Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques

2014/8, Computer Science & Engineering Department, Institute of Technology.

Nirma University, Ahmedabad, Gujarat, India.

**組員：**

**112352018 李彥霖**

**112352012黃柏翔**

**112352014 劉冠銘**

**Mentor：**

**阮彥勳 博士**

資料取得

1. 標的：以下列出該論文挑選的四檔標的以及本組研究台灣市場修正標的

1. Nifty 印度前50大公司 - > 加權指數

2. s and p 追蹤前30家公司 - > 0050

3. Infosys 印度第二大IT公司 - > 2330

4. Reliance 印度第二大私營集團 - > 2881

1. 資料期間：2003/7至2012/12，較論文晚了六個月，因為0050在2003/7才上市有股價可以研究。
2. 資料來源：TEJ

研究方法

1. 資料集拆分

　　原論文將訓練集與測試集分別以50%的比例拆分，並將其中上漲與下跌的比例調整至大致相等，我們認為這樣的作法會導致測試結果不符合現實情況的機率分布。因此，除了依照論用完全隨機的上漲下跌比例，確保測試結果較貼近實際市場情形。

1. 技術指標

論文中採用十種技術指標做為模型的訓練特徵，整理如下：

* 簡單移動平均（Simple Moving Average, SMA）
* 加權移動平均（Weighted Moving Average, WMA）
* 動能指標（Momentum, MOM）
* KD隨機指標（Stochastic Oscillator, %K and %D）
  + %K
  + %D
* 相對強弱指數（Relative Strength Index, RSI）
* 移動平均收斂發散指標（Moving Average Convergence Divergence, MACD）
* 威廉指標（Larry William’s %R, W%R）
* 累積分配指標（Accumulation/Distribtion Oscillator, A/D）
* 商品通道指數（Commodity Channel Index, CCI）

其中：

* 為時間點t的收盤價

1. 特徵工程

　　各項技術指標的數值分布差距太大，如下表所示。因此，在投入模型訓練前，需要先做標準化，將各項數值轉換為標準常態分布。

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SMA** | **WMA** | **MOM** | **K** | **D** | **RSI** | **MACD** | **WILLR** | **AD** | **CCI** |
| mean | 7110.16 | 7111.6 | 9.8327 | 56.260 | 56.260 | 53.104 | 6.7503 | -43.739 | 0.5069 | 12.941 |
| std | 1161.87 | 1162.0 | 296.83 | 32.305 | 23.297 | 15.242 | 113.43 | 32.305 | 0.3430 | 105.38 |

　　原論文採用兩種方法進行模型訓練與測試，分別為連續特徵與離散特徵。方法一直接使用標準化後的資料投入模型，方法二根據事先決定的規則，將各項技術指標轉換為離散特徵。其轉換規則如下表：

|  |  |  |
| --- | --- | --- |
| **技術指標** | **上漲（+1）** | **下跌（-1）** |
| SMA | 目前價格 >= 移動平均數值 | 目前價格 < 移動平均數值 |
| WMA | 目前價格 >= 移動平均數值 | 目前價格 < 移動平均數值 |
| Momentum | 動能數值為正 | 動能數值為負 |
| RSI | RSI < 30;  30 <= RSI <= 70 且  t時間的數值 >= t-1時間的數值 | RSI > 70;  30 <= RSI <= 70 且  t時間的數值< t-1時間的數值 |
| CCI | CCI > 200;  -200 <= CCI <= 200 且  t時間的數值 >= t-1時間的數值 | CCI < -200;  -200 <= RSI <= 200 且  t時間的數值< t-1時間的數值 |
| K% | t時間的數值 >= t-1時間的數值 | t時間的數值< t-1時間的數值 |
| D% | t時間的數值 >= t-1時間的數值 | t時間的數值< t-1時間的數值 |
| MACD | t時間的數值 >= t-1時間的數值 | t時間的數值< t-1時間的數值 |
| W%R | t時間的數值 >= t-1時間的數值 | t時間的數值< t-1時間的數值 |
| A/D | t時間的數值 >= t-1時間的數值 | t時間的數值< t-1時間的數值 |

模型介紹與參數應用

1. ANN
   1. 模型解釋：人工神經網路（ANN）模型受到生物神經網路的啟發，使用了三層前饋神經網路。模型由輸入層、隱藏層和輸出層組成。輸入層包括十個技術指標，對應到十個神經元。隱藏層中的每個神經元使用 tan sigmoid 作為傳遞函數。輸出層包含一個神經元，使用 log sigmoid 作為傳遞函數，生成在 0 到 1 之間的連續值。通過梯度下降與動量調整權重，透過全面的參數設定實驗來優化模型。
   2. 超參數設定：

* 隱藏層神經元數：(n): 10, 20, ..., 100
  + 這個超參數決定了隱藏層中神經元的數量。增加神經元數可能增加模型的複雜度，但也可能導致過擬合。適當的數量通常需要透過實驗和調整來確定。
* Epochs (ep): 1000, 2000, ..., 10,000
  + 指整個數據集在訓練過程中被使用的次數。這控制模型訓練的迭代次數。太少的 Epochs 可能導致模型未能完全收斂，而太多可能導致過擬合。
* Momentum constant (mc): 0.1, 0.2, ..., 0.9
  + 動量是一種幫助優化算法避免陷入局部極小值的技術。這個超參數控制先前更新的權重對當前更新的影響程度。
* Learning rate (lr): 0.1
  + 學習率控制權重更新的步長。太大的學習率可能導致不穩定的訓練，而太小可能導致訓練速度過慢或陷入局部極小值。

1. Random Forest
   1. 模型解釋： 隨機森林是一種集成學習算法，使用多個決策樹作為基學習器。模型通過創建多棵樹，每棵樹使用Bootstrap樣本和隨機選擇的特徵，最終輸出為多數樹的共同結果。這有助於防止過擬合。
   2. 超參數設定：

* 森林中樹的數量 (n\_trees): 10, 20, ..., 200
  + 決定了森林中有多少決策樹。增加樹的數量通常可以提高模型的性能，但也可能增加計算成本。`

1. SVM
   1. 模型解釋： 支持向量機（SVM）是一種在高維特徵空間中操作的學習系統，其主要目標是找到最大邊緣超平面。模型使用多項式和径向基核函數，並通過解決二次規劃問題找到最大邊緣超平面。
   2. 超參數設定：

* 多項式核的核函數次數 (d): 1, 2, 3, 4
  + 多項式核的次數。增加次數可能使模型更適應複雜的數據，但也可能導致過擬合。
* 径向基核的核函數gamma (c): 0.5, 1.0, 1.5, ..., 10.0
  + 控制徑向基函數的寬度。較大的 gamma 將導致更尖銳的峰值，可能產生更複雜的決策邊界。
* 正則化參數 (c): 0.5, 1, 5, 10, 100
  + 正則化參數控制邊界平滑度。較大的 c 將導致更精確的擬合，但可能增加過擬合的風險。
  1. Kernel Functions：
* Polynomial Function：
* Radial Basis Function：

1. Naïve Bayes Classifier
   1. 模型解釋： 樸素貝葉斯分類器基於對類條件獨立性的假設，使用貝葉斯定理預測數據屬於特定類別的概率。通過假設屬性值在給定類別標籤的情況下是相互獨立的，簡化了計算。根據貝葉斯定理，對於給定的測試數據 X，模型預測它屬於每個類別的概率。
   2. 超參數設定：本篇並未為此模型設定超參數

研究結果

研究結果將會以下面幾點來進行比較：

* 1. 模型之間的表現
  2. 資料預處理對表現的影響
  3. 訓練資料的分布對表現的影響

進而去對不同模型與不同方法來進行評論。

1. ANN
   1. Evenly distribute data each year

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0050 | 2330 | 2881 | TX |
| M1\_accuracy | 73.82% | 77.20% | 78.80% | 81.03% |
| M1\_f1 | 72.67% | 77.32% | 78.51% | 80.45% |
| M2\_accuracy | 96.76% | 95.66% | 96.19% | 94.84% |
| M2\_f1 | 96.74% | 95.61% | 96.14% | 94.84% |

* 1. Randomly distribute data each year

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0050 | 2330 | 2881 | TX |
| M1\_accuracy | 50.94% | 50.37% | 53.21% | 52.83% |
| M1\_f1 | 67.49% | 66.99% | 69.46% | 69.13% |
| M2\_accuracy | 51.83% | 50.28% | 51.10% | 53.77% |
| M2\_f1 | 68.27% | 66.91% | 67.64% | 69.93% |

可從上面兩張表看出Method 2 有較佳的預測準確度，而test資料集改成隨機分配之後，表現出現了大幅的下降。

1. SVM(Polynomial Function)
   1. Evenly distribute data each year

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0050 | 2330 | 2881 | TX |
| M1\_accuracy | 64.36% | 63.32% | 64.88% | 63.12% |
| M1\_f1 | 67.89% | 68.42% | 57.46% | 54.89% |
| M2\_accuracy | 73.87% | 76.68% | 71.50% | 73.79% |
| M2\_f1 | 78.52% | 74.61% | 70.49% | 72.94% |

* 1. Randomly distribute data each year

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0050 | 2330 | 2881 | TX |
| M1\_accuracy | 50.94% | 50.37% | 53.21% | 52.83% |
| M1\_f1 | 67.49% | 66.99% | 69.46% | 69.13% |
| M2\_accuracy | 51.83% | 50.28% | 51.10% | 53.77% |
| M2\_f1 | 68.27% | 66.91% | 67.64% | 69.93% |

可從上面兩張表看出Method 2 有較佳的預測準確度，而test資料集改成隨機分配之後，表現出現了小幅的下降。

1. SVM(Radial Basis Function)
   1. Evenly distribute data each year

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0050 | 2330 | 2881 | TX |
| M1\_accuracy | 59.73% | 63.57% | 66.80% | 67.55% |
| M1\_f1 | 43.03% | 57.04% | 64.38% | 61.89% |
| M2\_accuracy | 91.01% | 89.45% | 85.47% | 87.93% |
| M2\_f1 | 90.37% | 90.20% | 86.61% | 88.47% |

* 1. Randomly distribute data each year

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0050 | 2330 | 2881 | TX |
| M1\_accuracy | 75.47% | 77.90% | 75.55% | 77.14% |
| M1\_f1 | 76.64% | 78.76% | 78.02% | 78.45% |
| M2\_accuracy | 93.76% | 97.97% | 95.79% | 94.09% |
| M2\_f1 | 94.01% | 98.00% | 95.94% | 94.55% |

可從上面兩張表看出Method 2比起 Method 1，有相對優秀的預測準確度，而test資料集改成隨機分配之後，表現出現了小幅的提升。

1. Random forest
   1. Evenly distribute data each year

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0050 | 2330 | 2881 | TX |
| M1\_accuracy | 74.25% | 79.16% | 78.96% | 78.94% |
| M1\_f1 | 72.88% | 77.86% | 79.21% | 78.71% |
| M2\_accuracy | 95.60% | 97.23% | 94.03% | 94.27% |
| M2\_f1 | 95.52% | 97.20% | 93.89% | 94.13% |

* 1. Randomly distribute data each year

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0050 | 2330 | 2881 | TX |
| M1\_accuracy | 78.59% | 79.65% | 80.04% | 79.71% |
| M1\_f1 | 79.06% | 79.71% | 81.14% | 80.90% |
| M2\_accuracy | 94.11% | 98.34% | 96.43% | 94.86% |
| M2\_f1 | 94.31% | 98.36% | 96.56% | 95.24% |

可從上面兩張表看出Method 2比起 Method 1，有相對優秀的預測準確度，而test資料集改成隨機分配之後，表現並沒有太大的差異。

1. Naïve Bayes Classfier
   1. Evenly distribute data each year

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0050 | 2330 | 2881 | TX |
| M1\_accuracy | 70.47% | 72.93% | 73.42% | 71.87% |
| M1\_f1 | 71.97% | 73.56% | 75.42% | 74.38% |
| M2\_accuracy | 92.41% | 95.30% | 93.85% | 90.99% |
| M2\_f1 | 92.63% | 95.35% | 94.12% | 91.30% |

* 1. Randomly distribute data each year

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0050 | 2330 | 2881 | TX |
| M1\_accuracy | 70.47% | 73.11% | 72.71% | 70.21% |
| M1\_f1 | 72.11% | 73.45% | 74.40% | 73.11% |
| M2\_accuracy | 91.08% | 95.03% | 94.14% | 92.29% |
| M2\_f1 | 91.18% | 95.05% | 94.27% | 92.75% |

可從上面兩張表看出Method 2比起 Method 1，有相當優秀的預測準確度，而test資料集改成隨機分配之後，表現並沒有太大的差異。

1. 總整理
   1. Evenly distribute data each year

|  |  |  |
| --- | --- | --- |
| **RF** |  | Average |
| M1\_accuracy | 77.83% |
| M1\_f1 | 77.17% |
| M2\_accuracy | 95.28% |
| M2\_f1 | 95.19% |

|  |  |  |
| --- | --- | --- |
| SVM\_P |  | Average |
| M1\_accuracy | 63.92% |
| M1\_f1 | 62.17% |
| M2\_accuracy | 73.96% |
| M2\_f1 | 74.14% |

|  |  |  |
| --- | --- | --- |
| SVM\_R |  | Average |
| M1\_accuracy | 64.41% |
| M1\_f1 | 56.58% |
| M2\_accuracy | 88.47% |
| M2\_f1 | 88.91% |

|  |  |  |
| --- | --- | --- |
| ANN |  | Average |
| M1\_accuracy | 77.71% |
| M1\_f1 | 77.24% |
| M2\_accuracy | 95.86% |
| M2\_f1 | 95.83% |

|  |  |  |
| --- | --- | --- |
| NB |  | Average |
| M1\_accuracy | 72.33% |
| M1\_f1 | 74.11% |
| M2\_accuracy | 93.18% |
| M2\_f1 | 93.38% |

* 1. Randomly distribute data each year

|  |  |  |
| --- | --- | --- |
| RF |  | Average |
| M1\_accuracy | 79.50% |
| M1\_f1 | 80.20% |
| M2\_accuracy | 95.94% |
| M2\_f1 | 96.12% |

|  |  |  |
| --- | --- | --- |
| SVM\_P |  | Average |
| M1\_accuracy | 51.83% |
| M1\_f1 | 68.27% |
| M2\_accuracy | 51.74% |
| M2\_f1 | 68.19% |

|  |  |  |
| --- | --- | --- |
| SVM\_R |  | Average |
| M1\_accuracy | 76.51% |
| M1\_f1 | 77.97% |
| M2\_accuracy | 95.40% |
| M2\_f1 | 95.62% |

|  |  |  |
| --- | --- | --- |
| ANN |  | Average |
| M1\_accuracy | 50.42% |
| M1\_f1 | 50.99% |
| M2\_accuracy | 51.74% |
| M2\_f1 | 68.19% |

|  |  |  |
| --- | --- | --- |
| NB |  | Average |
| M1\_accuracy | 71.63% |
| M1\_f1 | 73.27% |
| M2\_accuracy | 93.14% |
| M2\_f1 | 93.31% |

模型總結 :

1. 方法2（使用趨勢確定性）表現較佳。
2. 隨機分割數據可能更現實。可以觀察到對於SVM\_p和ANN兩者的性能都有顯著的下降，而其他模型的性能保持相似。
3. 在均勻分佈數據的驗證下，ANN和RF表現最佳。然而，在隨機分佈數據的驗證下，RF仍然保持最佳性能，而ANN則表現最差。