

碩 士 論 文

不平衡資料集學習之少數類別過抽樣技 術的一個改良方法

An Improved Synthetic Minority Over-
sampling Technique for Imbalanced Data
Set Learning

系所別：資訊工程研究所

學號姓名：103062642 陳世承 (Shih-Cheng Chen)

指導教授：林華君博士 (Dr. Hwa-Chun Lin)

中華民國一百零六年七月

摘要

當資料集的少數類別實例有相對其他類別較少的實例數目時，則這樣的資料集可能隱含著類別不平衡的問題，也就是說訓練出的分類模型很可能因為少數類別實例發現機率較低的原因，而將少數類別實例錯誤判斷成多數類別實例。

透過人造少數類別資料實例以平衡多數類別以及少數類別之間的分佈不平衡是一種解決策略。有多種演算法已經依據此概念被設計出來。本研究提出一個改良的演算法 ISMOTE 來解決類別不平衡問題。ISMOTE 與以往演算法不同的地方是並非僅考慮少數類別的分布，而是同時衡量少數類別和多數類別在密度分布上的相對優勢，並以此作為權重衡量的基礎。另外，我們的方法會選擇以少數類別實例與距離其最近的多數類別實例作為參考實例產生人造實例。此作法可減少因為產生錯誤的人造資料實例而使分類器的學習更加地困難的狀況發生，並且透過此作法的人造實例能更好的幫助分類器的學習。

每一個少數類別實例具有一個分類器對於此資料實例困難學習的權重。權重公式的設計原則與此少數類別資料實例的困難學習程度呈成正比。因此 ISMOTE 可以針對每一個少數類別資料實例的權重，產生相對應數量的人造資料實例而逐漸改變分類決策的界線往較困難學習的方向。

這模擬實驗的結果證實了我們的方法與一些現存的方法是比較好或者可競爭的。透過使用 Recall、FP-rate、Precision、G-mean 以及較廣為人知的 AUC 測量指標(area under curve)和 F-measure 進行模型效能的測量。

Abstract

When a few categories of instances of a data set have fewer instances than other categories, such data sets may imply a problem of category imbalances, meaning that the trained classification model is likely to be found for a small number of instances Low cause, and a small number of instances of the wrong case to determine the majority of categories of examples.

It is a solution to the distribution of imbalances between the majority of categories and the few categories through the artificial minority category data examples. A variety of algorithms have been designed based on this concept. This study proposes a novel algorithm ISMOTE to solve the problem of class imbalance. ISMOTE differs from previous algorithms in that it does not take into account only a few categories of distributions, but rather measures the relative advantages of a few categories and most categories in density distributions as a basis for weighting. In addition, our approach will choose to produce artificial instances with a few category instances and most of the nearest category instances as a reference instance. This approach can reduce the situation where the classifier's learning is more difficult due to the generation of erroneous man-made data instances, and the artificial examples through this approach can better help the classifier to learn.

Each of the few category instances has a weight that the classifier has difficulty

studying for this data instance. The design principles of the formula are proportional to the degree of difficulty in learning with this few categories of data instances. So ISMOTE can be for each of a few categories of data instances of the weight, resulting in the corresponding number of examples of artificial data and gradually change the boundaries of classification decisions to more difficult to learn the direction.

致謝

這篇論文能夠順利，必須要感謝很多人。首先要感謝指導教授林華君博士，老師嚴謹的治學態度和對於研究的執著讓我這三年的研究生涯除了獲得更多的知識，也學習到從事研究的態度和方法，當然老師在我需要幫助的時候也給與適時的建議與指導，我才能有這一篇論文的產生。同時也要感謝學長、學弟以及學妹們，在苦悶的研究生生活之中陪伴我。

另外我由衷的感謝我的母親和父親，因為有你們無怨無悔的支持與照顧，我才能全心全意投入研究，並且順利完成學業，謝謝你們。我將繼續向前邁進。謹志於此，以申遠懷。

目錄

第一章 簡介.....	5
第二章 文獻探討.....	10
第三章 動機.....	24
第四張 ISMOTE.....	30
4.1 集合 S_{min} 以及集合 S_{majb} 的建構.....	33
4.2 取得權重 w 以及距離限制 d	35
4.3 人造實例的產生.....	44
第五章 實驗結果.....	48
5.1 效能指標.....	48
5.2 分類器的設定以及方法的參數設定.....	51

5.3 真實世界的資料集.....	52
5.3.1 Wilcoxon 符號等級檢定.....	53
5.3.2 成對樣本 T 檢定.....	57
5.4 討論.....	60
第六章 結論.....	63
參考文獻.....	65



第一章 簡介

在機器學習的領域中，我們對於分類問題的定義是建立一個系統並且使用監督式分類學習演算法去教導它能夠有系統地從既有的資料集中去學習出一項規則或是函數以便去預測未知(未見過)資料實例(instance)的類別。而且此規則或者函數必須要能夠有很好的推論普遍能力(generalization ability)。舉個例子，假設我們使用一個 X 來代表 k 筆資料實例: x_1, x_2, \dots, x_k 。這每一筆資料實例能透過它所擁有的 n 項特徵 (feature): a_1, a_2, \dots, a_n 來區別每一個特徵可能是連續值，

也可能是離散值，而且每一筆資料還會有一個類別標籤 (class label) $y_j \in C = \{c_1, c_2, \dots, c_c\}$ ，有了這樣的資訊後，我們能使用監督式分類學習演算法去建構一個系統，此系統將會遵循一個根據我們所具有的資料集合所找到的對映函數 $f: X \rightarrow C$ 進行預測，我們稱呼這樣的系統為分類器。

分類學習演算法通常會盡可能去尋找一個精確度最佳的對映函數，此函數若代入我們所具有的資料集合，將會有最小的錯誤比率或是最大的精確度。而類別的分布(某一個類別的資料數佔有整體資料數的比例)是分類學習演算法在處理分類問題上的一個很重要的關鍵。當一個類別在整體資料的比例中相對於其他類別是較於稀少的，換句話說，此少數類別可能不具有充分地代表性，這意味著預測模型可能會推論少數類別(minority)資料的發生頻率較低，因此若是使用精確度作為衡量表現的指標，則預測模型在做預測判斷時，是很有可能會預測少數類別(minority)資料為多數類別(majority)資料，舉一個例子，在一個類別傾斜比率程度為 1:99 的不平衡資料集(一筆資料為 positive class，九十九筆資料為 negative class)，一個學習演算法試圖去尋找一個能有著最高

$\text{Accuracy}(\frac{\text{判斷正確數}}{\text{全部}})$ 的判斷規則，並以此規則建立一個分類器，此分類器會忠實

依據此規則來做出預測判斷。而此分類器可能可以很輕易地得到 99% 的精確程度(只要它忽略 positive instance 並且預測所有的 instances 皆為 negative class)。

雖然此分類器在預測上有如此高的精確程度，但諷刺的是，分類器對於少數類別(minority)的正確判斷率遠遠的低於多數類別，而這樣的預測模型是矛盾的，

也因此我們會希望預測模型能夠有著最大化對於少數類別(minority)的辨識能力在盡可能不減少多數類別(majority)的判斷正確性上。此所提及的精確性高的矛盾，我們稱呼其為類別不平衡問題。

類別不平衡問題出現在真實世界的眾多角落，例如像是汙染偵測(pollution detection)[1]、風險管控(risk management)[2]甚至是醫療診治(medical diagnosis)[3]。在這些領域中，分類預測模型需要能夠準確的偵測出少數類別。否則將付出極其昂貴的代價。研究[4]說：「事實上在類別不平衡的資料集中學習如何判斷少數類別是一件很困難的事情」。因此解決類別不平衡問題將是研究者的一項重大的挑戰。

依據 paper[5][6][7][8][9][10]說:影響使用不平衡資料集所訓練出的分類模型效能有三項。

1. 第一項是訓練集樣本數過少(Small sample size)所導致的資訊缺乏(Lack of Information)，不充足的資料個數將增加學習演算法發現正確的分類規則的難度。依據研究[6]所說：「增加訓練集的實例個數，可降低不平衡分類模型的錯誤率」。
2. 第二項是類別重疊(Class overlapping)，少數類別資料實例分散在多數類別之中，依據[6]增加其分散的複雜度，會增加類別不平衡這項因素影響分類模型效能的影響。而研究[9]證明類別重疊與類別不平衡之間有一定程度的關聯，當類別重疊發生時，會使部分資料變的累贅或是無益於分

類器學習類別之間的界線(或是規則)

3. 第三項類別內分佈不均衡(Small disjoints)，當少數類別實例分布的群聚特性並不明顯，而導致少數類別資料實例在分布上可能同時有多個叢集。此種類別資料實例的在整個特徵空間上的分布常常會是不平衡的，此會造成部分叢集的少數類別資料實例不具備代表性，進而影響分類器的決策正確能力。研究[10]宣稱:我們可以透過對少數類別資料使用取出後放回抽樣的方式改變這樣不具代表性的少數類別叢集，以提升分類器的效能。

抽樣是一種很常見用來處理不平衡學習問題的策略，透過改變訓練集的資料分布來消除或是降低資料的不平衡特性。Under-sampling 和 Over-sampling 是抽樣這策略的基本呈現模式。Under-sampling 透過消除原始資料集合的資料實例(通常是多數類別的資料實例)來創造一個資料子集合。最簡單的實作方式是刪除部分多數類別樣本數來平衡資料的類別比率，但此方法的缺點是可能會刪除重要的資訊(樣本)，而導致分類器無法學習真實的分類規則。而 Over-sampling 則藉由替換或是創造資料實例來建立一個資料子集合來達到資料類別的平衡，最簡單的實作方式是對少數類別取出後放回的重複抽樣。但此方法的缺點除了會增加訓練集合的大小外，還可能會增加過度擬和(over-fitting)發生的可能性。另外依據研究[16]所說，在較複雜資料的處理上，Over-sampling 會有比 Under-

sampling 還要好的表現，因此我們以 Over-sampling 的方法作為我們主要的研究方向。

Chawla et al 在 2002 年時，提出 SMOTE 演算法[15]作為解決 Over-sampling 方法會有過度擬和問題的解決方法。SMOTE 演算法[15]透過人造方式產生任意倍數少數類別資料實例進而改善分類器對於少數類別實例的偏差(bias)。研究[11]說: SMOTE 演算法[15]以及與使用其核心概念衍伸的演算法[11][12][13][14]是現今解決不平衡學習問題較為流行的方法。在這篇研究，我們描述了包含 SMOTE 演算法在內的這些方法，在一些我們假定的情況下，可能會製造出錯誤或是不合邏輯的人造資料實例，或者產生的人造資料實例對分類器辨認少數類別上並沒有太大的幫助。這些情況使得相關演算法失去其幫助分類器更加容易學習的效果。在這一方面，我們提出一個改進的人造資料實例 over-sampling 方法。i.e. Improved Synthetic Minority Over-Sampling Technique (ISMOTE)，此方法的最終目標與其他人造實例方法相同皆是減緩分類器對於不平衡學習問題的困難。為了要達到這個目標，因此我們的方法將會去改善下列兩個階段，第一項階段是選取參考樣本的機制，第二項階段是產生人造實例的機制。

這研究剩下的章節如下。第二章將描述人造實例相關的演算法。第三章將呈現我們對於人造實例方法的一些觀察。第四章將敘述 ISMOTE 演算法的作法。各項實驗模擬的結果則將被詳述在第五章。最終，在第六章陳述我們的結論。



第二章 文獻探討

研究[5] [21] [26][29]說：解決類別不平衡問題的方法依照其解決問題的方式可被廣泛的區分成兩種，一種是演算法層級(Algorithm level)，另一種則是資料層級(Data level)。

演算法層級的實現主要的目標是透過調整那些描述少數類別的假說(這些假說通常比描述多數類別的假說要虛弱)，以此方式找出較為適當的歸納規則。在

演算法層級的實現方法中，Boosting 演算法[27]是廣為人知的一種演算法，Boosting 演算法主要的概念是透過增加錯誤分類實例的權重以降低類別不平衡學習的偏差(bias)。而另外一種常見的實現是隨機森林(Random forests)[28]，隨機森林演算法主要的概念是透過創造複數且彼此決策獨立的少數類別分類器投票以降低學習上的偏差。

資料層級的實現則是透過若干方式將類別不平衡的資料集，修正為類別平衡的資料集。由於資料層級的實現可應用在不同的分類學習演算法，因此資料層級的實現將會比演算法的實現還要具有彈性。資料層級的實現有兩種主要的策略，一種叫做 over-sampling[30]，另外一種叫做 under-sampling[31]。over-sampling 的想法是增加少數類別實例的數量，反之 under-sampling 則減少多數類別實例的數量，兩種策略的目標皆是為了要達到資料集類別分布的平衡。在此兩種策略的權衡上，研究[26][30]說：一般來說，當資料數量已經充分到能夠衡量資料特徵的時候，我們會傾向選擇使用 under-sampling 策略的演算法去降低分類演算法學習的時間，而當資料數量並不足以代表資料特徵，則會使用 over-sampling 策略的演算法去增加分類演算法在學習上的正確率。

研究[32][33]強調 over-sampling 策略的演算法是非常簡單但卻是在解決類別不平衡問題上，即使與其他策略比較也仍然非常具有競爭力的方法。但使用

over-sampling 策略的演算法皆會面臨一項共通的難題，過度擬和(over-fitting)問題。因此為了減緩這項問題，透過製造人造資料實例取代取出後放回抽樣的方式以達到類別資料分布平衡的方法大為盛行，像是[4][11-15][17-20][26][29][34-37]。

目前最常見的少數類別 over-sampling 方法是 SMOTE 演算法[15]。

SMOTE 演算法-2002

SMOTE 演算法[15]是解決不平衡問題的一種廣為人知的 over-sampling 方法，它藉由人為創造的實例而非取出後放回的動作來減緩分類器對於類別不平衡學習上的問題。人造實例的分布位置可能在少數類別實例與距離其最近 K 個鄰居實例中的任一條線段上的任何一個位置。SMOTE 的做法描述如下：

1. 依序選取每一個少數類別實例 i
2. 找出此少數類別實例 i 的最近 K 個鄰居實例
3. 隨機從 K 個鄰居實例中挑選一個鄰居實例
4. 將少數類別實例減去鄰居實例以得到一個向量
5. 隨機產生一個範圍 $0 \sim 1$ 的數字並與向量作相乘的動作以縮放該向量
6. 將少數類別實例加上該向量即可得到新的人造實例
7. 依據使用者設定的數量 N ，重複動作 2 到動作 6 共 N 次，才回到動作 1

研究[15]說: SMOTE 有效率地使得少數類別的決策區域變的比較具有概括化的

能力(generalization)，意即在整個特徵空間之中，分類器判別該區域是少數類別區域而非多數類別區域的部分增加。

研究[21]說 SMOTE 仍然有一些缺點，因此後來學者對於 SMOTE 的延伸改進，共可分為四種類型。

(1) 第一類 SMOTE 延伸變形：在使用 SMOTE 演算法之前或之後的前處理或是後置處理，像是研究[40]提出的 SMOTE+Tomek 以及 SMOTE+ENN 兩種方法，或是研究[41]提出的 SMOTE+RSB 或者是研究[42]的 FRIPS(Fuzzy Rough Imbalanced Prototype Selection)抑或是[29]的 SMOTE+IPF。

(2) 第二類 SMOTE 延伸變形：透過限制用來產生人造實例的少數類別實例範圍，以達到人造實例的產生位置會出現在被認定較有用的區域。較有用的區域是沒有絕對的定義的，有數種策略針對這個問題在發展之中，但其中被使用最廣泛的策略為找出坐落在類別之間邊界區域的實例並定義其為分類器可能較困難學習的實例。

(3) 第三類 SMOTE 延伸變形：透過某種方式量化每一個少數類別實例的困難學習程度，依據此困難學習程度的大小決定要在此少數類別實例附近產生多大數量的人造實例。

(4) 第四類 SMOTE 延伸變形：第四類 SMOTE 變形是透過某種規範影響產生人造實例的方式或者人造實例產生的位置，以達到產生較有價值的人造實

例。舉個例子，所產生的人造實例位置可能透過某種方式衡量而比較靠近或是遠離用來產生人造實例的少數類別實例。

由於第一類 SMOTE 延伸變形與我們的方法較無關西，因此我們將不會像第二類以及第三類變形那樣另作介紹。

SMOTE 延伸改進的第二類類型專注於尋求可以找出分類器可能較困難學習的實例的方式。分類器可能較困難學習的實例通常會被設定為位處於類別之間附近區域的邊界實例。以下列出對於 SMOTE 的延伸改進第二類類型較常見的方法：

Borderline-SMOTE 演算法-2005

Borderline-SMOTE 演算法[14]非常相似於 SMOTE 演算法，但 Borderline-SMOTE 演算法僅會替位置分布於較邊界的實例產生人造實例。或者根據研究[18]，我們可以更精確地說，Borderline-SMOTE 只針對分布於類別之間的少數類別執行 over-sampling 的動作。也就是說 Borderline-SMOTE 的基本想法是，若是分類器能夠正確認知到少數類別邊界區域，那麼其餘的少數類別資料實例，應該也能被分類器正確地辨認。依據研究[19]所描述 Borderline-SMOTE 演算法作法如下，首先對訓練集中每一個少數類別資料實例尋找它的最近 k 位鄰居實例。以及定義其多數類別鄰居數為 m 。若是 m 相等於 k ，則判別此少數類別實例為雜訊實例(noise)。若是 $k/2 \leq m$

$< k$ ，則代表此少數類別實例的多數類別鄰居實例數大於少數類別鄰居實例數，也因此可以合理的懷疑其較容易被分類器判別錯誤為多數類別實例，所以 Borderline-SMOTE 會將此資料實例加入名為危險集(DANGER SET)的集合內。若是 $0 \leq m < k/2$ ，則判別此資料實例是相對安全的。在判別每一個少數類別資料實例是否危險後，Borderline-SMOTE 的危險集資料實例由位處於類別邊界區域的少數類別實例組成。最後將危險集內每一個資料實例，與其最近 k 位鄰居實例中的少數類別鄰居實例，執行 SMOTE 演算法產生人造資料實例的步驟。依據研究[17][20]所說，在有類別重疊(Class overlapping)以及雜訊實例(Noise)存在的情況下，Borderline-SMOTE 產生的人造資料實例有可能位處於不適當的位置。研究[21]更是直述其實驗，在類別比一比九十九的極端不平衡資料集下，Borderline-SMOTE 並沒有明顯地展現比 SMOTE 優越的效能。另外研究[11]說 Borderline-SMOT 並沒有考慮少數類別與 K 位鄰居的分布情形是否密集，因此 K 位鄰居參數的設定，對於 Borderline-SMOTE 的影響巨大，錯誤設定參數 K 甚至有可能使其失敗執行。例如 K 參數過小，而導致應該位處於邊界區域的少數類別實例的 K 位鄰居皆是少數類別實例，因此 Borderline-SMOTE 將會因為危險集沒有任何成員，而無法運作。所以我們認為 Borderline-SMOTE 僅考量少數類別實例是否位處於多數類別實例的附近而決定其是否是分類器難以判斷的資料實例這點上是尚不夠細緻的。且由於其僅判斷資料實例的鄰居個數是否多餘 $k/2$

個多數類別資料實例以評估是否加入危險集，所以並沒有試圖去量化在危險集合內的各個資料實例的困難學習程度。

MSMOTE 演算法-2009

MSMOTE 演算法[12]是基於改善 SMOTE 演算法的變形版本，MSMOTE 演算法將少數類別實例區分成三種類型，安全實例(safe)、邊界實例(border)以及雜訊實例(latent noise)。MSMOTE 演算法對於雜訊實例，會將其刪除，而對於安全實例，MSMOTE 演算法則藉由 KNN 演算法隨機選擇一個資料實例做內插方法以產生新的人造資料實例(同 SMOTE 演算法)，至於邊界實例(border)，MSMOTE 演算法則會選擇最近的少數類別鄰居實例作為內插方法的參考點。MSMOTE 演算法的基本想法是安全實例越多則越能夠增加分類器的效能，反之雜訊實例越多則會降低分類器的精確度，而邊界實例是分類器，較難以學習的部分，因此選擇最近的鄰居實例(最保險的做法)，此作法可減少在類別重疊(overlapping)的情形下產生不適當位置的人造實例的可能。例如研究[11]所提出的一種狀況，當分佈不均衡(Small disjuncts)以及類別重疊(overlapping)同時發生，且兩群少數類別資料實例距離夠靠近，但兩群少數實例之間被多數類別實例所阻斷，當鄰居參數 k 選擇足夠大的時候，有可能造成兩群之中分別選擇一個資料實例做內插方法，而導致可能產生人造實例分布在多數類別區域。研究[11]認為這是使用 KNN

演算法作為判斷少數類別實例是否為邊界實例的缺點。

DBSMOTE 演算法-2012

DBSMOTE 演算法[45]結合了 DBSCAN[47]以及 SMOTE[15]兩種演算法，並且藉由 DBSCAN 的概念定義了一個新的資料結構為 Directly density-reachable graph，研究[45]認為這個資料結構可以將資料實例區分成邊界實例以及核心實例。DBSMOTE 演算法再製造人造實例前必須先透過 DBSCAN 演算法將訓練集合 D 中的所有少數類別實例分成數群 C_i ，並且對每一群資料實例使用 Directly density-reachable graph 去描述，找到距離 C_i 最近的資料實例(來源點 c)，再使用 Dijkstra 演算法找出來源點 c 到其他資料實例的距離。在完成上述動作後，每一個少數類別實例皆可找到其到來源點 c 的最短路徑，從中此最短路徑隨機抽選一邊，即可找到兩點作為使用內插法所需要的參考實例。每一群 C_i 皆依據上述動作製造人造實例集合 D_i ，最後將所有的 D_i 與 D 合併，即可透過此新的訓練集合 D 建立預測模型。與其他演算法在概念上相異的點是 DBSMOTE 認為邊界實例是較無資訊價值的，因此其產生的人造實例會比較靠近核心實例而非邊界實例。

SMOTE 延伸改進的第三類類型專注於尋找量化分類器可能較困難學習實例的困難程度的方式。依據此困難學習程度的大小決定要在此少數類別實例附近產生多大數量的人造實例。以下列出對於 SMOTE 的延伸改進第三類類型較常

見的方法:

ADASYN 演算法-2008

經由觀察研究[11-23]，位處於靠近邊界的少數類別實例應該是相對於其他少數類別實例較困難學習的。此觀念似乎已經成為一種共識。相較於 Borderline-SMOTE 試圖去辨識哪一個實例是困難學習的，ADASYN 演算法 [13]的想法更有企圖，其企圖量化資料實例學習困難的程度。也因此 ADASYN 演算法可以使分類器在學習時更容易專注在這些辨識較為困難的實例上。ADASYN 演算法與 SMOTE 演算法相同，也使用內插作為其製造少數類別實例的方法，但不同的是在 SMOTE 演算法中，其參考的每一個少數類別實例所產生的資料實例數量是相同的，而 ADASYN 演算法則有可能不同。也就是說 SMOTE 演算法假設資料是均勻分布的，但 ADASYN 不接受這個假設，所以透過計算每一個少數類別最近 K 位鄰居中的多數類別鄰居數 r_i ，並且將其正規化作為每一個少數類別實例產生的人造實例權重，此權重 \hat{r}_i 代表。這是一種密度分布(density distribution)，其定義為每一個少數類別的最近 k 位鄰居的多數鄰居的正規化數字(normalized number)。因為 ADASYN 設定產生的人造實例總數 s 為多數類別實例數與少數類別實例數的差，所以只需將 s 透過每一個少數類別的權重 \hat{r}_i 分配給每一個少數類別實例，則 ADASYN 可以為每一個少數類別實例自動產生其對應數目的人造實

例。依據 ADASYN 演算法的做法，每一個少數類別實例的鄰居實例中，若是存在越多的多數類別實例，則 ADASYN 將會分配給他較多的人造資料實例數，所以研究[23]說因為雜訊實例(noise)在 ADASYN 中將會得到最大的權重，因此對於分類器的負面影響也會成倍增加。雖然我們認為量化實例困難程度並以此作為分類器學習的依據是一個很好的想法，但這若是考慮不周的話，則很可能會將錯誤也放大數倍。另外 ADASYN 演算法其量化危險程度的方式僅僅考慮少數類別實例的多數類別鄰居數，並沒有考量到資料實例分布的狀況，所以當 k 鄰居參數設定過小時，也可能會發生類似於 Borderline-SMOTE 無法執行的問題，而當 k 鄰居參數設定過大時，因為 ADASYN 並沒有考量少數類別實例與其 k 位鄰居實例位置分布和組成的緣故，所以在類別重疊的情況下，很可能所產生的人造資料實例的位置是不恰當，或者不符合邏輯的，而此錯誤層級也可能會因為 r_i 而放大。

MWMOTE 同時具有 SMOTE 延伸改進的第二類類型以及第三類類型不只試圖量化分類器可能較困難學習實例的困難程度的方式，同時也決定要在哪些少數類別實例附近產生人造實例，並依據其困難學習程度的大小決定要在此少數類別實例附近產生多大數量的人造實例。

MWMOTE 演算法-2014

MWMOTE 演算法[11]透過產生有用的少數類別實例以減緩資料類別不

平衡問題所造成的分類器學習不佳的程度一直是使用內插方法產生資料實例研究所關注的重點[11-18]。差別只在於相關研究如何定義何謂有用的人造實例以及該演算法對於各種情況的考慮是否周全。MWMOTE 演算法[11]基本想法是他們認為應該要考量少數類別實例的分布位置以及所分布的密度做為在產生人造實例時，評估少數類別實例困難程度的依據。MWMOTE 演算法的做法是在一開始會先試圖去尋找困難學習的少數類別實例，以及指派給他們一個權重(基於他們與所有邊界多數類別實例的歐式距離)並在完成所有指派後，將權重正規化為機率。然後將所有少數類別實例分群。再對每一個困難學習的實例抽樣(其被抽到的機率與其所擁有的權重機率相同)。當抽樣出一個困難學習實例後，再從其所在的群體內隨機挑選一個少數類別實例，即可透過內插方法新增一個人造實例。而增加的實例數目由使用者自訂的參數 N 決定。雖然 MWMOTE 透過叢集演算法考量的少數類別實例分布的密度以及範圍，但我們認為其或許可以更進一步考量少數類別實例分布的密度相對於與其鄰近邊界區域的多數類別實例分布的密度。另外 MWMOTE 在考量少數類別實例困難學習程度時是考量了與所有邊界多數類別實例的距離，但我們認為邏輯上，應該只需要考量與其面對的邊界區域多數類別實例即可。

SMOTE 的延伸改進第四類類型將重心直接放在對於產生人造實例的規範限

制上，舉個例子，所產生的人造實例位置可能透過某種方式衡量而比較靠近或是遠離用來產生人造實例的少數類別實例。以下列出對於 SMOTE 的延伸改進第四類類型較常見的方法：

Safe Level-SMOTE 演算法-2009

與 SMOTE 演算法隨機地在每一個少數類別實例之間產生少數類別人造實例相比，Safe Level-SMOTE 演算法[43]是較為謹慎的透過衡量少數類別實例的安全程度以及其鄰居實例的安全程度以決定產生人造實例是較靠近此少數類別實例還是其鄰居實例。研究[46]說：「Safe Level-SMOTE 演算法最廣為人知的概念是安全等級(safe level)的比較」，安全等級數是指一個實例的少數類別鄰居數。如果一個少數類別實例的安全等級數是比較逼近於零，則代表此少數類別實例是比較有可能是雜訊實例(noise)，如果一個少數類別實例的安全等級數是比較逼近於使用者設定的 K 鄰居數，則代表參考此少數類別實例是比較有可能較為安全的。依據此概念，Safe Level-SMOTE 演算法在產生人造實例之前會先計算每一個少數類別實例的安全等級，對每一個少數類別實例隨機選擇一個鄰居實例，以此兩個少數類別實例作為人造實例的參考以使用內插方法產生人造實例，每一個少數人造實例的位置會依據兩個少數類別實例的安全等級大小，依照比例比較靠近具有較大安全等級數的少數類別實例，因此每一個人造實例將會被產生在比較安全的區域。Safe Level-SMOTE 是 SMOTE 演算法的一種變形，其差異在於產生人造實例時

的分布位置，會參考安全等級數，而非隨機分布。

Local Neighborhood Extension of SMOTE 演算法-2011

LN-SMOTE 演算法[44]以及 Safe Level-SMOTE 演算法所產生的人造實例分布會環繞在少數類別實例以及此少數類別實例的鄰居實例之間的區域，且此人造實例的位置比較靠近安全等級較高的類別實例。LN-SMOTE 演算法是 Safe Level-SMOTE 演算法的延伸改進。Safe Level-SMOTE 的策略是尚不完善的，例如當所選擇的少數類別實例為雜訊實例時(其最近 K 位鄰居皆為多數類別實例)，則此雜訊實例的鄰居實例必定為多數類別實例，因此會造成所產生的人造實例位置向雜訊實例無限地靠近。LN-SMOTE 的解決方法是透過修改對於擁有多數類別鄰居實例時的安全等級計算公式，因此 LN-SMOTE 並不會對於此雜訊實例產生人造實例，而將錯誤放大。

另外除了 SMOTE 以及其延伸的演算法之外，研究[35]使用高斯分布去達成隨機漫步的實現，是一種新奇的做法。其描述如下：

PDFOS 演算法-2014

研究[35]提出一個新奇的 over-sampling 方法，這項演算法不同於常見的以 SMOTE 演算法為基礎去作出任何步驟上的改變或者增加一項針對於 SMOTE 演算法的前處理或是後置處理，而是依據中央即限定理透過機率密度函數的估計製造新的人造資料實例以解決類別不平衡問題。研究[35]認為

改變類別資料的分布狀態對於後續學習演算法的訓練會有負面影響，所以反對像是研究[14]等改變類別資料實例分布的 over-sampling 方法，其假定少數類別資料實例將服從於常態分佈以及中央極限定理，因此可以透過訓練集中少數類別資料實例的平均值以及標準差推論母體的平均值以及標準差，也就是說其選定一個點作為基準點之後，再透過常態分佈機率去推論產生新產生的人造資料實例應該在基準點附近中較可能出現的位置。這方法的優點是擁有比 SMOTE 演算法更好的概括化能力(generalization)，因為 SMOTE 演算法產生的資料實例在少數類別實例與它的 K 位鄰居實例之一當中的線段上。

但此方法有一個的缺點，我們認為若是少數類別實例數目過少，則平均值以及標準差推論回母體的部分可能會有問題，抑或是少數類別實例其實並不服從於常態分佈，則使用此演算法會產生錯誤的資料實例進而嚴重改變整個類別實例的分布。



第三章 動機

產生人造實例的方法已經被證實是解決解決類別不平衡問題非常有用的方法之一，然而我們的研究還找到了一些現存方法[11,13-15]還尚不充足或者不適當的部分，這些部分可能會發生在不同的情況，因此我們將它描述成一個章節。

透過限制產生人造實例的少數類別實例範圍來產生人造實例的方法目前有 k-NN based 以及 clustering based 兩種方式。k-NN based 的方法透過衡量少數類別實例附近的最近 K 個實例中的資訊來判斷應該做出何種處理動作，人造實例

的範圍會在少數類別實例以及其最近 K 個實例其中一個的線段之間。使用內插方法產生人造實例需要選擇兩個實例才能產生人造實例，而 clustering based 的方法會先判斷少數類別彼此之間是否屬於同一集團以得到集團的資訊，因此 clustering based 的方法能夠輕易的選擇同一個集團的兩個實例執行內插方法，如此，可避免選擇到分屬不同集團的兩個實例產生人造實例。

在 k -NN based 方面，我們以 Borderline-SMOTE[14]以及 ADASYN [13]兩種方法作為我們描述 k -NN based 的方法如何使得這兩個方法失去作用的例子，在 clustering based 方面，我們則以 MWMOTE [11]作為我們描述 clustering based 的方法失效的範例。

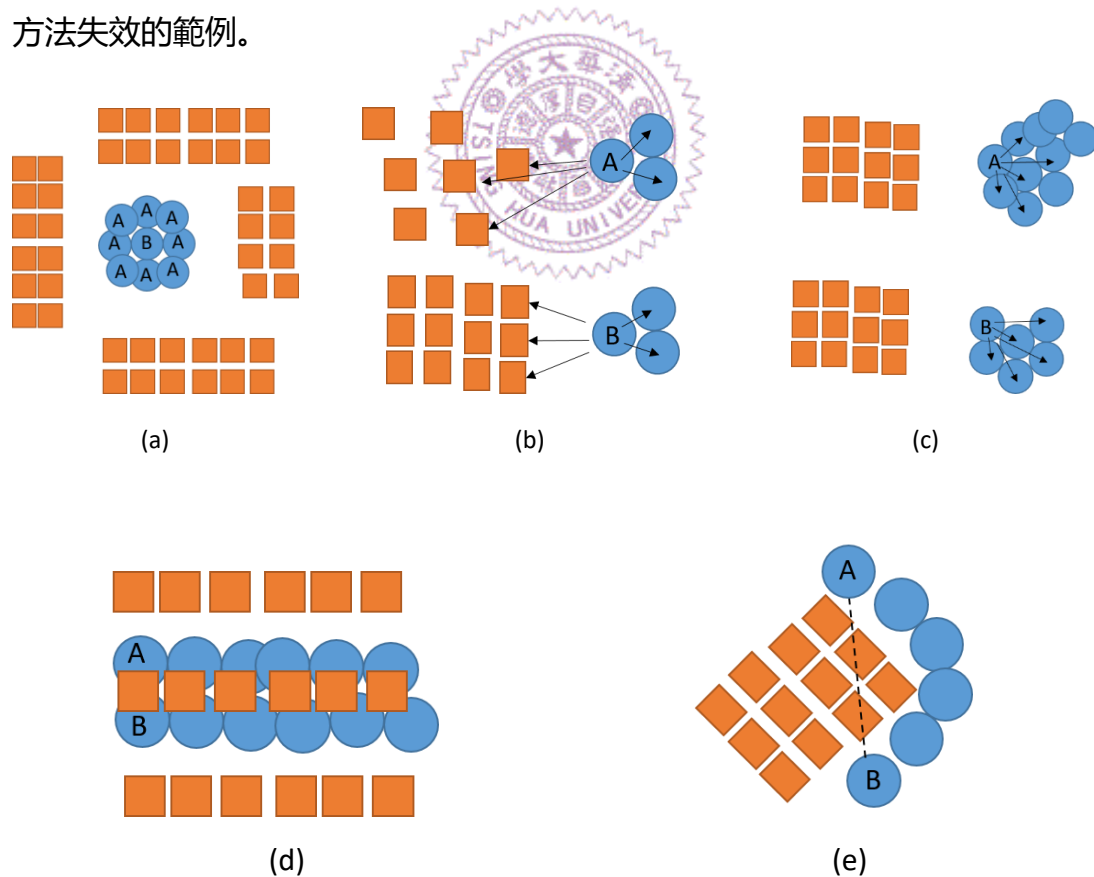


Fig. 1. (a)所有的點 A 是少數類別的外層實例. (b)點 A 與點 B 分屬不同集團且點 A 與點 B 的最近 $K=5$ 個鄰居皆有三個多數類別. (c) 點 A 所屬的集團比點 B 所屬的集團為大且點 A 與點 B 的最近 $K=5$ 個鄰居皆是少數類別. (d)少數類別實例分布相當靠近,但在其中夾雜著多數類別實例(e)

少數類別實例分布形狀並非為橢圓形

Borderline-SMOTE[14]認為邊界區域的少數類別實例是對於分類器來說，是比較困難學習的，因此其試圖去辨識並找出這些較困難學習的少數類別實例(稱為危險集)，Borderline-SMOTE 使用危險集作為它產生人造實例的參考樣本(seed samples)，因為這是最有可能被分類器分類錯誤的實例，而這也是為什麼 Borderline-SMOTE 產生的人造實例皆位處於邊界區域。依據[14]，少數類別實例是否危險是依據其最近 K 位鄰居中多數類別的個數，若多數類別鄰居數 m 滿足下列等式。

$$\frac{K}{2} \leq m < K$$

則代表此少數類別實例為危險集成員， K 是使用者自訂的參數。但

Borderline-SMOTE 可能會無法正確辨識出少數類別的邊界實例，Fig. 1a 描繪出了這樣的情況(圓圈代表少數類別實例，正方形代表多數類別實例)，假設使用者定義的最近鄰居數 $K=5$ ，則每一個少數類別實例 A 以及 B 在其最近 K 位鄰居實例中皆沒有任何的多數類別鄰居實例，所以下列式子並不成立

$$\frac{5}{2} \leq 0 < 5$$

因此所有的實例 A 以及實例 B 皆不會被判斷為邊界區域，這最終的結果是沒有人造實例將產生在實例 A 或者實例 B 附近。此結果代表 Borderline-SMOTE 可能會在 K 參數設定過小的情況下無法正確的判斷邊界實例，造成分類器學習上仍然是困難的。但 K 參數的正確設定是非常困難的，因為並不是所有的少數類

別邊界區域都能夠使用相同的 K 值。以 Fig. 1c 為例，實例 A 以及實例 B 分屬不同的集團，若是設定 $K=5$ ，則實例 A、B 皆不會有多數類別鄰居，因此 Borderline-SMOTE 此時對於此兩實例皆無法判斷是否為邊界實例，然而繼續將 K 加大，則實例 B 會比實例 A 較快有多數類別鄰居，繼續將 K 加大後，實例 A 也將具有多數類別鄰居，然而若是將 K 設定過大，則實例 B 的最近 K 位鄰居可能會具有實例 A 所屬集團的少數類別實例，這少數類別實例會佔據實例 B 最近 K 位鄰居中的一個位置，此種少數類別實例若是過多，則有可能會影響到實例 B 被選為危險集成員，而導致 Borderline-SMOTE 無法正確判斷實例 B。

ADASYN [13] 試圖透過給予每一個少數類別實例一個重要性權重 w_i 去避免上述的問題，一個實例若是具有比較大的 w_i 則產生較多的人造實例，反之則相反。權重 w_i 的計算是依據此少數類別實例的最近 K 位鄰居中的多數類別實例個數，多數類別實例個數越多則 w_i 越大。然而以此方式指派 w_i 的數值會有下列的問題。

1. w_i 的衡量數值可能是不適當的。Fig. 1a 能很明顯的表現這個情況，所有的少數類別實例 A 皆沒有任何的多數類別鄰居實例（假設 $K=5$ ）因此它們的 w_i 皆將會是零，這代表在 ADASYN 中所有的實例 A 都將被給予一個相同且非常低的權重，雖然它們似乎應該是比較重要的實例，應該要給予較高的權重才比較合理，因為它們所在的位置是比較靠近決策邊界

的。與此相比較，Fig. 1a 的實例 B 的權重竟然也是零，其權重與所有的實例 A 的權重相同，但以位置的分布來看，實例 B 的權重應該要比實例 A 還低才比較合理。當然，上述的問題可能可以透過增加 K 值的大小來避免。在一些少數類別實例的位置比較靠近多數類別實例的邊界區域，設定較小的 K 就足以包含多數類別實例在最近 K 位鄰居之中，但在另一些少數類別實例的分布上是比較緊密的邊界區域，則需要設定較大的 K 值，因此 K 值的正確設定是非常困難的，且並不是所有的少數類別邊界區域都能夠使用相同的 K 值。

2. w_i 的資訊可能不足以用來衡量少數類別實例的重要性。Fig. 1b 以及 Fig. 1c 能很明顯的表現這個情況，Fig. 1b 的少數類別實例 A 以及少數類別實例 B 其最近 K 位鄰居中皆只有三位多數類別鄰居實例(假設 $K=5$)，因此兩者的權重 w_i 皆等於 3，但很明顯實例 B 對於學習上的困難程度是比實例 A 還要高的，因為其面對的多數類別實例個數較多。另外 Fig. 1c 的少數類別實例 A 以及少數類別實例 B 其最近 K 位鄰居中皆為少數類別鄰居實例(假設 $K=5$)，但很明顯實例 B 對於學習上的困難程度是比實例 A 還要高的，因為其四周的少數類別實例個數較少。

MWMOTE [11] 透過分群演算法去找出少數類別實例的集團分布。並且考慮少數類別實例與多數類別實例之間距離上的問題，去衡量每一筆少數類別實例

的權重，權重越大，則代表被抽中作為參考實例的機率越高，當某一個少數類別實例被選為參考實例時，會再從其所屬的集團之中隨機選擇一個少數類別實例，以此二實例執行內插方法產生人造實例。但採用分群演算法會有下列問題

1. 直接採用分群演算法可能會分群錯誤。分群演算法是一種非監督式學習的演算法，因此其並不會考慮多數類別實例以及少數類別實例之間的差異。Fig. 1d 能很明顯的表現這個情況，被少數類別實例夾雜其中的六個多數類別實例，因為其與少數類別實例相對較近，因此會被判斷為與少數類別實例是同一群，若是同一群，則實例 A 有可能跨過中間的多數類別與同一群的實例 B 執行內插方法產生人造實例，而此人造實例將會座落在多數類別實例的附近，而這是不合理的。同理，若是只放入少數類別實例作分群演算法，則仍然有可能會因為並沒有認知到多數類別實例的存在而發生上述的問題。
2. 隨機挑選同一群的實例是有可能會產生錯誤的人造實例。由於分群演算法是以彼此之間的距離是否夠靠近作為是否同一群的依據，因此同一群的少數類別實例的分布形狀並不一定是呈現凸集合(convex set)的分布，而這可能會產生問題。Fig. 1e 能很明顯的表現這個情況，少數類別實例 A 以及少數類別實例 B 將被執行內插方法，因此其產生的人造實例將位處於多數類別實例的分布範圍，而這是不合理的。

由 Fig (a)(b)(c)，我們可以知道使用 k-NN based 的方式產生人造實例的方法在一些情況之下很有可能是很不適合的。E.g. 參考的少數類別實例其所在的分布是比較密集的，或者是參考的少數類別實例其所屬的集團是比較小的。這些方法會產生上述問題的原因是它們盲目地認為不需要考慮少數類別實例與其最近 K 位鄰居實例之間的距離以及其所在的位置分布，另外如何設定適當的 K 值也是 k-NN based 方法的一大問題。

由 Fig (d)(e)，我們可以知道使用 clustering based 的方式產生人造實例的方法在一些情況之下很有可能是很不適合的。這方法會產生上述問題的原因是它盲目地認為透過分群演算法可以絕對正確地找出少數類別集團的分布，或是在找出集團的分布之後，並不用採取一些特別的規則去規範所產生的人造實例。

第四張 ISMOTE

為了解決我們在第三章所描述的問題，我們提出一個改良的人造產生少數類別實例的演算法(oversampling method)ISMOTE。這演算法的主要目標有兩

個，改善選取參考樣本的機制以及改善產生人造實例的機制。我們的 ISMOTE 包含三個關鍵的步驟。第一階段，ISMOTE 從原始的少數類別集合中辨認所有的雜訊實例(noise)，並且建立一個排除了雜訊實例的少數類別集合 S_{min} ，再以此集合去找出邊界多數類別集合 S_{maj}^b 。第二階段， S_{min} 內每一個成員將被指派一個權重 w_i 以及距離限制 d_i ， w_i 的大小則與其在資料中的重要程度呈正相關， d_i 的大小則與參考成員所在的分布密度相關。第三階段，ISMOTE 使用 S_{min} 中每一個成員作為參考實例，每一個 $x_i \in S_{min}$ 都將與距離其最近的多數類別實例 $y_i \in S_{maj}^b$ 執行內插方法產生人造實例，產生人造實例的數量則與 $x_i \in S_{min}$ 的權重 w_i 呈正比。ISMOTE 的方法將被完整的呈現在[Algorithm 1]。步驟一到步驟四屬於第一階段然後步驟五到步驟七屬於第二階段，最後步驟八屬於第三階段。

Algorithm 1. ISMOTE (trainSet , K)

Input:

1. trainSet: 訓練集 T
2. K: 最近鄰居的個數。

Procedure Begin

1. 建立空集合 S_{min}
2. 尋找每一個少數類別實例 $x_i \in T$ 的最近 K 位類別鄰居實例 $\hat{N}(x_i)$ ，若 $\hat{N}(x_i)$ 內成員並非全是多數類別，則將 x_i 加入集合 S_{min} ，否則視其為雜訊(Noise)
3. 尋找每一個少數類別實例 $x_i \in S_{min}$ 的最近 K 位多數類別鄰居 $\hat{N}_{maj}(x_i)$
4. 對所有的 $x_i \in S_{min}$ 做 $\hat{N}_{maj}(x_i)$ 的聯集，找到位於邊界區域的多數類別集合 S_{maj}^b

$$S_{maj}^b = \bigcup_{x_i \in \hat{N}_{maj}(x_i)},$$

5. 以集合 S_{min} 為參考實例集(SEED SET)，在集合 S_{maj}^b 之內，尋找 S_{min} 內每一個實例 $x_i \in S_{min}$ 的最近多數類別實例 $y_i \in S_{maj}^b$ ，當 $x_i \in S_{min}$ 的最近多數類別實例 $y_i \in S_{maj}^b$ 被尋找到時，以此兩點之歐式距離作為半徑 r ，分別將 x_i 和 y_i 作為圓心點並以半徑 r 之距離畫圓，計算在以 x_i 作為圓心點之圓內的少數類別實例數(需要在集合 S_{min} 內)，此數目以 C_{min} 標示。計算在以 y_i 作為圓心點之圓內的多數類別實例數，此數目以 C_{maj} 標示
6. 給予每一個 $x_i \in S_{min}$ 一個重要性權重 w_i
$$w_i = \frac{C_{maj} + 1}{C_{min} + 1}$$
7. 給予每一個 $x_i \in S_{min}$ 一個距離限制 d_i
$$d_i = \frac{C_{maj} + 1}{C_{maj} + 1 + C_{min} + 1}$$
8. For each $x_i \in S_{min}$
 1. 計算 x_i 需要產生的人造實例數 $C_i = (\text{多數類別實例數} - \text{少數類別實例數}) * \frac{w_i}{\sum w_i}$
 2. 以 x_i 以及 x_i 的最近多數類別鄰居實例 y_i 兩點執行內插方法，產生人造少數類別實

例 newNode C_i 個。令 n 為資料集的特徵個數

For $i = 1$ to C_i

初始化 dif 為 n 維向量

For attr = 1 to n

$$dif[attr] = rand(1) * d_i * (y_i[attr] - x_i[attr])$$

End LOOP

$$newNode[i] = x_i + dif$$

End LOOP
3. 將產生的人造實例加入訓練集
End LOOP
End

4.1 集合 S_{min} 以及集合 S_{maj}^b 的建構

雜訊區域通常位於遠於少數類別實例的群集之外。分類器常會判斷這些雜訊實例為多數類別，因為這些雜訊實例被多數類別實例所環繞。安全區域通常位於少數類別實例的群集之內。分類器是較容易去辨認位於安全區域的實例的，因為它具有充足數量的實例數。然而對於類別不平衡問題來說，少數類別實例的安全區域很有可能是並不包含足夠分類器學習的少數類別實例數。因此產生人造實例的方法是應該要具備偵測雜訊區域的能力，才能僅針對在安全區域上的少數類別實例運作以及避免對在雜訊區域上的少數類別實例作用。另外對於人造實例方法來說，大部分的多數類別實例實際上是沒有用處的，較有用處的是較靠近少數類別實例的邊界多數類別實例，若有這些多數類別實例的資訊，可以幫助演算法節省許多計算上的時間。在 ISMOTE，我們透過建構集合 S_{maj}^b (去除雜訊實例的少數類別實例) 以及集合 S_{maj}^b (邊界多數類別實例) 這樣的作法以提供足夠的資訊產生人造實例。這整個 S_{min} ， S_{maj}^b 的描述如下：

1. 我們的 ISMOTE 首先從原始的少數類別實例集合中過濾掉雜訊(此處雜訊的定義為少數類別實例的最近 K 位鄰居實例皆為多數類別實例)，然後得到已經去除掉雜訊實例的少數類別實例集合 S_{min} 。為了做到這件事，我們必須先計算每一位少數類別實例與其他資料實例的歐

式距離，如此才能找到此少數類別實例的最近 K 位鄰居，然後再判斷這 K 位鄰居實例是否皆為多數類別實例，若是，則移除掉此少數類別實例。這個移除動作代表了雜訊實例參入人造資料實例產生過程是不可能發生的事情。如此，ISMOTE 將能夠移除訓練集中的雜訊實例且能夠避免因為雜訊實例的參入而產生的人造雜訊實例。

2. ISMOTE 將為 S_{min} 中的每一個少數類別實例 $x_i \in S_{min}$ 建立一個最近 K 位多數類別實例鄰居集合 $\hat{N}_{maj}(x_i)$ 。當 K 這個參數是被設定成相對小的時候， $\hat{N}_{maj}(x_i)$ 中的每一個多數類別實例將能夠被假設其分布的位置是比較靠近決策邊界區域的。然後我們將所有的 $\hat{N}_{maj}(x_i)$ 執行聯集的動作，以此方式就能夠得到多數類別的邊界實例集合 S_{maj}^b 。當少數類別實例彼此是非常的靠近或是其距離多數類別實例皆非常的遙遠，則所有的少數類別實例皆是有可能沒有多數鄰居類別實例的，此狀況造成了 Borderline-SMOTE [14] 或者是 ADASYN [13] 演算法無法達成其設計者希望它們達到的能力。但 ISMOTE 卻仍然可以良好運作地得到 S_{min} 以及 S_{maj}^b ，因為我們一開始便有考慮到這種可能性，因此我們不可能因為上述狀況而無法產生人造資料實例。

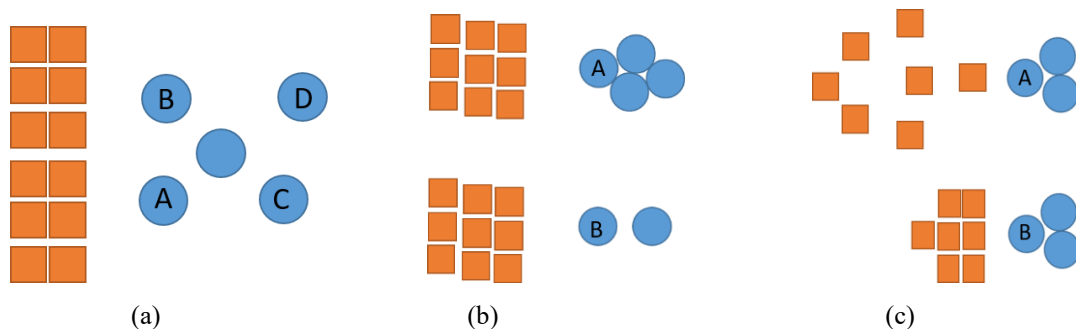



Fig. 2. (a)-(c)圖解了三個我們所觀察到的少數類別分布圖

4.2 取得權重 w 以及距離限制 d

在先前的章節，我們描述了 ISMOTE 如何獲取產生人造實例必要的資訊而將有價值的少數類別實例以及多數類別邊界實例分別建構成集合 S_{min} 以及 S_{maj}^b 。然而即使是我們認為具有價值的少數類別實例集合 S_{min} 內的成員，其具有價值的程度仍然不一定相同，一些實例有可能提供比其他實例所提供的訊息更有用的資訊。因此，替每一個少數類別實例依據他們的重要程度指派權重是必要的。一個實例若是具有比較大的權重則代表其需要產生較多的人造實例環繞在其四周。這是由於它提供的少數類別資訊是較不充分的。一些 oversampling 的方法(e.g., ADASYN [13]) 使用最近鄰居中的多數實例個數來判斷其少數類別實例的重要性，然而我們已經很清楚地在第三章描述了這個機制在很多的情況之下是不充分以及不適當的。所以我們的 ISMOTE 使用了新的機制去指派適當的參數給少數類別實例。在[Algorithm 1]的第五個到第七個步驟顯示了我們的 ISMOTE 會替每一個少數類別實例 $x_i \in S_{min}$ 計算其重要性權重 w_i 以及距離限制 d_i 。這權重 w_i 計算的概念是依據下列三項重要的觀察：

1. 第一項觀察：分布位置較靠近於決策邊界的資料實例比分布位置較遠於資料實例對於決策邊界的影響提供較多的資訊量。這項觀察暗示了應該給予那些分布位置較靠近決策邊界的資料實例相較於較遠的資料實例較高的權重係數。以 Fig. 2a. 為例來說，若少數類別實例 A 以及 B 分布在少數類別集團分布的外側相鄰多數類別實例而少數類別實例 C 以及 D 則坐落在少數類別集團分布的外側遠離多數類別實例，這代表實例 A 以及 B 的位置相對實例 C 以及實例 D 的位置是比較靠近決策邊界的。因此實例 A 以及實例 B 應該是比實例 C 和實例 D 還要具有資訊價值的。同理，實例 C 是比實例 D 還要具有資訊價值的。此狀況顯示實例 A 以及實例 B 應該被給予比實例 C 和實例 D 還要高的重要性權重才對，當然，實例 C 的權重應該也要高於實例 D 的權重才會比較合理。
2. 第二項觀察：少數類別實例與四周少數類別實例的分布狀況比較稀疏的將比與四周較稠密的少數類別實例還要重要。訓練集內少數類別實例的分布狀況有可能是不均勻的，從產生人造實例的觀點來看，四周分布比較稀疏的少數類別實例將較比較稠密的少數類別實例還要重要。這是因為其所在分布較稠密的實例比起分布較稀疏的實例內含較多的資訊。因此我們需要為了其四周分布較稀疏的少數類別實例製造較多的人造實例以增加其四周分布的密度來降低類別內的不平衡狀況。以 Fig. 2b 舉例，少數類別實例 B 是比少數類別實例 A 還要來的重要的，這理由很

明顯地是因為少數類別實例 A 是比較密集的集團成員之一而少數類別實例 B 則屬於比較鬆散的集團。

3. 第三項觀察：若是少數類別實例面對分布較稠密的多數類別實例，則其重要性將比面對多數實例分布較稀疏的少數類別實例還要來的重要。

以 Fig. 2c 為例，少數類別實例 A 以及少數類別實例 B 所屬於的集團大小相同且其面對最近多數類別實例的距離亦相同，然而少數類別實例 B 所面對的多數類別分布密度是比少數類別實例 A 還要高的。這分布密度上的不平衡將使得分類器對於少數類別實例 B 的學習是比較困難。

因此少數類別實例 B 在人造實例的產生上，應該是比少數類別實例 A 還要重要的，所以實例 B 的權重應該要高於實例 A 的權重才比較合理。

另外這距離限制 d_i 的概念是依據下列兩項重要的觀察，但在描述這些觀察之前，我們必須要先描述我們的 ISMOTE 將以少數類別實例與多數類別實例作為參考實例以產生內插方法。我們假設每一個少數類別最難學習的方位是距離其最近的多數類別實例所在的方向。一個很直覺的概念是若是在少數類別實例與其最近多數類別實例之間布置人造實例，則這些人造實例將可以很輕易的改變分類器對於少數類別認知的決策邊界，且人造實例分布的位置距離多數類別實例越靠近，則代表可能拓寬少數類別的決策邊界越多。以下是對於距離限制

d_i 的兩項重要的觀察。

1. 第一項觀察：我們依據上述權重第二項觀察的描述以及 Fig. 2b.可以發現一件事情，就是我們想產生人造實例的位置是應該經過限制的。依據少數類別實例 B 做為參考實例而產生的人造實例，其與人造實例之間的距離應該相較於少數類別實例 A 以及依據少數類別實例 A 做為參考實例而產生的人造實例之間的距離還要來的遠。這原因很簡單，因為實際上實例 B 所處位置的密度是比實例 A 的所處位置的密度還要來的鬆散的，所以我們不該與其他人造實例的演算法[12-15]一樣任意地假設實例 A 以及實例 B 所處區域皆是均勻分布的。
2. 第二項觀察：我們依據上述權重第三項觀察的描述以及 Fig. 2c.可以發現一件事情，就是當我們配置的人造實例將在少數類別實例與其最近多數類別實例之間時，則若是少數類別實例所面對的多數類別實例密度較高時，為了平衡少數類別實例所面對的學習壓力則或許應該給予少數類別實例較寬的決策邊界。以 Fig. 2c.為例，在實例 A 與實例 B 的狀態均相同，其差異就只有各自面對的多數類別實例密度不同。我們可以明顯的認知到實例 B 的學習難度將是比實例 A 還要困難許多。因此在實例 B 以及其最近多數類別實例之間布置人造實例且依據實例 B 產生的人造實例距離最近多數類別實例的距離應該要比依據實例 A 產生的人造實例與其最近多數類別實例的距離還要來的近。

我們提出的 ISMOTE 對於權重 w_i 以及距離限制 d_i 是在考量到上述的觀察而特別設計的。權重 w_i 以及距離限制 d_i 的計算描述如下:

1. 計算每一個少數類別實例 $x_i \in S_{min}$ 以及距離其最近的多數類別實例

$y_i \in S_{maj}^b$ 之間的距離 r 。

距離少數類別實例 $x_i \in S_{min}$ 最近的多數類別實例 y_i 是可以保證必定是集合 S_{maj}^b 的成員之一，原因是集合 S_{maj}^b 是所有分布位置較靠近於決策邊界的多數類別實例集合。

2. 以少數類別實例 $x_i \in S_{min}$ 作為圓心，距離 r 作為半徑。計算在此圓之內的所有 S_{min} 成員個數 c_{min} (去除圓心)。同理，以多數類別實例 $y_i \in S_{maj}^b$ 作為圓心，距離 r 作為半徑。計算在此圓之內的所有多數類別實例個數 c_{maj} (去除圓心)。

c_{min} 的個數可以用來衡量少數類別實例分布的密度大小，若其個數越大，則代表此少數類別實例 $x_i \in S_{min}$ 位處少數類別實例分布較密集的區域。與此同時， c_{maj} 的個數可以用來衡量多數類別實例分布的密度大小，若其個數越大，則代表此多數類別實例 $y_i \in S_{maj}^b$ 位處多數類別實例分布較密集的區域。

3. 在步驟 2 得到每一個少數類別實例 $x_i \in S_{min}$ 的 C_{min} 和 C_{maj} 後，計算此少數類別實例 $x_i \in S_{min}$ 的權重 w_i ，權重 w_i 越大，代表此少數類別實例 $x_i \in S_{min}$ 越重要。權重 w_i 的計算如下。

$$w_i = \frac{C_{maj} + 1}{C_{min} + 1}$$

依據我們在權重 w_i 的觀察一，位處少數類別實例集團分布較外側的少數類別實例應該要比分布較內側的實例有較高的權重。

而我們的 ISMOTE 可以很好的達成這點。以 Fig. 2a 為例，實例 A 以及實例 C 的 C_{maj} 應是相近的，但實例 A 的 C_{min} 個數應該會比實例 C 的 C_{min} 個數還要小，因為其距離最近多數類別實例較近，因此距離 r 將會比較小，所以能納入考量的少數類別實例也會比較少。因此實例 A 的權重 w_i 會比實例 C 的權重 w_i 還要來的大。

依據我們在權重 w_i 的觀察二，隸屬於分布較稀疏的少數類別集團的少數類別實例其權重 w_i 應該要比隸屬於分布較稠密的少數類別集團的少數類別實例的權重 w_i 還要高。而我們的 ISMOTE 可以很好的達成這點。以 Fig. 2b 為例，實例 A 以及實例 B 距離其各自最近多數類別實例的距離約略相同，但可以很明顯看出，實例 B 的 C_{min} 個數將會比實例 A 的 C_{min} 個數還要小，實例 B 的 C_{maj} 個

數將會比實例 A 的 C_{maj} 個數是差不多的，因此實例 B 的權重將會比實例 A 還要大。

依據我們在權重 w_i 的觀察三，坐落在多數類別分布較稠密的多數類別實例附近的少數類別實例，其權重 w_i 應該要比坐落在多數類別分布較稀疏的多數類別實例附近的少數類別實例，還要來的重。而我們的 ISMOTE 可以很好的達成這點。以 Fig. 2c 為例，實例 A 以及實例 B 的 C_{min} 個數將會是差不多的，但實例 B 的 C_{maj} 將會比 C_{maj} 的實例 A 的 C_{maj} 還要大，因此實例 B 的權重將會比實例 A 還要大。

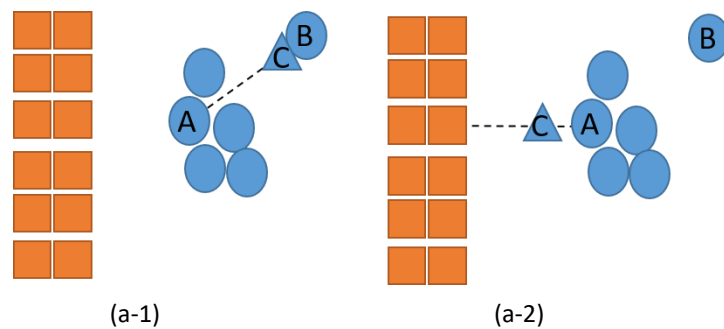
4. 在步驟 2 得到每一個少數類別實例 $x_i \in S_{min}$ 的 C_{min} 和 C_{maj} 後，計算此少數類別實例 $x_i \in S_{min}$ 的距離限制 d_i ，距離限制 d_i 數值越大，代表人造實例的位置可能距離參考少數類別實例越遠，反之則相反。距離限制 d_i 的計算如下。

$$d_i = \frac{C_{maj} + 1}{C_{maj} + 1 + C_{min} + 1}$$

依據我們在距離限制 d_i 的觀察一，少數類別的參考實例所在位置的分布密度若是越高，則人造實例距離此少數類別參考實例的距離應該越近。而我們的 ISMOTE 可以很好的達成這點。以 Fig. 2b 為例，我們可以很明顯的看出實例 A 所在區域的少數類別實例分

布密度較高，而實例 A 的 C_{min} 個數比實例 B 的 C_{min} 個數還要大，因此在兩者的 C_{maj} 個數差不多的情狀之下，實例 A 的距離限制 d_i 比實例 B 的距離限制 d_i 還要小。

依據我們在距離限制 d_i 的觀察二，少數類別實例的參考實例其最近多數類別實例集團的分布密度若是越高，則代表此參考實例具有較大的學習難度，因此需要為其調整比較多的決策邊界，意即我們應該要允許人造實例的分布位置能夠距離少數類別實例較遠。而我們的 ISMOTE 可以很好的達成這點。以 Fig. 2c 為例，我們可以很明顯的看出實例 B 所面對的多數類別實例分布密度較實例 A 高，而實例 B 的 C_{maj} 個數比實例 A 的 C_{maj} 個數還要大，因此在兩者的 C_{min} 個數差不多的情況之下，實例 B 的距離限制 d_i 比實例 A 的距離限制 d_i 還要大。



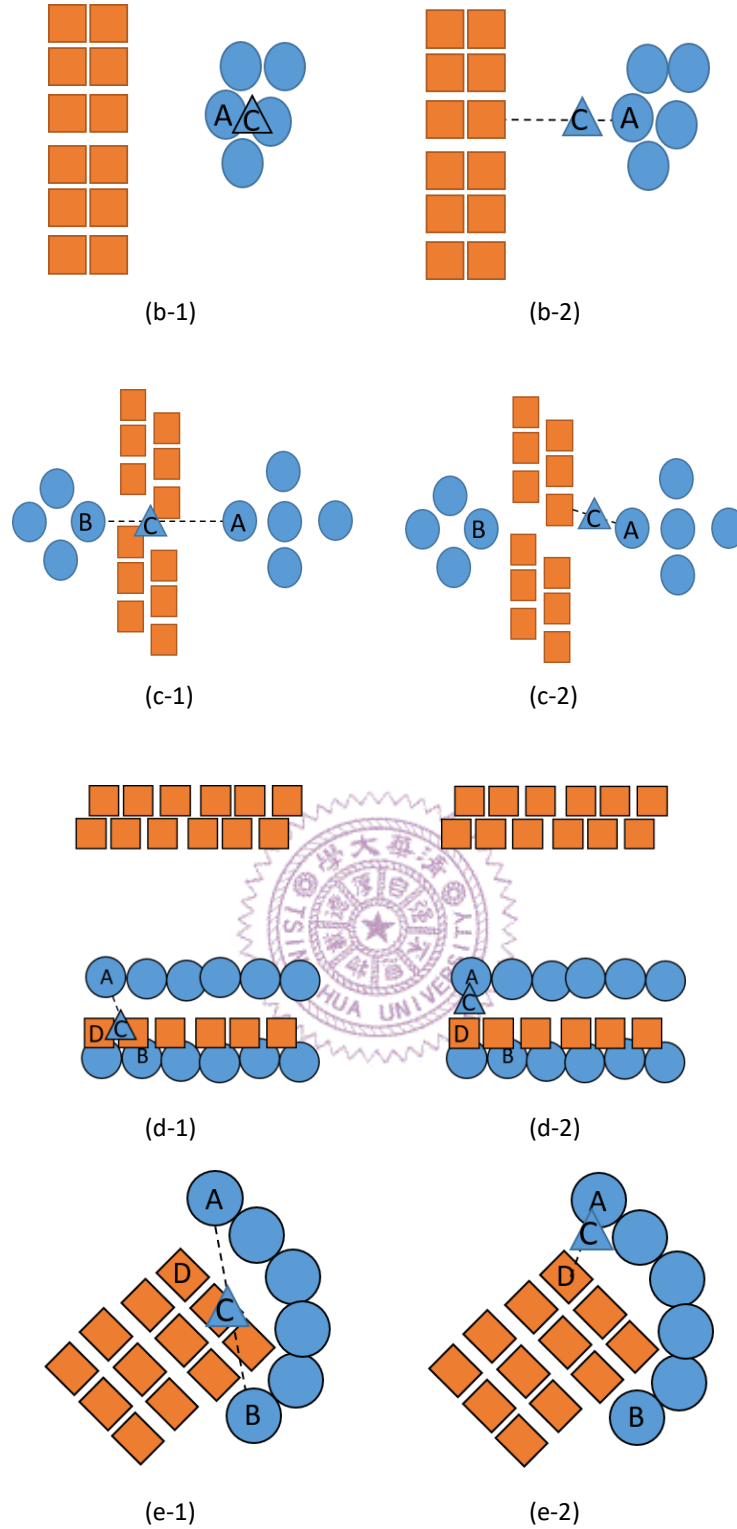


Fig. 3. 圖解了我們的方法與 kNN-based 以及 clustering-based 的方法對於產生人造實例的比較結果。(a-1)使用 kNN-based 的方法(假設 $K=5$)產生的人造實例對於實例 A 所屬集團並無太大影響。(a-2)使用 ISMOTE 產生的人造實例提升分類器對於實例 A 所屬集團的判斷結果。(b-1)使用 kNN-based 的方法(假設 $K=3$)產生的人造實例與實例 A 過近，造成此實例的效果接近於實例 A 的重疊。(b-2)使用 ISMOTE 產生的人造實例能提升分類器對於實例 A 的判斷結果。(c-1)使用 kNN-based 的方法(假設 $K=20$)產生的人造實例落入多數類

別的區域。(c-2)使用 ISMOTE 產生的人造實例提升分類器對於實例 A 的判斷正確。

(d-1)使用 clustering-based 的方法產生的人造實例落入多數類別的區域。(d-2)使用 ISMOTE 產生的人造實例位於實例 A 以及實例 D 之間。(e-1)使用 clustering-based 的方法產生的人造實例落入多數類別的區域。(e-2)使用 ISMOTE 產生的人造實例位於實例 A 以及實例 D 之間。

4.3 人造實例的產生

在第三章，我們描述了許多 kNN-based 以及 clustering-based 的人造實例產生方法所會發生的問題。為了解決這些問題，我們採取了一個改良的作法，使用兩個不同類別的實例做為參考實例執行內插方法。乍聽之下，兩個不同類別的實例內插似乎沒有道理，目前大多數的人造實例產生方法都希望其產生的人造實例是非常有可能真實存在的，因此會直覺地選取兩個少數類別實例做為參考實例以產生人造實例(雖然實際上並不一定符合預期)。然而我們認為類別不平衡問題最重要的事項應該是要解決分類器對於數量稀少的少數類別實例學習效率不佳的問題，因此 ISMOTE 並不以產生非常有可能真實存在的實例為最高目標而是以所產生的人造實例是否能真的提升少數類別實例學習效率這一目的作為最高的準則。

我們假設對於每一個少數類別實例來說，最困難學習的方向是距離其最近

的多數類別實例所在的方位，因此我們使用少數類別實例 $x_i \in S_{min}$ 以及距離其最近的多數類別實例 $y_i \in S_{maj}^b$ 兩者作為參考實例並透過內插方法產生人造實例。此作法一個明顯比 kNN-based 的方法還要好的地方在於使用者設定的最近鄰居參數 K 並不會影響產生人造實例的動作。此作法一個明顯比 clustering-based 的方法還要好的地方在於產生人造實例的位置絕對不可能坐落在多數類別實例的集團之中。

我們將以 Fig. 3. 的圖示證實我們的方法為何比 kNN-based 以及 clustering-based 的人造實例產生方法還要好。kNN-based 的人造實例方法並不考慮最近 K 個鄰居實例的位置分布，其最近 K 個鄰居實例並不一定與參考實例隸屬於同一個集團，由 Fig. 3a-1. 描述的參考實例 A 選擇與它不屬於同一個集團的實例 B 執行內插方法這一情況，可知 kNN-based 的人造實例方法產生的人造實例可能對於少數類別並無幫助。然而我們的方法會將人造實例配置在對於少數類別實例來說，其覺得最困難學習的方位(Fig. 3a-2. 人造實例 C 將位於實例 A 與距離實例 A 最近的多數類別實例之間)，以此方式幫助分類器辨認少數類別實例。

kNN-based 方法的另一個問題是當使用者自訂參數 K 設定的相對過小，則很有可能會產生與參考實例近乎重疊的人造實例，由 Fig. 3b-1. 可以明顯的看出人造實例 C 近乎與人造實例 A 重疊，然而藉由我們的方法產生的人造實例 C 將能夠更好的保護少數類別實例(Fig. 3b-2.)。增加使用者自訂參數 K 的數值能夠避免 Fig. 3b-1 所描述的 kNN-based 方法的問題。但是將 K 值設定的過大卻可能產生另

外的問題，kNN-based 方法可能產生錯誤的人造實例，以 Fig. 3c-1 為例(假設 $K=20$)，實例 A 可能與不同集團的實例 B 執行內插方法產生人造實例，此人造實例 C 將落在多數類別區域。但我們的方法不可能有這種問題，以 Fig. 3c-2 為例，實例 A 不可能選擇距離其橫跨多數類別區域的一個少數類別實例執行內插方法，原因是我們僅允許實例 A 與其最近的多數類別實例執行內插方法。一般來說，一個通用的 K 值是非常不容易找到的。但我們的方法能夠避免選擇正確的 K 值這個問題，透過將 K 值參數與產生人造實例的處理分隔開來這個動作。

clustering-based 方法的問題是 clustering 演算法的分群結果可能是錯誤的。clustering 演算法是一種非監督式的演算法，其並沒有類別的概念(label)。對於 clustering 演算法來說，是否將實例歸類在同一群，是依據實例與實例之間的距離是否夠靠近。以 Fig. 3d-1 為例，實例 A 所屬的集團以及實例 B 所屬的集團因為其彼此距離過近，因此有可能被分群演算法判斷為同一群。所以 clustering-based 方法可能會選擇實例 A 以及實例 B 作為參考實例，並產生可能落在多數類別區域的人造實例 C。但我們的方法不可能有這種問題，原因是參考實例 A，只可能與距離其最近的多數類別實例 D 執行內插方法，因此人造實例 C 不可能落在多數類別的區域(Fig. 3d-2.)。另外 clustering 演算法其對於實例之間距離的設定也將會是個問題，因為這個設定牽涉到兩個集團之間彼此距離多靠近才合併成一群。

clustering-based 方法的另一個問題是即使分群結果正確，但同一個集團的

實例分布卻不一定是呈現常態分布(橢圓形分布)，而這可能會導致人造實例落入多數類別區域。以 Fig. 3e-1 為例，clustering-based 方法可能選擇實例 A 以及實例 B 做為參考實例，產生可能落入多數類別區域的人造實例 C。但我們的方法能完全避免這個問題，因為我們的方法是以少數類別實例與距離其最近的多數類別實例作為參考實例，因此人造實例絕對不可能落在多數類別區域。



Table 1. A confusion matrix for a two-class imbalanced problem

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

第五章 實驗結果

在這個章節，我們測量我們提出的 ISMOTE 的效能以及比較其與 SMOTE[15]，Borderline-SMOTE[14]，ADASYN[13]，MSMOTE[12]，MWMOTE[11]之間的差異。我們從 UCI 網站[48]以及 KEEL 網站[50]收集二十個真實世界的資料集，進行兩個統計上的檢定。Wilcoxon 符號等級檢定 (Wilcoxon signed rank test)[49]以及成對樣本 T 檢定(paired t-tests)。

5.1 效能指標

對於兩個類別的類別不平衡問題的評量，我們經常將數量較稀少的類別稱為 Positive class，而數量較多的類別稱為 Negative class。confusion matrix 是一種非常典型的評估方法。我們將其展現在 Table 1，列代表真實的類別標籤，行代表分類器所預測的類別標籤。TP(True Positive)是被分類器正確地分類的少數類別個數。FN(False Negative)是被分類器錯誤地分類的少數類別個數。FP

(False Positive)是被分類器錯誤地分類的多數類別個數。TN (True Negative) 是被分類器正確地分類的多數類別個數。另外運用 Table 1 的 confusion matrix , 還有數種複合的效能指標能被計算。我們將分類問題的衡量指標詳述如下。

$$1. \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Accuracy是在全體實例中，分類器分類正確的比例，一般來說，Accuracy越高代表所衡量的演算法效能越好。但它並不適用在類別不平衡問題，原因是 Positive class 實例的個數比 Negative class 實例的個數還要少。

$$2. \text{TP}_{rate} = \frac{TP}{TP + FN} \quad (\text{亦稱為 Recall})$$

Recall 是在所有 Positive class 實例中，分類器分類正確的比例。
i.e. 少數類別的 Accuracy。

$$3. \text{FP}_{rate} = \frac{FP}{TN + FP}$$

FP_{rate} 是在所有 Negative class 實例中，分類器分類錯誤的比例。
常見的例子是:假警報，當 FP_{rate} 越高，則代表假警報發生的次數可能越高。

$$4. \text{Precision} = \frac{TP}{TP + FP}$$

Precision 是在所有被分類器判斷為 positives 的實例中，分類器分類正確的比例。與 Recall 之間的選擇在於 positives 實例被分類器判斷錯誤的代價是否非常高，若是，則選用 Recall 會較好。

$$5. AUC = \frac{1+TP_{rate}-FP_{rate}}{2}$$

AUC (the Area Under the Curve) 是 ROC curve 和座標軸之間的區域面積。AUC 代表隨機選擇一個 positive 實例以及隨機選擇一個 negative 實例，然後分類器將此 positive 實例預測正確的比率將比分類器將 negative 實例預測錯誤的比率還要高的機率。AUC 是一個經常被使用去評量分類器效能的指標。越大的 AUC 值，代表分類器的效能越好。

$$6. G - mean = \sqrt{PositiveAccuracy * NegativeAccuracy} = \sqrt{\frac{TP}{TP + FN} * \frac{TN}{TN + FP}}$$

G-mean 反映了分類器在平衡兩個類別之間的能力，G-mean 是一個較全面地評估分類器效能的指標，因為它同時考量分類器對於 positive class 實例的 Accuracy 以及 Negative class 實例的 Accuracy。因此 G-mean 指標越大，代表分類器對於兩類別實例的正確判斷能力越好。

$$7. F - measure = \frac{(1+\beta^2) * Recall * Precision}{\beta^2 * Recall + Precision}$$

F-measure 的參數 β 是使用者可自行調整的參數，用來權衡 Recall 以及 Precision 兩指標之間的重要性，但經常會被設定為 1，代表 Recall 與 Precision 是一樣重要的(我們的實驗也是將 β 設定成 1)。F-measure 是一個同時考量 Precision 以及 Recall 的數值，若 Precision 以及 Recall 都很高則 F-measure 也將很高。因此 F-

measure 能被作為衡量分類器在處理類別不平衡問題的能力指標。

5.2 分類器的設定以及方法的參數設定

為了使檢定結果是有意義的，因此我們需要盡可能地使這些方法的比較起點是相同的。我們限制所有方法在製造人造實例的數量到達類別比 1:1 之後即停止製造人造實例，另外在每一個方法的參數設定上，我們選擇與該方法在發表的論文上的實驗設定相同，因此 SMOTE、Borderline-SMOTE、ADASYN 以及 MSMOTE 的最近鄰居參數 K ，我們將設定為 5。MWMOTE 的參數設定則將 K_1 設定為 5， K_2 設定為 3， K_3 設定為 3， C_{MAX} 設定成 2，人造實例製造數量為多數類別實例數與少數類別實例數的差， $C_f(th)$ 設定成 5， C_p 設定成 3。最後是我們提出的方法 ISMOTE 唯一的參數最近鄰居數 K ，我們將設定成 5。

我們使用 Matlab 作為我們實作實驗所需要的分類演算法的程式語言。我們並沒有特別執著在必須使用哪種分類演算法而是選擇五種較常見的分類演算法作為我們人造實例演算法的實驗環境，這五種演算法分別是 K-NN 演算法 [54]、Decision Tree 的 CART 演算法 [52]、Support Vector Machine [51] 算法、Naive Bayes 演算法 [55]、Logistic Regression 演算法 [53]。對於 K-NN 演算法，我們設定 K 為 10。對於 CART 演算法，我們將事後合併樹葉的能力關閉，降低 CART 演算法的能力，因為我們不希望實驗數據被分類演算法過佳的能力影響。對於 SVM 演算法，我們使用的 Kernel Function 是 gaussian 並且設定執行

資料標準化的動作(Standardize)，以及設定使用 SMO 演算法尋找解。對於 Naive Bayes 演算法以及 Logistic Regression，我們採取 Matlab 預設的參數。

5.3 真實世界的資料集

這一章節將展現 ISMOTE 與其他人造實例方法在二十個真實世界資料集的效能比較。這些資料集在實例數、特徵數、類別數以及類別比例皆不相同，一些資料集超過兩個類別，所以我們會轉換它成為兩個類別的資料集。有一些資料集的屬性不適合人造實例方法，因此必須將之剷除在實驗之外，這部份的動作，我們透過人工觀察該屬性之描述並做出判斷。由於我們其中一個實驗採取 Naive Bayes 演算法。Naive Bayes 假設資料呈常態分佈，因此我們必須剷除一些資料集的資料變化性太小的屬性(變異係數是零)。Table 2 展現了我們所使用的資料集以及其多數類別和少數類別的詳細狀況。

Table 2. Description of Minority Class, Majority Class, and Characteristics of Real-World Data Sets

DataSet	Minority Class	Majority Class	UsedFeatures	UsedFeaturesNum	MinorityNum	MajorityNum	InstanceNum	IR rate
abalone9-18	Class of '18'	Class of '9'	除了'Sex'之外	7	42	689	731	16.4048
Breast Tissue	Class of 'CAR,TAD'	All other class	All	9	36	70	106	1.9444
appendicitis	Class of '0'	Class of '1'	All	7	21	85	106	4.0746
bupa	Class of '1'	Class of '2'	All	6	145	200	345	1.3793
cleveland-0 vs 4	Class of '0'	Class of '4'	除了'sex','fbs'之外	11	13	160	173	12.3077
ecoli1	Class of 'im'	All other class	除了'Lip','Chg'之外	5	77	259	336	3.3636
ecoli3	Class of 'imu'	All other class	除了'Lip','Chg'之外	5	35	301	336	8.6
glass0	Class of '0'	All other class	除了'R1'之外	8	70	144	214	2.0571
Haberman	'Class of 'positive'	Class of 'negative'	All	3	81	225	306	2.7778
heart	Class of '1'	Class of '2'	除了'Age','Sex','ExerciseInduced'之外	10	120	150	270	1.25
new-thyroid1	Class of '2'	All other class	All	5	35	180	215	5.1429
page-blocks0	All other class	Class of 'text'	All	10	559	4913	5472	8.7889
Pima	Class of 'positive'	Class of 'negative'	All	8	268	500	768	1.8657
Robot	Class of 'slight-left-turn', 'slight-right-turn'	All other class	All	24	1154	4302	5456	3.7279
segment0	Class of 'Grass'	All other class	除了'Region-pixel-count', 'Short-line-density-2'之外	17	329	1979	2308	6.0152
vehicle1	Class of 'Saab'	All other class	All	18	217	629	846	2.8986
vehicle3	Class of 'Opel'	All other class	All	18	212	634	846	2.9906
wdbc	Class of 'Malignant'	Class of 'Benign'	除了'Fractal_dimension1', 'Smoothness2', 'Concave_points2', 'Symmetry2', 'Fractal_dimension2'之外	25	212	357	569	1.684
Wisconsin	Class of 'positive'	Class of 'negative'	All	9	239	444	683	1.8577
yeast	Class of 'ME3','ME2', 'EXC','VAC','POX','ERL'	All other class	All	8	304	1180	1484	3.8816

5.3.1 Wilcoxon 符號等級檢定

我們使用無母數檢定中的 Wilcoxon 符號等級檢定(Wilcoxon signed rank test)[49]去成對地兩兩比較我們的方法與上述方法在效能上的差異。首先，我們計算每一對之間的差異(我們的方法與任一比較方法)，令差異 v_i $i=1,2,3,\dots,u$ (差異是絕對值)。由小到大的排序所有的 v_i 並且依據此大小順位給予每一個 v_i 一個排名分數，最小值 v_i 給予排名分數 1，第二小的 v_i 給予排名分數 2，以此類推。如果發生數個差異 v_i 是相同的情況，則將他們的排名分數平均[49]。將每一個 v_i 依據其原始差異是屬於正還是負分類並加總，正分類加總分數以 R^+ 表示，負分類加總分數則以 R^- 表示。 R^+ 和 R^- 之中的最小值可以被轉換成 P-Value 值。我們檢定的假設如下。

$$H_0 : v_D = 0$$

$$H_1 : v_D > 0$$

v_D 表示成對樣本之差異。顯著水準 α 經常會設定成 0.05 或者是 0.1 (我們的實驗是設定成 0.1)。虛無假設是此成對的兩個方法並無顯著的差異，對立假設是此成對的兩個方法具有顯著的差異。如果計算出的 P-Value 是小於或等於顯著水準 α ，則我們會拒絕虛無假設並表示此兩個方法具有顯著的差異，此差異的發生並非僅是偶然。另外為了證明我們的方法在虛無假設成立的時候是真正的與比較方法無二，因此與每一個方法的比較，我們皆會檢定兩次，一次檢定我們的方法是否真正地贏過比較方法，另一次檢定比較方法是否真正地贏過我們的方法。我們將實驗的結果呈現在 Table 3 到 Table 7。



Table 3. ISMOTE 與 B-SMOTE 在 Wilcoxon 符號等級檢定的比較，Win 欄位代表 ISMOTE 是否顯著勝過 B-SMOTE 的數據，Lose 代表 ISMOTE 是否顯著輸給 B-SMOTE 的數據，”*”符號代表具有顯著差異，TotalCount 欄位是計算 ISMOTE 與 B-SMOTE 在五個分類環境的顯著次數比較

Method	Index	KNN		DT		SVM		NaiveB		LogitR		TotalCount	
		Win P-Value	Lose P-Value	Win P-Value	Lose P-Value	Win P-Value	Lose P-Value	Win P-Value	Lose P-Value	Win P-Value	Lose P-Value	Win Count	Lose Count
B-SMOTE	Recall	*0.001531	0.998672	*0.010894	0.99021	*0.000543	0.999524	*0.072267	0.933537	*0.001235	0.998932	5	0
	FP-rate	0.996408	*0.004014	0.962932	*0.040735	0.997855	*0.002412	0.997293	*0.003035	0.997293	*0.003035	0	5
	Precision	0.998309	*0.001906	0.847707	0.161254	0.998827	*0.001327	0.997588	*0.002707	0.999523	*0.000544	0	4
	AUC	*0.060654	0.943703	*0.070155	0.93473	*0.001035	0.999087	0.571851	0.443995	*0.003210	0.997135	4	0
	G-mean	*0.035099	0.967696	*0.025000	0.977104	*0.003399	0.996965	0.44802	0.566707	*0.001327	0.998827	4	0
	F-measure	0.777959	0.233311	*0.098877	0.907466	0.375497	0.638578	0.880198	0.127425	0.46282	0.55198	1	0

Table 4. ISMOTE 與 ADASYN 在 Wilcoxon 符號等級檢定的比較，Win 欄位代表 ISMOTE 是否顯著勝過 ADASYN 的數據，Lose 代表 ISMOTE 是否顯著輸給 ADASYN 的數據，”*”符號代表具有顯著差異，TotalCount 欄位是計算 ISMOTE 與 ADASYN 在五個分類環境的顯著次數比較

		KNN		DT		SVM		NaiveB		LogitR		TotalCount	
		Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose
Method	Index	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	WinCount	LoseCount
ADASYN	Recall	*0.002569	0.997734	*0.001795	0.998459	*0.005233	0.995496	*0.073242	0.936401	*0.002251	0.998062	5	0
	FP-rate	0.999759	*0.000277	0.995987	*0.004613	0.99975	*0.000290	0.999533	*0.000545	0.99971	*0.000336	0	5
	Precision	0.999379	*0.000707	0.91924	*0.086498	0.992763	*0.008021	0.999664	*0.000389	0.997855	*0.002412	0	5
	AUC	0.404133	0.610259	*0.040013	0.963335	0.532077	0.483948	0.852037	0.158254	*0.044694	0.958715	2	0
	G-mean	0.375497	0.638578	*0.003035	0.997293	0.307134	0.705858	0.890161	0.117585	*0.015911	0.985517	2	0
	F-measure	0.766689	0.244891	*0.056297	0.947809	0.638578	0.375497	0.911189	*0.095459	0.200458	0.809843	1	1

Table 5. ISMOTE 與 MSMOTE 在 Wilcoxon 符號等級檢定的比較，Win 欄位代表 ISMOTE 是否顯著勝過 MSMOTE 的數據，Lose 代表 ISMOTE 是否顯著輸給 MSMOTE 的數據，”*”符號代表具有顯著差異，TotalCount 欄位是計算 ISMOTE 與 MSMOTE 在五個分類環境的顯著次數比較

		KNN		DT		SVM		NaiveB		LogitR		TotalCount	
		Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose
Method	Index	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	WinCount	LoseCount
MSMOTE	Recall	*0.023175	0.978945	*0.088396	0.918382	*0.002415	0.997947	*0.016577	0.98528	*0.000994	0.999143	5	0
	FP-rate	0.999311	*0.000792	0.343642	0.671044	0.992547	*0.008324	0.963335	*0.040013	0.964901	*0.038089	0	4
	Precision	0.999818	*0.000209	0.307134	0.705858	0.975	*0.027263	0.943703	*0.060654	0.86462	0.143669	0	3
	AUC	0.559359	0.455409	*0.041271	0.961923	*0.055948	0.948446	0.190157	0.819812	*0.000621	0.999456	3	0
	G-mean	0.692866	0.320372	*0.013166	0.988047	*0.038089	0.964901	0.566707	0.44802	*0.002412	0.997855	3	0
	F-measure	0.93473	*0.070155	*0.013166	0.988047	*0.080760	0.924685	0.347532	0.666157	*0.014483	0.986834	3	1

Table 6. ISMOTE 與 MWMOTE 在 Wilcoxon 符號等級檢定的比較，Win 欄位代表 ISMOTE 是否顯著勝過 MWMOTE 的數據，Lose 代表 ISMOTE 是否顯著輸給 MWMOTE 的數據，”*”符號代表具有顯著差異，TotalCount 欄位是計算 ISMOTE 與 MWMOTE 在五個分類環境的顯著次數比較

		KNN		DT		SVM		NaiveB		LogitR		TotalCount	
		Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose
Method	Index	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	WinCount	LoseCount
MWMOTE	Recall	0.516052	0.5	0.904721	0.102282	0.961706	*0.041763	0.952698	*0.053497	0.996515	*0.004013	0	3
	FP-rate	*0.038089	0.964901	*0.000520	0.99955	*0.001173	0.998965	*0.000738	0.999365	*0.000072	0.999939	5	0
	Precision	*0.019134	0.98254	*0.001327	0.998827	*0.001035	0.999087	*0.004721	0.995804	*0.000544	0.999523	5	0
	AUC	*0.030654	0.97203	*0.032304	0.970305	*0.010839	0.990184	*0.088498	0.918283	*0.023186	0.978935	5	0
	G-mean	0.119802	0.887495	*0.015911	0.985517	*0.013166	0.988047	0.13422	0.874289	*0.022896	0.979056	3	0
	F-measure	*0.014483	0.986834	*0.001691	0.998501	*0.002707	0.997588	*0.033549	0.969346	*0.002145	0.998094	5	0

Table 7. ISMOTE 與 SMOTE 在 Wilcoxon 符號等級檢定的比較，Win 欄位代表 ISMOTE 是否顯著勝過 SMOTE 的數據，Lose 代表 ISMOTE 是否顯著輸給 SMOTE 的數據，”*”符號代表具有顯著差異，TotalCount 欄位是計算 ISMOTE 與 SMOTE 在五個分類環境的顯著次數比較

		KNN		DT		SVM		NaiveB		LogitR		TotalCount	
		Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose
Method	Index	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	P-Value	WinCount	LoseCount
SMOTE	Recall	0.147963	0.86187	0.960002	*0.043590	0.308204	0.706954	0.652522	0.363707	0.180155	0.831053	0	1
	FP-rate	0.943718	*0.060638	*0.019134	0.98254	0.652468	0.361422	0.692866	0.320372	0.619761	0.39693	1	1
	Precision	0.991979	*0.008879	0.143669	0.86462	0.256774	0.755109	0.929845	*0.075315	0.200458	0.809843	0	2
	AUC	0.743244	0.268933	0.686292	0.327065	*0.025000	0.977104	0.829447	0.180188	*0.027263	0.975	2	0
	G-mean	0.847707	0.161254	0.610259	0.404133	*0.041285	0.961911	0.880198	0.127425	*0.017460	0.984089	2	0
	F-measure	0.819812	0.190157	0.268951	0.743226	*0.032304	0.970305	0.86462	0.143669	*0.065270	0.939346	2	0

我們總結這些人造實例方法在 TotalCount 欄位的勝負結果，依照效能指標

AUC、G-mean、F-measure 的評比，我們發現除了在 MSMOTE 以及 ADASYN 的 F-measure 有一個顯著輸的次數外，其餘皆為零。但這三個指標顯著贏的次數總和卻至少超過五次。也就是說，在這三個指標評測之下，ISMOTE 顯著贏過這些方法的次數是遠遠大於顯著輸的次數的。因此我們可以認為 ISMOTE 的效能是比 SMOTE、B-SMOTE、ADASYN、MSMOTE 以及 MWMOTE 還要好的。這之中勝過 MWMOTE 的次數最多，其次是 MSMOTE 以及 B-SMOTE。

另外觀測效能指標 Recall、 FP_{rate} 、Precision 在 TotalCount 欄位的數值，可以發現 ISMOTE 在與 B-SMOTE、ADASYN、MSMOTE 的比較中，其 Recall 顯著贏的次數是遠遠大於顯著輸的次數的。我們認為造成這結果的原因是 ISMOTE 是使用少數類別實例以及距離其最近的多數類別實例作為參考實例以產生人造實例，也就是說 ISMOTE 會明顯改變分類器對於少數類別決策邊界的認知。拓寬少數類別的決策邊界就表示有可能提高多數類別實例會被判斷錯誤的可能性，因此 ISMOTE 在 FP_{rate} 顯著輸的次數是遠大於顯著贏的次數，也就是說 ISMOTE 具有偏袒少數類別的特性，而這特性連帶使得 ISMOTE 在 Precision 指標的表現也不太好。但觀察 SMOTE 以及 MWMOTE 可以發現其結果與上述三個方法相反。我們認為會造成 MWMOTE 的 Recall 效能比 ISMOTE 的效能好的原因是 MWMOTE 有時候會錯誤將人造實例分配在多數類別區域之中，而這會造成其改變少數類別決策邊界的影響比 ISMOTE 還要多，但也就表示其造成

分類器錯誤辨別多數類別實例的可能性是比 ISMOTE 還要高的，因此其 FP_{rate} 將比 ISMOTE 還要差很多。由目前的數據說明 SMOTE 在 Recall 以及 Precision 的表現是比 ISMOTE 還要好的，但因為其勝負的差距並不大，因此我們認為這個結果的認定是有待商榷的。

5.3.2 成對樣本 T 檢定

我們運用成對樣本 T 檢定(Paired t-tests)去再一次驗證 ISMOTE 的效能真的是比其他人造實例方法還要好的。這虛無假設以及對立假設如下：

$$H_0 : u_1 - u_2 = 0$$

$$H_1 : u_1 - u_2 > 0$$

u_1 代表欲測量方法的平均值， u_2 代表比較的人造實例方法的平均值。我們共計算三十次的 5-fold 平均值數據。加上比較人造實例方法的數據，我們具有三十對 5-fold 平均值數據，因此自由度 df 是 29。每一對數據求出其差 v_i ，再依據這三十個 v_i 計算出差的平均值 Δu 以及差的標準差 $\Delta \sigma^2$ ，再以此計算出 t 統計值之後，再求出 P-Value 值即可開始檢定。如果 P-Value 小於顯著水準 α 則拒絕虛無假設 H_0 ，也就是說欲測量方法是顯著優於比較方法的。顯著水準 α 一般會設定成 0.05 或是 0.1，此處的實驗設定為 0.1。為了證實我們的方法是真正的贏過比較的人造實例方法，因此與每一個比較人造實例方法，我們皆會檢定兩次。一次檢定我們的方法是否顯著贏過比較人造實例方法，另一次檢定比較人造實

例方法是否顯著贏過我們的方法。檢定的結果將會表現在 Table 8 到 Table 12。

Table 8. 透過 t 檢定，加總 ISMOTE 與 B-SMOTE 在 20 個資料集的顯著次數，Win 欄位代表 ISMOTE 是否顯著勝過 B-SMOTE 的次數，Lose 代表 ISMOTE 是否顯著輸給 B-SMOTE 的次數，TotalCount 欄位是計算 ISMOTE 與 B-SMOTE 在五個分類環境的勝負次數比較

Method	Index	KNN		DT		SVM		NaiveB		LogitR		TotalCount	
		Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose
B-SMOTE	Recall	17	2	12	5	17	3	13	5	18	2	5	0
	FP-rate	3	17	4	14	3	17	4	16	2	18	0	5
	Precision	5	15	3	14	2	16	6	14	3	16	0	5
	AUC	17	2	7	7	14	5	9	11	17	2	3	1
	G-mean	14	5	7	6	14	6	9	10	16	2	4	1
	F-measure	11	8	5	9	10	9	8	11	11	6	3	2

Table 9. 透過 t 檢定，加總 ISMOTE 與 ADASYN 在 20 個資料集的顯著次數，Win 欄位代表 ISMOTE 是否顯著勝過 ADASYN 的次數，Lose 代表 ISMOTE 是否顯著輸給 ADASYN 的次數，TotalCount 欄位是計算 ISMOTE 與 ADASYN 在五個分類環境的勝負次數比較

Method	Index	KNN		DT		SVM		NaiveB		LogitR		TotalCount	
		Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose
ADASYN	Recall	17	2	15	2	16	4	15	4	19	1	5	0
	FP-rate	2	17	2	15	2	18	3	17	2	18	0	5
	Precision	3	17	5	15	4	14	2	17	4	15	0	5
	AUC	12	7	9	6	10	6	8	10	12	6	4	1
	G-mean	10	7	8	3	10	6	7	11	15	5	4	1
	F-measure	10	9	7	6	10	9	5	12	11	6	4	1

Table 10. 透過 t 檢定，加總 ISMOTE 與 MSMOTE 在 20 個資料集的顯著次數，Win 欄位代表 ISMOTE 是否顯著勝過 MSMOTE 的次數，Lose 代表 ISMOTE 是否顯著輸給 MSMOTE 的次數，TotalCount 欄位是計算 ISMOTE 與 MSMOTE 在五個分類環境的勝負次數比較

Method	Index	KNN		DT		SVM		NaiveB		LogitR		TotalCount	
		Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose
MSMOTE	Recall	13	7	8	8	14	5	14	5	15	4	4	0
	FP-rate	5	15	7	9	7	11	7	12	6	14	0	5
	Precision	4	16	4	9	8	10	7	13	10	9	1	4
	AUC	10	9	10	7	14	5	14	6	15	4	5	0
	G-mean	10	9	10	7	15	3	12	7	16	4	5	0
	F-measure	7	11	8	8	12	7	10	9	12	5	3	1

Table 11. 透過 t 檢定，加總 ISMOTE 與 MWMOTE 在 20 個資料集的顯著次數，Win 欄位代表 ISMOTE 是否顯著勝過 MWMOTE 的次數，Lose 代表 ISMOTE 是否顯著輸給 MWMOTE 的次數，TotalCount 欄位是計算 ISMOTE 與 MWMOTE 在五個分類環境的勝負次數比較

		KNN		DT		SVM		NaiveB		LogitR		TotalCount	
		Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose
Method	Index	Count	Count	Count	Count	Count	Count	Count	Count	Count	Count	WinCount	LoseCount
MWMOTE	Recall	12	7	6	13	6	12	8	11	7	12	1	4
	FP-rate	9	10	18	0	15	3	14	5	17	1	4	1
	Precision	9	8	14	2	15	4	13	6	16	3	5	0
	AUC	12	6	13	5	11	4	13	5	13	7	5	0
	G-mean	12	8	13	5	10	4	12	7	14	6	5	0
	F-measure	10	7	12	3	12	3	10	7	13	5	5	0

Table 12. 透過 t 檢定，加總 ISMOTE 與 MWMOTE 在 20 個資料集的顯著次數，Win 欄位代表 ISMOTE 是否顯著勝過 MWMOTE 的次數，Lose 代表 ISMOTE 是否顯著輸給 MWMOTE 的次數，TotalCount 欄位是計算 ISMOTE 與 MWMOTE 在五個分類環境的勝負次數比較

		KNN		DT		SVM		NaiveB		LogitR		TotalCount	
		Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose	Win	Lose
Method	Index	Count	Count	Count	Count	Count	Count	Count	Count	Count	Count	WinCount	LoseCount
SMOTE	Recall	11	6	4	13	8	11	8	9	10	10	1	3
	FP-rate	7	12	15	5	11	7	9	10	13	6	4	1
	Precision	7	12	10	8	13	6	7	12	14	5	3	2
	AUC	11	9	7	9	14	4	10	9	15	5	4	1
	G-mean	10	9	7	10	13	4	7	10	16	4	3	2
	F-measure	8	11	8	8	12	5	8	10	15	4	2	2

我們總結這些人造實例方法在 TotalCount 欄位的勝負結果，依照效能指標 AUC、G-mean、F-measure 的評比，我們發現 ISMOTE 在 20 個資料集的勝過其他人造實例方法的顯著次數是遠遠多過於其他人造實例方法勝過 ISMOTE 的顯著次數的，由此數據我們可以推論 ISMOTE 的整體效能是比其他人造實例方法還要好的。其中勝差最大的是 MWMOTE，再來是 MSMOTE。

依據效能指標 Recall、FP_{rate}、Precision 在 TotalCount 欄位的數值，可以發現 ISMOTE 在與 B-SMOTE、ADASYN、MSMOTE 的比較中，其 Recall 顯著贏的次數是遠遠大於顯著輸的次數的。我們認為造成這結果的原因是 ISMOTE 是使用少數類別實例以及距離其最近的多數類別實例作為參考實例以產生人造實例，也就是說 ISMOTE 會明顯改變分類器對於少數類別決策邊界的認知。拓寬

少數類別的決策邊界就表示有可能提高多數類別實例會被判斷錯誤的可能性，因此 ISMOTE 在 FP_{rate} 顯著輸的次數是遠大於顯著贏的次數，也就是說 ISMOTE 具有偏袒少數類別的特性，而這特性連帶使得 ISMOTE 在 Precision 指標的表現也不太好。但觀察 SMOTE 以及 MWMOTE 可以發現結果與上述三個人造實例方法相反。MWMOTE 因為其產生的人造實例有時會坐落在多數類別區域，因此其改變較 ISMOTE 更多的少數類別決策邊界，因此 Recall 會比 ISMOTE 高。但相對的， FP_{rate} 以及 Precision 將會比 ISMOTE 還要低。SMOTE 因為其有可能選擇多數類別實例作為產生人造實例的參考實例，但 SMOTE 假設資料是均勻分布，其並沒有限制人造實例產生的位置，而這可能會過多的影響少數類別決策邊界，因此 SMOTE 的 Recall 會比 ISMOTE 的 Recall 較高。但相對的， FP_{rate} 以及 Precision 將會比 ISMOTE 還要低。

5.4 討論

有兩個與 ISMOTE 有關的有趣問題值得我們去探討，我們詳述如下

1. ISMOTE 的最近鄰居 K 參數應該如何設定？

對於找出雜訊的少數類別實例。如果某一個少數類別實例的最近的 K 位鄰居實例皆是多數類別實例，ISMOTE 將會假定此少數類別實例為雜訊實例並將其移出參考實例的考量之外。在先前的實

驗，我們使用 K 等於 5 這個參數。另外其他可能可以的設定是 3、4 或者是 10 等等。然而若是設定像是 20 這樣大的數字，則可能是不合理的，因為雜訊少數類別實例並不一定會擁有二十個多數類別實例鄰居。另外若是設定像是 3 這樣小的數字，則可能會錯誤移出有價值的少數類別實例。因此我們認為任何在 5 到 10 之間的數字將會是 K 值比較理想的設定。

2. 在 AUC、G-mean、F-measure 這些較全面衡量的指標，為何 ISMOTE 會表現得比較好？

對於 QSTMOT 我們一開始設計的概念是希望其能夠盡可能的保障分類器對於少數類別實例的辨認，在此前提之下，盡可能不影響多數類別實例的辨認，因此 ISMOTE 在認定上是較偏袒少數類別的方法。此特性可從 Recall 這個指標觀察到。但為了不降低過多的多數類別實例的辨認率，因此我們加上距離限制 d_i 去減緩 FP_{rate} 的提升。我們有非常高的 Recall，以及略低的 FP_{rate} ，所以 ISMOTE 的 AUC 會比 B-SMOTE、ADASYN、MSMOTE 還要高。SMOTE 並沒有設定距離限制 d_i 去減緩 FP_{rate} 的提升，MWMOTE 則是本身就有 FP_{rate} 過高的問題，因此 ISMOTE 的 AUC 會比此兩個人造實例方法高。在 G-mean 部分，因為多數類別實例的數量是較多的，因此 TN 不會表現太差，但因為我們的

FP_{rate} 是略低於 B-SMOTE、ADASYN、MSMOTE，所以 ISMOTE 的 $NegativeAccuracy = \frac{TN}{TN + FP}$ 將會略低於 B-SMOTE、ADASYN、MSMOTE 這三個人造實例方法，但因為我們有非常高的 Recall，因此 ISMOTE 的 G-mean 將會勝過此三個方法，MWMOTE 因為其會將人造實例分布在多數類別區域，因此會嚴重影響到 TN 的數量，導致 $NegativeAccuracy$ 降低，因此 ISMOTE 的 G-mean 將會比 MWMOTE 高。而 SMOTE 並沒有 MWMOTE 這樣嚴重，但 FP_{rate} 是比 ISMOTE 嚴重許多，因此 ISMOTE 的 G-mean 將會比 SMOTE 高。另外在 F-measure 部分，雖然 Precision 將會略低，但因為有非常高的 Recall，因此 ISMOTE 的 F-measure 將會勝過 B-SMOTE、ADASYN、MSMOTE。因為 MWMOTE 的 FP_{rate} 是較高的，導致 Precision 遠低於 ISMOTE，因此 ISMOTE 的 F-measure 將會勝過 MWMOTE。

第六章 結論

類別不平衡問題是一個非常重要的問題，因此已經有許多的研究者投入找出這項問題的解決方法。產生人造實例的方法是一項廣為被使用的技術，因為它改善分類器容易忽略分布較稀疏的少數類別實例的特性。kNN-based 的人造實例方法常面臨最近鄰居 K 參數的設定問題，因為 K 參數的設定會影響其對於人造實例重要性的衡量以及影響參考實例的選擇。當 K 參數的設定值越小，則越有可能發生權重衡量錯誤的狀況，此狀況我們已經在第三章的 Fig. 1a 詳細描述過了。另外，產生的人造實例可能對於分類器辨認少數類別的幫助不大，此問題我們描述在 Fig. 3a-1、Fig. 3a-1。而當 K 參數設定過大，則所產生的人造實例將有可能會配置在錯誤的位置，此問題我們描述在 Fig. 3c-1。我們在這篇研究的實驗將 K 一律設定在 5，因此 Fig. 3c-1 的問題在我們的實驗中應該並不嚴重。ISMOTE 率先採用區域的密度概念去衡量少數類別實例的權重，此方式解決重要性權重配置不合理的問題。另外，ISMOTE 選擇以少數類別實例以及距離其最近的多數類別實例作為參考實例，此作法保證產生的人造實例對於幫助分類器辨認少數類別上有一定程度的幫助。如此，ISMOTE 也解決使用者需要尋找適當 K 值設定的困擾，因為 ISMOTE 對於最近鄰居參數 K 的依賴並不如 kNN-based 的人造實例方法這樣嚴重。clustering-based 的人造實例方法的問題在於其選擇參考實例的基準是非常依賴 clustering 演算法的結果，我們已經在

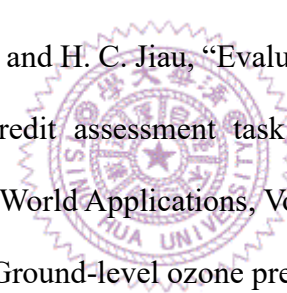
Fig. 1d 以及 Fig. 1e 描述過它的問題。

依據我們的實驗結果，ISMOTE 有比較好的 AUC 值、F-measure 值以及 G-means 值相對於其他的人造實例方法。這樣好的整體表現是因為 ISMOTE 改善選取參考樣本的機制以及改善產生人造實例的機制。此兩項機制能保障 ISMOTE 對於提升分類器辨認少數類別實例的正確率，以及盡可能地不影響分類器對於多數類別實例的判斷。這統計的分析結果將支持我們的論點。

雖然我們已經成功地改善分類器對於少數類別的判斷率，但 ISMOTE 仍然有一些需要改進的地方。像是在保持現有的優勢之下改善 FP_{rate} 的數值，或者結合 clustering 演算法以增加演算法能夠參考的資訊，比如實例所隸屬的集團等。



參考文獻

- 
- [1] Y. M. Huang, C. M. Hung and H. C. Jiau, “Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem,” *Nonlinear Analysis: Real World Applications*, Vol. 7, No. 4, pp. 720–747, 2006
- [2] W. Z. Lu and D. Wang, “Ground-level ozone prediction by support vector machine approach with a cost-sensitive classification scheme,” *Sci Total Enviro*, Vol. 395, No. 2–3, pp. 109–116, 2008
- [3] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, G. D. Tourassi, “Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance,” *Neural Netw.*, Vol. 21, No. 2–3, pp. 427–436, 2008
- [4] S. Barua, M. M. Islam, X. Yao, K. Murase, “MWMOTE – majority weighted minority oversampling technique for imbalanced data set learning,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 2, pp. 405–425, 2014
- [5] A. Ali, S. M. Shamsuddin, A. L. Ralescu, “Classification with class imbalance

- problem: a review,” *Advance Soft Compute Appl*, Vol. 7, No. 3, pp. 176–204, 2015
- [6] Japkowicz, N. and S. Stephen, “The class imbalance problem: A systematic study,” *Intelligent data analysis*, Vol. 6, No. 5, pp. 429–449, 2002
- [7] Ronaldo C. Prati, Gustavo E. A. P. A. Batista, “Class Imbalances versus Class Overlapping: An Analysis of a Learning System Behavior,” *Mexican International Conference on Artificial Intelligence*, pp. 312–321, 2004
- [8] Jo. T. and N. Japkowicz, “Class imbalances versus small disjoints,” *ACM SIGKDD Explorations Newsletters*, Vol. 6, No. 1, pp. 40–49, 2004
- [9] Denil. M, T. Trappenberg, “Overlap versus imbalance,” *Advances in Artificial Intelligence*, Vol. 6085, pp. 220–231, 2010
- [10] Weiss, G. M. and F. Provost, “Learning when training data are costly: The effect of class distribution on tree induction,” *Journal of Artificial Intelligence Research*, pp. 315–354, 2003
- [11] S. Barua, M. M. Islam, X. Yao, K. Murase, “MWMOTE – majority weighted minority oversampling technique for imbalanced data set learning,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 2, pp. 405–425, 2014
- [12] Hu. S. Liang, Y. Ma, L. and He. Y. “MSMOTE: improving classification performance when training data is imbalanced,” *Workshop on Computer Science and Engineering*, Vol. 2, pp. 13–17. IEEE, 2009
- [13] H. He, Y. Bai, E. A. Garcia, S. Li, “ADASYN: adaptive synthetic sampling approach for imbalanced learning,” *IEEE International Joint Conference on Neural Networks*, pp. 1322–1328, 2008
- [14] H. Han, W.Y. Wang, B.H. Mao, “Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning,” *International Conference on Intelligent Computing*, pp. 878–887, 2005.

- [15] N.V. Chawla, K. K. W. Bowyer, L. O. Hall, W. P. Kegel Meyer, "SMOTE: Synthetic Minority Over-Sampling Technique," *Journal of Artificial Intelligence Research* 16, pp. 321–357, 2002
- [16] N. Japkowic and S. Stephen, "The Class Imbalance Problem: A Systematic Study," *Intelligent Data Analysis*, Vol. 6, No. 5, pp. 429–449, 2002.
- [17] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, "Safe-level-SMOTE: Safe-level-synthetic minority over-sampling Technique for handling the class imbalanced problem," *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp. 475-482, 2009
- [18] Bunkhumpornpat. C and Subpaiboonkit. S "Safe level graph for synthetic minority over-sampling techniques," *13th International Symposium on Communications and Information Technologies*, pp. 570–575, 2013
- [19] Enislay. Ramentol, Yael. Caballero, Rafael. Bello, and Francisco. Herrera. "SMOTE-RSB*: A hybrid preprocessing approach based on oversampling and under sampling for high imbalanced data-sets using SMOTE and rough sets theory," *Knowledge and Information Systems*, pp. 245–265, 2012
- [20] Chumphol. Bunkhumpornpat, Krung. Sinapiromsaran, Chidchanok. Lursinsap. "DBSMOTE: Density-based synthetic minority over-sampling technique," *Applied Intelligence*, Vol. 36, No. 3, pp. 664–684, 2012
- [21] P. Branco, L. Torgo, and R. P. Ribeiro "A Survey of Predictive Modeling on Imbalanced Domains," *ACM Computing Surveys*, Vol. 49, No. 2, pp. 31:1–31:50, 2016.
- [22] He. H. Garcia, E. A. "Learning from imbalanced data," *IEEE Transactions On Knowledge and Data Engineering*, Vol. 21, pp. 1263–1284, 2009
- [23] S. Chen, H. He, E.A. Garcia, "Ramoboot: Ranked minority oversampling in boosting," *IEEE Transactions on Neural Networks*, Vol. 21, No. 10, pp.1624–1642,

2010

- [24] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera “A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches,” IEEE Transactions on Systems, Man, and Cybernetics – part C: Applications and Reviews, Vol. 42, No. 4, pp. 463–484, 2011
- [25] R. C. Prati, G. E. Batista, D. F. Silva, “Class imbalance revisited: a new experimental setup to assess the performance of treatment methods,” Knowledge and Information Systems, pp. 1–24, 2014
- [26] M. Nakamura, Y. Kajiwar, A. Otsuka, H. Kimura, “LVQ-SMOTE – learning vector quantization based synthetic minority over-sampling technique for biomedical data,” BioData Min, 2013
- [27] Yoav. Freund. “Boosting a weak learning algorithm by Majority,” Information and Computation, Vol. 121, pp. 256–285, 1995
- [28] L. Breiman, “Random forests,” Machine learning, Vol. 45, No. 1, pp. 5–32, 2001
- [29] J.A. Saez, J. Luengo, J. Stefanowski, F. Herrera, “SMOTE-IPF: addressing the noisy and borderline examples problem in imbalanced classification by a resampling method with filtering,” Information Sciences, Vol. 291, pp. 184–203, 2015
- [30] R. Barandela, R.M. Valdovinos, J.S. Sánchez, F.J. Ferri, “The imbalanced training sample problem: under or over sampling?,” Structural, Syntactic, and Statistical Pattern Recognition, Lectures Notes in Computer Science, Vol. 3138, pp. 806–814, 2004
- [31] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, “Study of the behavior of several methods for balancing machine learning training data,” ACM SIGKDD Explorations Special Issue on Learning from Imbalanced Datasets, Vol. 6, No. 1, pp. 20–29, 2004

- [32] N. Japkowicz, S. Stephen, "The class imbalance problem: a systematic study, *Intell,*" *Data Anal*, Vol. 6, No. 5, pp. 429–449, 2002
- [33] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, "Study of the behavior of several Methods for balancing machine learning training data," *ACM SIGKDD Explorations Special Issue on Learning from Imbalanced Datasets Expl Newsl*, Vol. 6, No. 1, pp. 20–29, 2004
- [34] H. Zhang and M. Li, "RWO-sampling: A random walk over-sampling approach to imbalanced data classification," *Inform Fusion*, Vol. 20, pp. 99–116, 2014.
- [35] M. Gao, X. Hong, S. Chen, C.J. Harris, E. Khalaf, "PDFOS: PDF estimation based over-sampling for imbalanced two-class problems," *Neurocomputing*, 2014
- [36] Sezer EA, Nefeslioglu HA, Gokceoglu "An assessment on producing synthetic samples by fuzzy C-means for limited number of data in prediction models," *Appl Soft Comput*, Vol. 24, pp. 126–134, 2014
- [37] B. Tang and H. He, "KernelADASYN: Kernel based adaptive synthetic data generation for imbalanced learning," *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 664–671, 2015
- [38] A. Fernandez, V. V. L´opez, M. J. del Jesus, and F. Herrera, "Revisiting ´ evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges," *Knowledge-Based Systems*, In Press, Accepted Manuscript, pp. 109–121, 2015
- [39] S. García, J. Luengo, F. Herrera, "Tutorial on practical tips of the most influential data preprocessing algorithms in data mining," *Knowledge-Based Systems*, Vol. 98, pp. 1–29, 2016
- [40] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and María Carolina Monard, "A study of the behaviour of several methods for balancing machine learning training data," *SIGKDD Explorations*, Vol. 6, No. 1, pp. 20–29, 2004
- [41] Enislay. Ramentol, Yael. Caballero, Rafael. Bello, and Francisco. Herrera.

- “SMOTE-RSB*: A hybrid preprocessing approach based on oversampling and under sampling for high imbalanced data-sets using SMOTE and rough sets theory,” Knowledge and Information Systems, pp. 245–265, 2012
- [42] VERBIEST, N., RAMENTOL, E., CORNELIS, C., AND HERRERA, “Improving smote with fuzzy rough prototype selection to detect noise in imbalanced classification data.” In Proceedings of the 13th Ibero-American Conference on Artificial Intelligence, pp. 169–179, 2012
- [43] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, “Safe-level-SMOTE: Safe-level-synthetic minority over-sampling Technique for handling the class imbalanced problem,” Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, pp. 475-482, 2009
- [44] T. Maciejewski, J. Stefanowski, “Local neighbourhood extension of SMOTE for mining imbalanced data,” Proceeding of the IEEE symposium on computational intelligence and data mining, pp. 104–111, 2011
- [45] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, “DBSMOTE: Densitybased synthetic minority over-sampling technique,” Applied Intelligence, Vol. 36, pp. 664–684, 2011.
- [46] A. Fernández, V. López, M. Galar, M.J. del Jesus, F. Herrera, “Analysing the classification of imbalanced data-sets with multiple classes: binarization techniques and ad-hoc approaches,” Knowledge-Based Systems Vol. 42, pp. 97–110, 2013
- [47] Ester M, Kriegel H-P, Sander J, Xu X, “A density-based algorithm
Ford is covering clusters in large spatial databases with noise,” International Conference on Knowledge Discovery in Databases and Data Mining, Vol. 2, pp. 226-231, 1996

- [48] UCI Machine Learning Repository, <http://archive.ics.uci.edu/ml/> , 2009
- [49] G.W. Corder and D.I. Foreman, “Nonparametric Statistics for NonStatisticians: A Step-by-Step Approach,” John Wiley & Sons, 2009
- [50] KEEL (Knowledge Extraction based on Evolutionary Learning)
<http://sci2s.ugr.es/keel/>
- [51] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf. “Support vector Machines,” Intelligent Systems and their Applications, IEEE, Vol. 13, No. 4, pp. 18–28, 1998.
- [52] R. J. Lewis. “An introduction to classification and regression tree (CART) analysis” Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, California, pp. 1–14, 2000
- [53] J.H. Friedman, T. Hastie, R. Tibshirani, “The Elements of Statistical Learning : Data Mining, Inference, and Prediction,” 2001.
- [54] Wu, W. et al, “Chemometric strategies for normalisation of gene expression data Obtained from cDNA microarrays Analytica,” Chimica Acta, 446, pp. 449–464, 2001.
- [55] Sucar, E. Probabilistic Graphical Models: Principles and Applications, Springer. 2015