 DOI: 10.6861/TANET.201810.0229

動態多模型融合分析研究

蘇俊儒　陳弘軒
中央大學 資訊工程學系
jim83531@gmail.com, hhchen@ncu.edu.tw

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。我們已經嘗試了三種不一樣的方法，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；及 (3) 根據樣本空間的判斷能力，這三種不同的方法來實做「動態」整合。

本文將說明我們設計的三種方法，並在人工生成資料集上進行初步的實驗。我們設計的三個整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應該是可行的。然而，三種方法中只有一種方法略優於靜態的多模型融合方式 Majority Voting，故在動態多模型融合的設計上還能再加強。

關鍵詞：多模型融合、動態多模型融合、監督式學習

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 fusion of the basic models will not adapt based on 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 three different methods, based on (1) the category probability predicted by the base learners; (2) the loss of the base learners; and (3) the correctness in feature space of base learner. These three methods realize the "dynamic" ensemble.

This article will explain the three methods we designed and the preliminary experimental results based on a simulated dataset. We found that all three ensemble methods perform better than each of the single base learners. However, only one of the three methods is better than Majority Voting, a popular static ensemble model. Therefore, it may still be possible to improve our methods.

Keywords: ensemble learning, dynamic ensemble learning, supervised learning

1. 簡介

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固定，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，有很大的機會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。

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正在進行中的研究方向。

2. 背景及相關論文

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，大部分的融合學習能夠贏過單一模型[1]。融合學習也在不同的領域大放異彩[2,3,4]。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一支，在這樣的基礎上，融合學習藉由整合多個簡單的監督式學習對某一項任務的預測，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，通常把被整合的模型稱為基礎學習器 (base learner) 或弱學習

 DOI: 10.6861/TANET.201810.0229

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，最常見也較直觀的可能是隨機森林 (random forest)。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同的決策樹 (decision tree)，再將這些決策樹的結果經由簡單的投票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 (*k*-nearest neighbors，簡稱 KNN) 或是支援向量機 (support vector machine，簡稱 SVM)，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) [5]。理論及實證上都顯示：若兩兩模型之間有足夠的差異，那整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) [6]。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) [7]，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 [10,11]，故自適應增強演算法可被視為一種梯度增強演算法。

近期 XGBoost 在許多資料競賽中得到優越的成績 [8,9]，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。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。

無論是上述的哪種方法，皆是以「靜態」的方式做為整合的策略，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」的整合方式來增進多模型融合的效果。

3. 動態多模型融合法

此篇論文提出並試驗了三種不同的想法並依次開發出三個動態模型，以下分述三種融合方法的原理。

3.1 根據基礎模型判斷類別機率做整合

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。在此方法中，我們將實驗資料分成：training set 和 testing set，並將權重的設計與加權後的輸出以 (1)、(2) 式表示：

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

$$E_j = \sum_{i=1}^{n} w_{ij} * C_i(X_j) \tag{2}$$

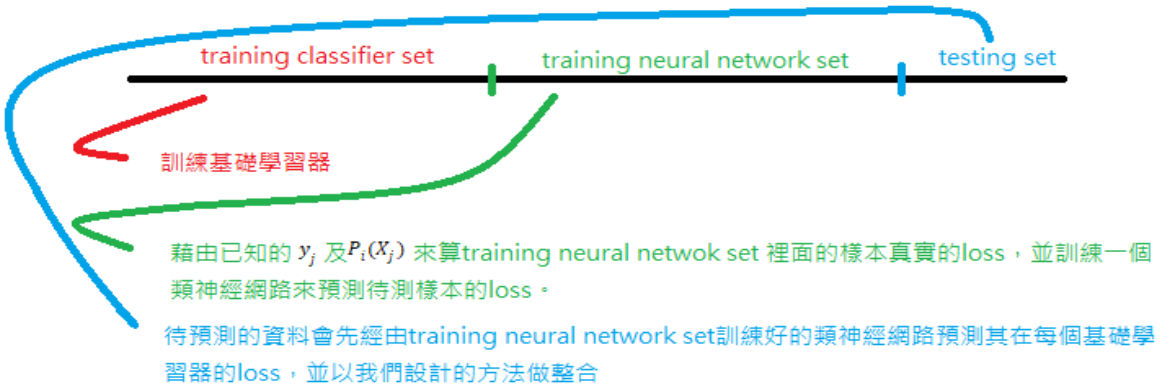
其中，w_{ij}表示第 i 個學習器對第 j 個樣本的權重，$C_i(X_j)$表示第 i 個學習器對第 j 個樣本的預測，$P_i(X_j)$表示第 i 個學習器將第 j 個樣本預測成$C_i(X_j)$的機率，E_j表示模型融合後對第 j 個樣本的預測。

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整合

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整合。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。

如圖一所示，這個模型將實驗資料集分成 training classifier set 、 testing set 以及 training



圖一：模型 2 設計概念圖

 DOI: 10.6861/TANET.201810.0229

neural network set，會將資料集分成這樣的原因主要是因為在訓練時，我們可以根據目標變數 (target variable) 的真實值及預測值定義損失 (loss)，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，故無法計算損失值動態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 training neural network set 來訓練一個預測損失的類神經網路，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)式：

$$Loss_{ij} = |y_j - P_i(X_j)| \tag{3}$$

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(4)、(5)式表示：

$$w_{ij} = \frac{\frac{1}{Loss_{ij}}}{\sum_{i=1}^{n}\frac{1}{Loss_{ij}}} \tag{4}$$

$$E_j = \sum_{i=1}^{n} w_{ij} * C_i(X_j) \tag{5}$$

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整合

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之正確率，並藉由正確率高低，來設計整合的方法。正確率的設計以(6)式表示：

$$Acc_{ij} = \frac{c_{ij}}{k_j} \tag{6}$$

其中，Acc_{ij}代表第 i 個學習器在第 j 個樣本附近資料的正確率，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。

這個模型將實驗資料集分成如同模型2的分割法，training classifier set 、 testing set 以及 training neural network set。但在使用上有些微不同。在這個模型中 training neural network set 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正確率，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，勢必是要花上許多時間的，所以決定訓練一個類神經網路來減少之後時間上的花費。權重的設計與加權後的輸出以 (7)、(8)式表示：

$$w_{ij} = \frac{Acc_{ij}}{\sum_{i=1}^{n} Acc_{ij}} \tag{7}$$

$$E_j = \sum_{i=1}^{n} w_{ij} * C_i(X_j) \tag{8}$$

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 的值有可能可以防止模型過適 (overfitting)。

4. 實驗

此篇論文的實驗，所使用的人工資料集、實驗設定與結果將會依次的說明。

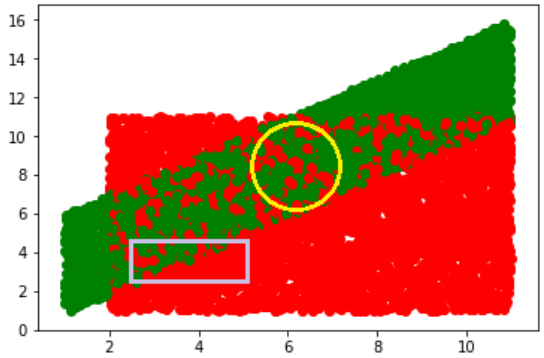
4.1 實驗資料

我們目前在人工生成的資料集上比較上述三個模型。以下說明資料的生成方式。

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、Decision tree，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。Decision tree 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，我們意讓資料在某一區塊資料的類別是斜線分割的，這樣 Decision tree 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。

資料的分布如圖二所示。其中黃色圈圈是 KNN 有劣勢的地方，淡紫色框框是 Decision tree 有劣勢的地方。

DOI: 10.6861/TANET.201810.0229



圖二：人工生成資料之正樣本及負樣本分布情況

4.2 實驗設定

三個模型的基礎分類器皆是採用 KNN、SVM、Decision tree 三種非常常見的分類器，KNN 採用 K = 5，SVM 的 kernel 採用 RBF kernel。

4.3 實驗結果

三個模型均使用相同的資料集與分類器 (KNN、SVM、Decision tree)。以下的準確率 (accuracy) 均為多次實驗後的平均值。

表一：三種基礎學習器(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在人工資料及上的準確率比較 (單位：%)

分類器	準確率
KNN	77.09
SVM	72.77
Decision tree	75.46
Majority Voting	77.64
Model 1	77.31
Model 2	77.62
Model 3	77.80

我們設計的三中動態模型在人工生成資料集上表現得均比每個基礎學習器好，然而與 Majority Voting 相比時，只有模型三略優於 Majority Voting。

5. 討論與未來發展

我們提出了三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：三種動態融合方式的預測效果優於基礎學習器，然而，三種動態融合方式中只有一種的預測效果略優於實驗比較的靜態多模型融合方法 Majority Voting。這顯示動態融合模型雖然可能是個可行的方向，但目前設計的動態融合方式仍嫌粗糙。

在實驗的過程中，我們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。我們也希望能找到有這樣特質的現實生活中的資料，這樣我們就可以更明確的確定與了解「動態」的多模型融合是不是真的有它厲害的地方。目前我們從兩個方向適圖得到更適合的資料集：(1) 設計更符合我們一開始初衷的人工生成的方法；(2) 尋找合適的公開資料集。另外，我們也計畫將動態融合模型與更先進的靜態融合模型 (如：XGBoost、LightGBM [12]、CatBoost [13] 等) 做比較，或將動態融合模型與這些較先進之靜態融合模型做整合。

致謝

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。

參考文獻

[1] Freund, Yoav, and Robert E. Schapire. "Experiments with a new boosting algorithm." In Icml, vol. 96, pp. 148-156. 1996.

[2] Dietterich, Thomas G. "Ensemble learning." The handbook of brain theory and neural networks 2 (2002): 110-125.

[3] Lin, Chih-Jen et al. "Feature Engineering and Classifier Ensemble for KDD Cup 2010."

[4] Niculescu-Mizil, Alexandru, Claudia Perlich, Grzegorz Swirszcz, Vikas Sindhwani, Yan Liu, Prem Melville, Dong Wang et al. "Winning the KDD cup orange challenge with ensemble selection." In Proceedings of the 2009 International Conference on KDD-Cup 2009-Volume 7, pp. 23-34. JMLR. org, 2009.

[5] Breiman, Leo. "Bagging predictors." Machine learning 24, no. 2 (1996): 123-140.

[6] Hastie, Trevor, and Robert Tibshirani. Generalized additive models. John Wiley & Sons, Inc., 1990.

[7] Freund, Yoav, and Robert E. Schapire. "A desicion-theoretic generalization of on-line learning and an application to boosting." In European conference on computational learning theory, pp. 23-37. Springer, Berlin, Heidelberg, 1995.

[8] Chen, Tianqi, and Carlos 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.

[9] Chen, Tianqi, Tong He, and Michael Benesty. "Xgboost: extreme gradient boosting." R package version 0.4-2 (2015): 1-4.

[10] Breiman, Leo. "Arcing classifier (with discussion and a rejoinder by the author)." The annals of statistics 26, no. 3 (1998): 801-849.

[11] Breiman, Leo. "Prediction games and arcing algorithms." Neural computation 11, no. 7 (1999): 1493-1517.

[12] Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "Lightgbm: A highly efficient gradient boosting decision tree." In Advances in Neural Information Processing Systems, pp. 3146-3154. 2017.

[13] Dorogush, Anna Veronika, Vasily Ershov, and Andrey Gulin. "CatBoost: gradient boosting with categorical features support." In Workshop on ML Systems at NIPS 2017.