基於關聯規則的分類演算法 (CBA) 詳解

摘要 (Abstract)

CBA (Classification Based on Associations) 是一種創新的資料探勘技術，它成功地將**關聯規則挖掘**的優勢應用於監督式**分類**任務。CBA 的核心是透過挖掘易於理解的 **類別關聯規則 (CARs)**，建立一個具有高準確度與強大可解釋性的預測模型。本文件將詳細介紹 CBA 演算法的原理、兩個主要階段及其在資料分類中的應用。

一、 CBA 簡介與核心概念

1.1 演算法概述

特性	說明
全名 (中)	基於關聯規則的分類
全名 (英)	Classification Based on Associations
提出者	Liu, Hsu, & Ma (1998)
所屬技術家族	關聯規則分類 (ARC)
核心目標	建立一個由 **IF-THEN 規則** 組成的、準確且透明的分類器。

1.2 核心基石：類別關聯規則 (CARs)

CBA 專注於挖掘 **類別關聯規則 (Class Association Rules, CARs)**。CARs 是一種特殊的關聯規則，其規則的後件 (Consequent) **固定**為單一的類別標籤。

$$Condset \quad \Rightarrow \quad Class$$

- **前件 (Condset)**：由一組屬性-值對組成 (例如： 年齡=年輕 且 收入=高)。
- **後件 (Class)**：資料集的目標類別標籤 (例如： 購買意願=高)。

二、 規則強度的衡量指標

在生成 CARs 之前，我們必須定義規則的「強度」。CBA 採用關聯規則領域的兩個標準指標：支持度 (Support) 和置信度 (Confidence)。

2.1 支持度 (Support, S)

支持度衡量規則中所有項目集 (前件和後件) 在整個資料集中共同出現的頻率。

$$S(Condset \Rightarrow Class) = P(Condset \cup Class)$$

- **作用**：代表規則的**重要性**和**普遍性**。

2.2 置信度 (Confidence, C)

置信度衡量在滿足規則前件 (Condset) 的前提下，後件 (Class) 也成立的條件機率。

$$C(Condset \Rightarrow Class) = P(Class \mid Condset)$$

- **作用**：代表規則的**預測可靠性**。

門檻要求：只有同時滿足使用者定義的最小支持度 (MinSup) 和最小置信度 ($\mathrm{MinConf}$) 的規則，才能成為候選的 CARs。

三、 CBA 演算法的兩大階段

CBA 是一個兩階段過程：先生成所有合格的規則，然後將其縮減並組合成一個高效的分類器。

3.1 階段 I：關聯規則生成 (CBA-RG)

此階段的目標是找出所有滿足 MinSup 和 $\mathrm{MinConf}$ 的 **CARs 集合** R。

1. **資料準備**：
   - 將所有連續型數值屬性離散化（轉換為區間或標稱值）。
   - 將每一筆資料轉換為類似購物籃分析中的「交易」記錄。
2. **規則挖掘**：
   - 使用 Apriori 演算法或其優化版本進行挖掘。
   - 挖掘過程與一般關聯規則挖掘類似，但重點是確保規則的後件是類別屬性。

3.2 階段 II：分類器建立 (CBA-CB)

此階段的目標是從大量的 R 集合中，選出一個最小且最優的規則子集 C 來建立分類器。

1. **規則排序 (Rule Ordering)**：將所有 CARs 根據以下優先級進行降序排序：
   - 置信度 (最高優先)
   - 支持度 (次高優先)
   - 規則長度 (最短優先)
2. **規則剪枝與選擇 (Pruning and Selection)**：
   - CBA 採用一種**貪婪覆蓋 (Greedy Covering)** 的剪枝策略（常見的有 M1 和 M2 版本）。
   - **核心邏輯**：從排序列表頂端開始，依序將規則納入分類器 C。每納入一條規則，將所有被該規則**正確分類**的訓練實例從資料集中移除。
   - 這個過程確保了分類器中的每條規則都對分類準確性有所貢獻，並移除冗餘規則。
3. **預設規則 (Default Rule)**：
   - 在規則列表的末尾添加一條預設規則 r_{default}。
   - **作用**：用於分類那些未被前面任何 CARs 覆蓋的新實例。
   - **類別**：通常設定為剩餘未被覆蓋實例中最常見的類別。

四、 CBA 分類器的運作與應用

4.1 最終分類器結構

最終的 CBA 分類器 C 是一個**有序的規則列表**：

$$C = \{r_1, r_2, r_3, \ldots, r_k, r_{\text{default}}\}$$

4.2 分類預測機制 (The "First Match" Principle)

對於一個新的資料實例 t (待分類記錄)，CBA 分類器按照規則列表的順序進行預測：

1. 從 r_1 開始，依序檢查 t 是否**匹配**當前規則 r_i 的前件條件。
2. **當 t 匹配到第一條規則 r_i 時，無論是 CARs 還是 r_{default}，該規則的後件類別即為 t 的最終預測結果。**
3. 後續的規則將不再檢查。

這個「第一次匹配」機制確保了模型的高效性和確定性。

五、 結論與建議

CBA 演算法在學術界和工業界都具有重要價值，特別是在需要模型透明度的領域。

5.1 主要優點總結

- **高可解釋性**：規則是人類可讀的 IF-THEN 語句。
- **高準確度**：能與主流分類演算法競爭。
- **強大的資料探勘基礎**：利用了成熟的關聯規則挖掘技術。

5.2 應用建議

CBA 特別適合應用於：

1. **醫療診斷**：根據病人的症狀組合 (前件) 預測疾病 (後件)。
2. **市場籃子分析的延伸**：根據顧客的購買歷史 (前件) 預測他們下次會購買的產品類別 (後件)。
3. **金融風控**：根據客戶的行為模式 (前件) 預測其信用風險等級 (後件)。

六、 範例說明 (Example Illustration)

我們將使用一個小型貸款核准資料集，演示 CBA 的兩個階段。

6.1 訓練資料集

假設我們有 6 筆客戶的貸款申請資料：

ID	收入 (Income)	信用評分 (Credit Score)	學生 (Student)	**核准貸款 (Class)**
1	高	好	否	是
2	高	普通	否	否
3	中	好	是	是
4	低	好	是	否

5	中	差	否	否
6	低	普通	是	是

設定門檻：

- 總實例數 (N) = 6
- 最小支持度 (MinSup)：$2/6 \approx 33\%$ (即出現次數 ≥ 2)
- 最小置信度 (MinConf)：60%

6.2 階段 I：CARs 生成 (CBA-RG)

目標是找出所有滿足 $\text{MinSup} \geq 2$ 且 $\text{MinConf} \geq 60\%$ 的 CARs。

規則 (R)	前件支援數 (Count)	規則支援數 (Support Count)	置信度 (Confidence)	結果
R1: $\{學生 = 否\} \Rightarrow \{C = 否\}$	3 (ID 1, 2, 5)	2 (ID 2, 5)	$2/3 \approx 66.7\%$	**合格**
R2: $\{學生 = 是\} \Rightarrow \{C = 是\}$	3 (ID 3, 4, 6)	2 (ID 3, 6)	$2/3 \approx 66.7\%$	**合格**
R3: $\{信用評分 = 好\} \Rightarrow \{C = 是\}$	4 (ID 1, 3, 4)	2 (ID 1, 3)	$2/4 = 50\%$	不合格 (<60%)
R4: $\{收入 = 高\} \Rightarrow \{C = 否\}$	2 (ID 1, 2)	1 (ID 2)	$1/2 = 50\%$	不合格 (<60%)

候選規則集 $R = \{R1, R2\}$

6.3 階段 II：分類器建立 (CBA-CB)

初始資料 D: {ID 1, 2, 3, 4, 5, 6}

1. **規則排序**：$R1$ 和 $R2$ 具有相同的置信度 (66.7%)、支持度 (33.3%) 和長度 (1)。我們按 $R1, R2$ 的順序處理。
2. **貪婪覆蓋與剪枝 (M1 策略)**：

規則 (r)	納入 C	r 涵蓋實例 (Covered)	r 正確分類實例 (Correct)	移除實例	剩餘 D
R1: $\{學生 = 否\} \Rightarrow \{C = 否\}$	**是**	{1, 2, 5}	{2, 5}	{2, 5}	{1, 3, 4, 6}
R2: $\{學生 = 是\} \Rightarrow \{C = 是\}$	**是**	{3, 4, 6}	{3, 6}	{3, 6}	{1, 4}

3. **預設規則 (Default Rule)**：
   - 剩餘未覆蓋的實例 D' 為 {ID 1 (Class: 是), ID 4 (Class: 否)}。
   - 兩類別數量相等 (各 1)。假設遇到平手時，預設選擇第一條規則較少分類錯誤的類別。在此範例中，我們選擇**「是」**。
   - R_{default}：$\{\} \Rightarrow \{C = 是\}$

6.4 最終分類器與預測

最終分類器 C (有序列表)：

1. r_1: $\{\text{學生} = \text{否}\} \Rightarrow \{\text{C} = \text{否}\}$
2. r_2: $\{\text{學生} = \text{是}\} \Rightarrow \{\text{C} = \text{是}\}$
3. r_{default}: $\{\} \Rightarrow \{\text{C} = \text{是}\}$

預測一個新實例 t：

- **新客戶 t_{new}**: (收入=中, 信用評分=好, 學生=否)
  - 檢查 r_1：**匹配** $(\text{學生} = \text{否})$。
  - **預測結果：否 (不核准貸款)。**
- **新客戶 t'_{new}**: (收入=高, 信用評分=差, 學生=是)
  - 檢查 r_1：不匹配 $(\text{學生} = \text{是})$。
  - 檢查 r_2：**匹配** $(\text{學生} = \text{是})$。
  - **預測結果：是 (核准貸款)。**