Imbalanced Dataset

104403044 吳謦廷 104403011 范哲豪

Abstract

不平衡資料集(Imbalanced Data Set)是資料科學中核心的問題。現實生活中的資料集幾乎都是不平衡的，也因此面對不平衡資料集，我們必須要思考如何透過適當的樣本採集方法或演算法去平衡類別的樣本數。本文獻會先定義不平衡資料集，並列舉其特徵，再依據這些特徵，討論五個解決不平衡資料集的等級，每個等級有其解決不平衡資料集的方法與學習演算法。最後我們再探討當今最適合處理不平衡資料集的演算法。

關鍵字：不平衡資料集；樣本採集；學習演算法

1. 介紹不平衡資料集

1.1. 定義不平衡資料集

在資料科學與機器學習領域中，不平衡資料集是一個難以避免的核心問題，主要存在於監督式機器學習中，指的是一個資料集中，不同類別的樣本數量分配不均或是有顯著的差別，不平衡資料集會讓樣本數量小的類別在資料分析的過程中所提供的資訊，被樣本數量大的類別所覆蓋掉，而樣本數量小的類別，其樣本所攜帶的特徵資訊通常是資料分析的關鍵，也因此會讓資料分析的結果雖然準確率高，但是應用在實際生活中的資料集時，卻不能準確的預測出我們預想的結果。

1.2. 生活中的不平衡資料集

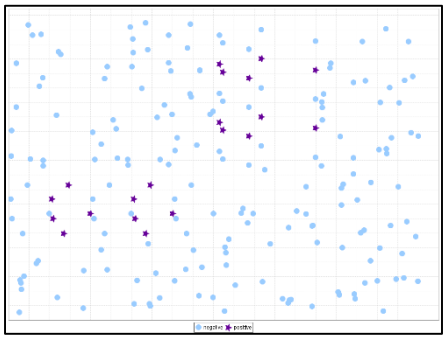
不平衡資料集四處存在於我們的生活之中，像是商場上信用卡盜刷、黃牛訂單、大型企業客戶流動率……等，舉信用卡盜刷來說，在現實生活中，信用卡交易絕大多數都是正常的，僅有少數信用卡盜刷的案例，。在生物醫學領域中的資料科學，不平衡資料集存在於罕見疾病的預測、基因突變……等。因此，不平衡資料集幾乎是難以避免的，我們做的是如何透過採集樣本的方式、調整學習演算法，讓資料探勘的結果不只是一味的追求高準確度，而是能夠在監督式機器學習中，有效的預測出實際資料集中樣本的特徵。

1.3. 不平衡資料集的特點

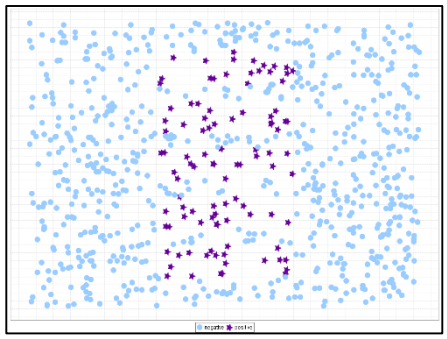
不平衡資料集會有以下的特點[i]:

(1) Small Disjuncts / Rare Data Set

由於很多的分類演算法是採用 Divide and Conquer，樣本會被慢慢的切割，導致 Small Disjuncts 的產生。只能在各個獨立的區塊中尋找數據的規律，對於小樣本數類別來說，每個區塊中包含了很少的樣本訊息，會讓跨區塊的訊息沒有辦法被找出來。

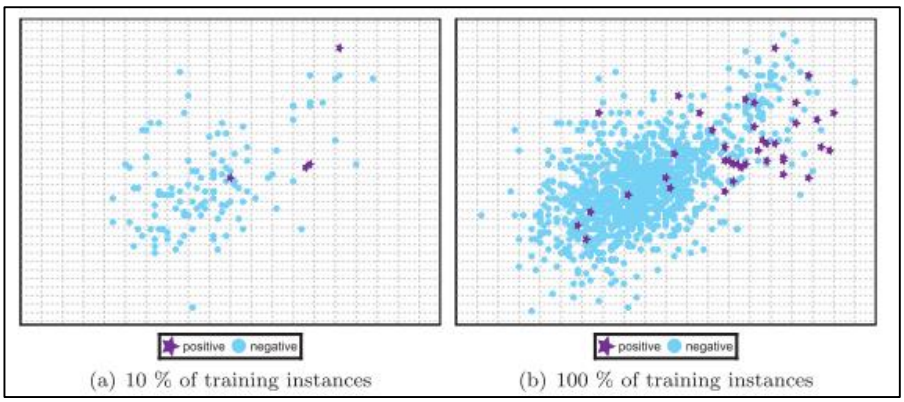


。



(2) Lack of Density

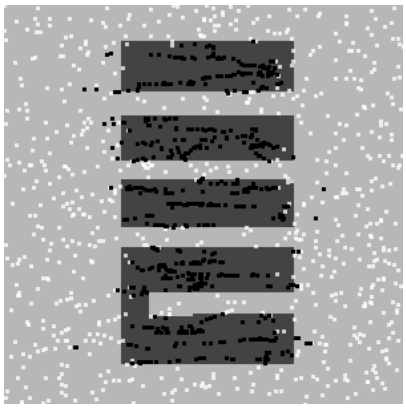
資料密度不足意味著資料集沒有辦法提供足夠的資訊做資料探勘，也就是樣本規模過小， 這樣會讓演算法沒有足夠的數據涵蓋樣本的分布。當小樣本類別的密度太低時，還會被當成是噪音數據。

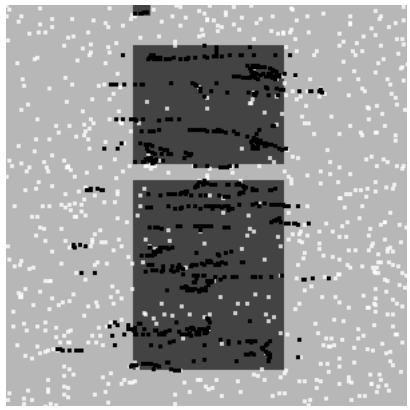


(a) 10 % of training instances (b) 100 % of training instances

(3) Noisy Data

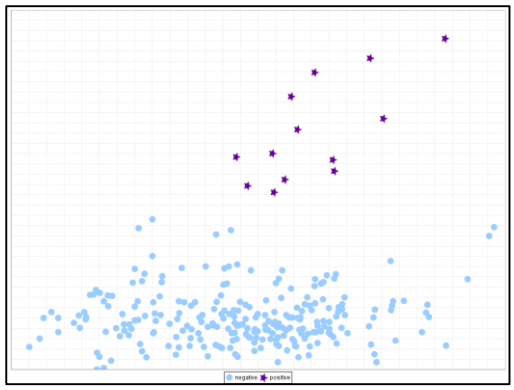
Noisy data 指的是資料集中沒有意義的資料，在資料探勘中，Noisy data 會影響探勘的結果，所以通常會被剃除在外。在不平衡資料集的情況下，Noisy Data 對少樣本數的類別的影響會更大，因為只要有很少的 Noisy Data 就會直接影響子概念的學習。下面兩張圖可以看到，在加入噪音數據以後，會產生錯誤的決策邊界。

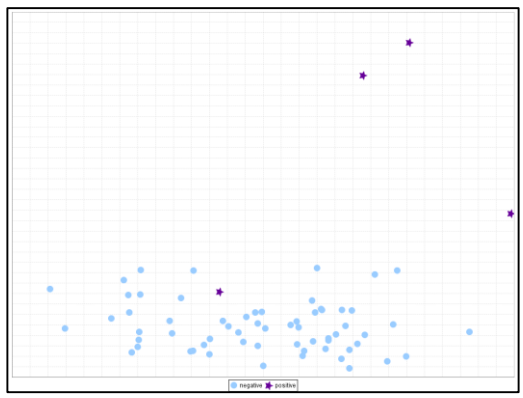




(4) 資料集的偏移(Dataset Shift)

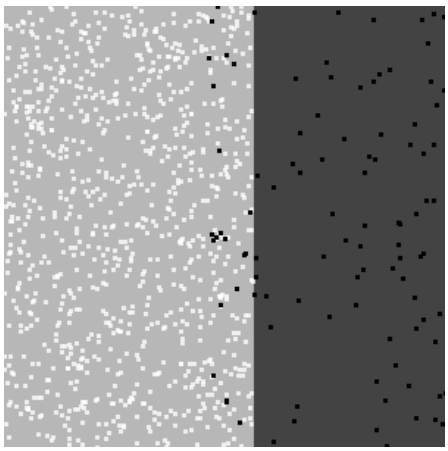
當 training 和 testing dataset 服從不同的分布時，就是資料集偏移的現象，在處理不平衡資料集時，Dataset Shift 會變得特別重要，因為小樣本數類別因為樣本數量很少，所以會對一個錯誤分類很敏感，導致 training 和 testing 性能上有差異。

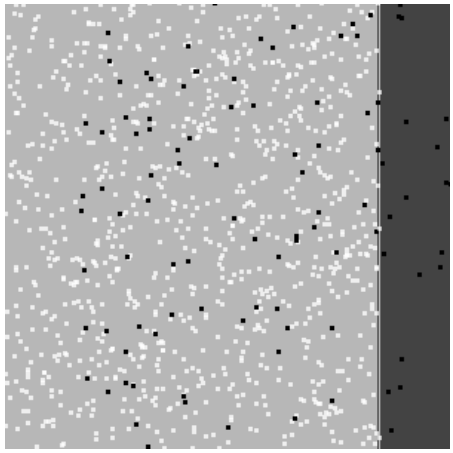




(5)類別重疊(Overlapping)

在一個數據集中的類別，倘若每個類別的訓練數據都相似的話，就會出現類別重疊的問題，隨著重疊程度的增加，分類的性能也會變差。

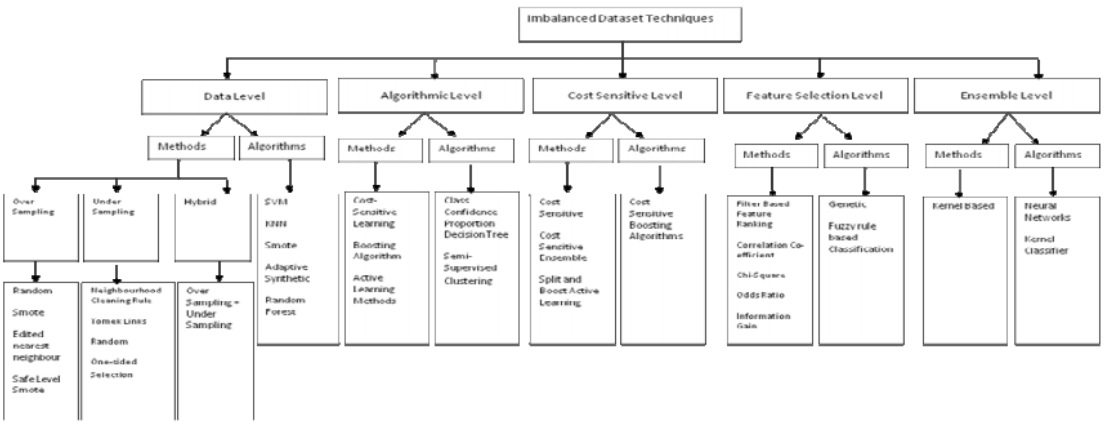




Overlap degree (%)	TP	TN	AUC
0	1.000	1.000	1.000
20	.79.00	1.000	.8950
40	.4900	1.000	.7450
50	.4700	1.000	.7350
60	.4200	1.000	.7100
80	.2100	.9989	.6044
100	.0000	1.000	.5000

2. 不平衡資料集實例

我們在 Kaggle 上找了一個 Imbalanced Dataset 的實例[ii]—Financial Distress Prediction，在 CSV 檔上可以看到，Financial Distress Column 小於-0.5 與大於-0.5 的 Firm year 比例為 136:3546，可以明顯的感覺到此資料集是一個不平衡資料集。面對這樣的不平衡資料集，我們要怎麼處理它呢?



3 如何處理不平衡資料集

處理不平衡資料集的方式可以歸納成五個等級[iii]:資料等級(Data Level)、演算法等級(Algorithmic Level)、代價敏感等級(Cost Sensitive Level)、特徵選擇等級(Feature selection level)、集成等級(Ensemble Level)。

3.1 資料等級

資料等級的方法主要是針對重新採集樣本(Resampling)，來處理不平衡資料集。資料等級的演算法常見的有 SMOTE、SVM、KNN…等。目前可以歸納成主要的三種方法:[iv]

(1)欠抽樣(Under-sampling):

透過減少多樣本數類別的樣本數量，讓各個類別的樣本數量盡量相近。可以直接隨機的丟棄多樣本數類別的樣本數量，但是必須要承擔因為捨棄掉樣本資訊而影響的結果。

(2)過抽樣(Over-sampling):

透過增加少樣本數類別的樣本數量，以實現樣本均衡，最簡單的方法就是複製少樣本數類別的樣本，這樣的缺點就是會有過度擬和(Overfitting)的現象，也就是在調適模型的過程中，可能使用過多的參數了。SMOTE[v](Synthetic Minority Oversampling Technique)演算法就是 Over-sampling 的改良方案，主要是對少樣本數類別進行分析，再將樣本人工合成新樣本新增到類別底下。

(3)混合(Hybrid)[vi]

混和前面兩者的特性，一邊合成新樣本到樣本數少的類別下，一邊刪除大樣本數類別的樣本。像是 SMOTE+Tomek 或是 SMOTE+ENN 都算是 Hybrid 的演算法。

3.2 演算法等級[vii]

透過直接套用現有的學習演算法，去調整不平衡資料集的偏差。

3.3 代價敏感學習[viii]等級

代價敏感學習主要在計算錯誤的分類所造成的代價(Cost)，並期望可以將機器學習過程中的代價盡可能降到最低，通常會把不同的代價存在 N*N 的矩陣中，N 是類別的個數。舉醫療的例子來說，“將病人誤診為健康人的代價”與“將健康人誤診為病人的代價”應該要不同。通常對小樣本賦予較高的代價，大樣本賦予較小的代價，期望以此來平衡樣本之間的數目差異。常見的代價敏感度的演算法有 Cost sensitive boosting。

3.4 特徵選擇等級[ix]

當樣本數量分布不平均時，特徵的分布也會不平均，而特稱選擇主要是在資料集中挑選出與目標比較有相關的特徵資料，並把其餘沒有辦法提供資訊的樣本淘汰掉，使判斷準確率能夠提升。常見的特徵選擇演算法有 Genetic、Fuzzy Rule。

3.5 集成學習等級[x]

集成學習指的是組合多個模型，以獲得更好的效果，常見的集成學習演算法有 Neural Network。主要有下列三種方法:

(1)在 Testing data set 中，找到表現最好的模型當作是最終的預測模型
(2)對多個模型的預設結果取平均值
(3)對多個模型的預測結果做加權平均

4.對不平衡資料集表現最好的學習演算法

經過幾天的學習下來，我們覺得針對不平衡資料集，並無一個表現最好或是最壞的學習演算法,應該要視資料集的狀況而定,也可以嘗試多個學習演算法並相互比較。我們認為，決策樹通常在不平衡資料集表現比較好，因為它可以評估每個特徵在分類問題上重要性。尤其是結合多個決策樹的隨機森林(Random Forest)，如果每個決策樹都是一個分類器，隨機森林就會是綜合所有的分類器的投票結果。

References

---

[i] Herrera, 「Classification with Imbalanced Data Sets」.
[ii] 「Financial Distress Prediction」.
[iii] Ramyachitra 及 Manikandan, 「IMBALANCED DATASET CLASSIFICATION AND SOLUTIONS: A REVIEW」.
[iv] www.dataivy.cn, 「3.4 解決樣本類別分佈不均衡的問題」, 4.
[v] 「不平衡資料分類 | 程式前沿」.
[vi] 「非均衡數據處理？」.
[vii] Krawczyk, 「Learning from Imbalanced Data」.
[viii] 「代價敏感學習- CSDN 博客」.
[ix] 「特徵選擇 Feature Selection · Machine Learning: Python 機器學習：使用 Pytho-n」.
[x] 「簡單易學的機器學習算法——集成方法(Ensemble Method) - CSDN 博客」.