額外的融合屬性對多模型融合的影響是好的，又因為 Model 機率和沒有使用，其效果反而不出色。Model 2-1 與 Model

2-2 就是反向，他雖然多用了損失這個額外融合屬性，但由於這個屬性有點過適或是學習這個屬性的學習器有偏差，導致實驗結果不佳，如同章節 3.3 中提到 Model 2 是 Model 3 的特例，可以發現到考慮多個資料能解決過適的問題，也可能讓實驗結果更好，而 Model 3 的確在實驗結果的部分有更好的表現。雖然我們的動態多模型融合法，沒有超過 XGBoost，以下是幾個推測：

1. 基礎分類器中的 XGBoost，其學習效果就沒想像中好。
2. 額外的融合屬性學習器可能學得不夠好。

如果能將上述兩個改進，可能就有會超過 XGBoost。

表 4.5: Fashion-MNIST 資料集在基礎學習器 (KNN、SVM、Decision tree、XGBoost)、兩種典型的融合模型 (Majority Voting、XGBoost)、及五種我們提出的動態融合模型的準確率比較 (單位 : %)

分類器	準確率 (%)
KNN (10 個 one vs all)	85.02
SVM (10 個 one vs all)	85.97
Decision tree (10 個 one vs all)	74.77
XGboost (10 個 one vs all)	88.30
Majority Voting	89.80
Model 1-1	89.88
Model 機率和	85.43
Model 2-1(1/loss)	87.59
Model 2-2(1-loss)	87.05
Model 3-1(準確率 without softmax)	89.77
Model 3-2(準確率 with softmax)	89.80
Model 4-1(個數 without softmax)	89.77
Model 4-2(個數 with softmax)	89.08
Model 5-1(準確率 with KSD without softmax)	88.46
Model 5-2(個數 with KSD without softmax)	88.49
Model 5-3(準確率 with KSD & softmax)	90.10
Model 5-4(個數 with KSD & softmax)	89.98
XGBoost	90.60
Model 5 理想值 (without softmax)	98.84
Model 5 理想值 (with softmax)	94.88

4.6 Kuzushiji-MNIST 結果

下表 4.6為 Kuzushiji-MNIST 資料集在五個模型之結果。其中 Model 機率和、Model 2-1 及 Model 2-2，這三個模型雖然都輸給多數決，原因跟在 Fashion-MNIST 一樣。我們的動態多模型融合法，也在這個資料集上超過了 XGBoost，如同我們在章節 4.5推測，只要基礎分類器的 XGBoost，學習的效果還不錯，再加上能將額外的融合屬性學好，我相信每個資料集都能有很好的成果。

表 4.6: Kuzushiji-MNIST 資料集在基礎學習器 (KNN、SVM、Decision tree、XGBoost)、兩種典型的融合模型 (Majority Voting、XGBoost)、及五種我們提出的動態融合模型的準確率比較 (單位 : %)

分類器	準確率 (%)
KNN (10 個 one vs all)	88.62
SVM (10 個 one vs all)	84.25
Decision tree (10 個 one vs all)	56.93
XGboost (10 個 one vs all)	81.91
Majority Voting	89.59
Model 1-1	89.56
Model 機率和	81.99
Model 2-1(1/loss)	86.19
Model 2-2(1-loss)	83.62
Model 3-1(準確率 without softmax)	89.93
Model 3-2(準確率 with softmax)	89.91
Model 4-1(個數 without softmax)	89.93
Model 4-2(個數 with softmax)	89.56
Model 5-1(準確率 with KSD without softmax)	90.10
Model 5-2(個數 with KSD without softmax)	90.05
Model 5-3(準確率 with KSD & softmax)	89.96
Model 5-4(個數 with KSD & softmax)	89.96
XGBoost	88.21
Model 5 理想值 (without softmax)	99.70
Model 5 理想值 (with softmax)	92.52

五、 討論與未來發展

我們提出了五種不同的動態融合學習方法，藉由「動態」的給予權重，來試著將不同基礎學習器對不同事件的預測考慮進來。實驗結果顯示，在不同資料集有不一樣的表現結果：

1. 人工生成資料集中，實驗結果顯示超過 XGBoost，但由於資料的生成並不符合我們生成的初衷，但有從中發現額外的融合屬性學習器的重要。
2. 車險資料集中，多數模型優於多數決模型，但由於額外的融合屬性學習器的學習，沒有達到預期的學習效果，導致有些模型的表現不如預期。
3. Fashion-MNIST 中，有些模型沒有優於多數決，但也有其發生的原因，例如：沒有採用好的額外融合屬性或額外融合屬性的訓練不優秀，使得融合效果變差。在跟 XGBoost 的比較中，我們的動態多模型融合法沒有超過 XGBoost，原因在於基礎學習器用 *one versus all* 訓練出來的效果與單純的使用 XGBoost 差距太多，再加上額外的融合屬性學習器無法完美的學習，才導致目前的結果。
4. Kuzushiji-MNIST 中，依舊有些模型沒有優於多數決，發生的原因如同 Fashion-MINST，但在跟 XGBoost 的比較中，我們的動態多模型融合法些許的超過了 XGBoost，跟 Fashion-MNIST 不同的地方是基礎學習器用 *one versus all* 訓練出來的效果沒有低於單純的使用 XGBoost 太多，而額外的融合屬性學習器與應用在 Fashion-MNIST 上差不多，使得整體的融合效果超過了 XGBoost，這顯示動態融合模型看似是一種可行的作法，但還在額外融合屬性的訓練上就需要多加注意與確認。

從比較目前結果與理想結果中，我們也發現一些有趣的地方，像是

額外的融合屬性中，KSD 屬性是影響力最大的，其次是答對個數或答對率，這兩個屬性都是對整個動態多模型融合法至關重要的，但由於這兩個問題都是回歸問題，而在回歸問題中，很難說現在訓練的結果是好是壞，只能從 R2 score 中知道，目前的學習與正確答案呈現正相關，但要如何將這些額外的融合屬性學好又是一個可以專研的課題。

參考文獻

- [1] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, “Xgboost: extreme gradient boosting,” *R package version 0.4-2*, pp. 1–4, 2015.
- [2] R. M. Cruz, R. Sabourin, G. D. Cavalcanti, and T. I. Ren, “Meta-des: A dynamic ensemble selection framework using meta-learning,” *Pattern recognition*, vol. 48, no. 5, pp. 1925–1935, 2015.
- [3] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [4] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.
- [5] Y. Freund, R. E. Schapire, *et al.*, “Experiments with a new boosting algorithm,” in *icml*, vol. 96, pp. 148–156, Citeseer, 1996.
- [6] T. G. Dietterich *et al.*, “Ensemble learning,” *The handbook of brain theory and neural networks*, vol. 2, pp. 110–125, 2002.
- [7] H.-F. Yu, H.-Y. Lo, H.-P. Hsieh, J.-K. Lou, T. G. McKenzie, J.-W. Chou, P.-H. Chung, C.-H. Ho, C.-F. Chang, Y.-H. Wei, *et al.*, “Feature engineering and classifier ensemble for kdd cup 2010,” in *KDD Cup*, 2010.
- [8] A. Niculescu-Mizil, C. Perlich, G. Swirszcz, V. Sindhwani, Y. Liu, P. Melville, D. Wang, J. Xiao, J. Hu, M. Singh, *et al.*, “Winning the kdd cup orange challenge with ensemble selection,” in *KDD-Cup 2009 Competition*, pp. 23–34, 2009.
- [9] A. Liaw, M. Wiener, *et al.*, “Classification and regression by randomforest,” *R news*, vol. 2, no. 3, pp. 18–22, 2002.
- [10] L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [11] T. J. Hastie, “Generalized additive models,” in *Statistical models in S*, pp. 249–307, Routledge, 2017.

- [12] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [13] L. Breiman *et al.*, “Arcing classifier (with discussion and a rejoinder by the author),” *The annals of statistics*, vol. 26, no. 3, pp. 801–849, 1998.
- [14] L. Breiman, “Prediction games and arcng algorithms,” *Neural computation*, vol. 11, no. 7, pp. 1493–1517, 1999.
- [15] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 785–794, ACM, 2016.
- [16] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems*, pp. 3146–3154, 2017.
- [17] A. V. Dorogush, V. Ershov, and A. Gulin, “Catboost: gradient boosting with categorical features support,” *arXiv preprint arXiv:1810.11363*, 2018.
- [18] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.
- [19] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, “Deep learning for classical Japanese literature,” *arXiv preprint arXiv:1812.01718*, 2018.

附錄一

實驗程式碼：https://github.com/ncu-dart/Dynamic_Engsemble

DELR (Dynamic Ensemble Learning Research)

1.if you want to try the dataset I used in my experiment set as follow
set 'setup.txt' as follow
for the synthetic dataset set classnumber=2, dataset=1
for the allstate dataset set classnumber=2, dataset=2
for the fashion-mnist dataset set classnumber=10, dataset=3
for the kuzushiji-mnist dataset set classnumber=10, dataset=4

after setting all things that mention above run DELR.py

2.if you want to try my method on your own dataset set as follow
preprocessed the dataset as you wish
the first column should be the label of the instance and saved as csv
set 'setup.txt' as follow
set classnumber to the numbers of class in your dataset
set dataset=5
set dataname as taskname
rename your dataset as dataname-train and dataname-test

after setting all things that mention above run DELR.py