Recursive Stock Picking with Decision Trees

Anton Andriyashin

C.A.S.E. — Center for Applied Statistics and Economics
Chair of Statistics
Humboldt-Universität zu Berlin
http://ise.wiwi.hu-berlin.de
**Economic Modeling vs Object Recognition**

Stock picking frequently comes as the result of rigorous asset pricing models and simulations of dynamic general equilibria.

- Economic agent behavior is to be limited by a certain objective functional and relevant budget constraints
- Agents are supposed to behave rationally on the market

Object recognition restricts the class of equilibrium trajectories and may not impose additional modeling assumptions.

- No explicit model is assumed
- Solution class is restrained (e.g. quadratic functions class)
Basic Pricing Equation

\( u(\cdot) \) – utility function, \( c_t \) – consumption at time \( t \), \( p_t \) – stock price at time \( t \), \( d_{t+1} \) – dividends paid at time \( t \), \( x_{t+1} \) – next period payoff.

\[
\max_{\xi} u(c_t) + E_t[\beta u(c_{t+1})] \quad s.t. \\
\quad c_t = e_t - p_t \xi, \\
\quad c_{t+1} = e_{t+1} + x_{t+1} \xi, \\
\quad x_{t+1} = p_{t+1} + d_{t+1}
\]

Solution: \( p_t = E_t[\beta \frac{u'(c_{t+1})}{u'(c_t)} x_{t+1}] \)
Bloomberg Report

The benchmark DAX Index slid 42.16, or 0.7 percent, to 6012.56 at 9:46 a.m. in Frankfurt. "European stocks are set to take some hits from the weakening dollar, prospects for interest rates and oil prices near records", said Christian Wrede. [...] Credit Suisse said investors should cut their holdings in continental European equities, citing the rising euro and higher interest rates. Oil prices added 1.7 percent yesterday to $73.32 a barrel in New York after climbing as high as $73.90. Futures slipped back to $72.76 today.

Factors: euro/dollar exchange rate, interest rates, and oil spot and future prices.
Relevant Studies in the Area

**SmithBarney** (1999): average return – 19.62% (annualized), st. deviation – 11.96% and Sharpe ratio – 1.23

**JPMorgan** (2003): average return – 14.60% (annualized), st. deviation – 9.50% and Sharpe ratio – 1.54
What is a Decision Tree?

Recursive Stock Picking with Decision Trees
Defining Stock Classes

Three stock classes are assumed at the moment, each class corresponds to one of possible stock positions:

- short,
- neutral,
- long

Let $P_t$ be the current stock price, $R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$, let $\bar{R}$ be some positive threshold.

If $R_t > \bar{R}$, the stock is assumed to be undervalued at $t - 1$
If $R_t < -\bar{R}$, the stock is assumed to be overvalued at $t - 1$
If $-\bar{R} \leq R_t \leq \bar{R}$, the stock is assumed to be fairly priced at $t - 1$
Setting Up a Classification Problem – I

If $t$ is the current time period, then $R_{t+1}$ – the stock price movement at $t + 1$ – is of one’s interest.

For fixed $t$:
Let $X \in \mathbb{R}^p$ be the factor space of explanatory variables (fundamental, technical, macroeconomic indicators and stock prices).
Let $Y \in \mathbb{R}^1$ be the vector of stock classes.

$\forall t : Y_t \in \{Short, Neutral, Long\}.$
Setting Up a Classification Problem – II

From known market data one can obtain:

\[
\{X_0, X_1\} \rightarrow Y_0 \\
\ldots \\
\{X_{t-2}, X_{t-1}\} \rightarrow Y_{t-2} \\
\{X_{t-1}, X_t\} \rightarrow Y_{t-1}
\]

Stock picking implies unknown next period price:

\[
\{X_t\} \rightarrow Y_t
\]
Problem Statement

**XETRA DAX Example: BMW**

Recursive Stock Picking with Decision Trees
How to Pick the Stock?

Market data from the past along with stock prices constitute the learning sample. Individual stock price movements are predicted separately using different trees.

Major steps:

- use the data from the past to build the decision tree,
- input the available data from today \((X_t)\) into the tree,
- get the prediction of the class of \(R_{t+1} = \frac{P_{t+1} - P_t}{P_t}\)

The estimation is performed in the class of orthogonal step functions
Parent and Child Nodes

Let $t_P$ be the parent node, $t_L$ and $t_R$ – children nodes.

Let $n_P$ be the number of observations in $t_P$, $n_L$ and $n_R$ – in $t_L$ and $t_R$ respectively.

\[
p_L = 1 - p_R = \frac{n_L}{n_P},
\]

\[
p(j|t) = \frac{n_t(j)}{n_t}
\]
**Impurity Function**

Node **homogeneity** is measured by the *impurity function* $i(t)$.

$$\Delta i(t) = i(t_P) - E\{i(t_C)\}$$

where $E(\cdot)$ is the expected value operator:

$$E\{i(t_C)\} = p_L i(t_L) + p_R i(t_R)$$

and $C = \{L, R\}$.

**$s^*$ – optimal split**: question variable and its value

$$s^* = \arg\min_s \{p_L i(t_L) + p_R i(t_R)\}$$
Getting the Right Values for the Tree Questions

To get $s^*$, $i(t)$ must be explicitly defined.

Gini index:

$$i(t) = \sum_{k \neq 1}^J \sum_{l \neq k}^J p(k|t)p(l|t)$$

where $k, l = 1, J$ are class indices.

$$s^* = \arg\max_s \left\{ - \sum_{j=1}^J p^2(j|t) + p_L \sum_{j=1}^J p^2(j|t_L) + p_R \sum_{j=1}^J p^2(j|t_R) \right\}$$
The Challenge of Optimum Tree Size

Maximum tree $T_{MAX}$ – there are no observations left to split.

- easy to build,
- perfect in-sample prediction rate,
- poor out-of-sample rate: overfitting

Too small and primitive trees lead to underfitting.

Therefore, some tradeoff is required.
Change of Impurity Rule

Since the change of impurity value is maximized at every step, select those node that lead only to significant fluctuations of $i(t)$. Enable split $s$, if for a given node $t$

$$\Delta i(s, t) \geq \bar{\beta}$$

However, the following challenges do arise:

- $\Delta i(s, t)$ is usually not a monotone function
- how should one define $\bar{\beta}$?
**V-fold Cross-Validation Measure of Tree Quality – I**

Define $L \setminus L_v$, $\forall v = 1, V$ as the *training set* and $L_v$ as the *test set* where $L$ is the *learning sample* itself. For a given classification rule $d_v$ derived from the learning set $L \setminus L_v$, its quality can be represented in the following way:

$$E^1 (d^{(v)}) = \frac{1}{N_v} \sum_{(l_n, j_n) \in L_v} I (d^{(v)}(l_n) \neq j_n)$$

where $l_n$ is a test set observation with class $j_n$, $N_v$ – the number of observations in $L_v$ and $E^1 (d^{(v)})$ is the *one-iteration internal misclassification error estimate*. 

Recursive Stock Picking with Decision Trees
$V$-fold Cross-Validation Measure of Tree Quality – II

Note that none of the observations from $L_v$ was engaged during construction of decision rule $d^{(v)}$.

$$E^{CV}(d) = \frac{1}{V} \sum_{v=1}^{V} E^1\left(d^{(v)}\right)$$

where $E^{CV}(d)$ is the $V$-fold cross-validation measure of tree quality.

This brute force approach is, however, rather slow.
Internal Misclassification Error

Define *internal misclassification error* of an arbitrary observation at node $t$ as

$$e(t) = 1 - \max_j p(j | t)$$

Let us define also

$$E(t) = e(t)p(t)$$

Then *internal misclassification tree error* is

$$E(T) = \sum_{t \in \tilde{T}} E(t)$$

where $\tilde{T}$ is a set of terminal nodes.
Cost-Complexity Function

Let $|\tilde{T}|$ be the number of terminal nodes – the measure of complexity of tree $T$.

Cost-complexity function is defined as:

$$E_\alpha(T) = E(T) + \alpha |\tilde{T}|$$

where $\alpha \geq 0$ is the complexity parameter and $\alpha |\tilde{T}|$ is the cost component.
Cost-Complexity Function Optimization

Although $\alpha$ can have infinite number of values, the number of subtrees of $T_{MAX}$ resulting in minimization of $E_\alpha(T)$ is finite.

Pruning of $T_{MAX}$ leads to the creation of subtree sequence $T_1, T_2, T_3, \ldots$ with a decreasing number of terminal nodes.

It can be shown that $\forall \alpha \geq 0$ there exists an optimal tree $T(\alpha)$ in the sense that

1. $E_\alpha \{ T(\alpha) \} = \min_{T \leq T_{MAX}} E_\alpha(T) = \min_{T \leq T_{MAX}} \left[ E(T) + \alpha \left\| \tilde{T} \right\| \right]$

2. If $E_\alpha(T) = E_\alpha(T(\alpha))$ then $T(\alpha) \leq T$. 

Recursive Stock Picking with Decision Trees
Cross-Validation and the Optimum Tree

This way one can get a sequence of optimal subtrees
\( T_{\text{MAX}} \succ T_1 \succ T_2 \succ T_3 \succ \ldots \succ \{t_0\}. \)

The sequence \( \{\alpha_k\} \) is increasing. For \( k \geq 1: \alpha_k \leq \alpha < \alpha_{k+1} \) and
\( T(\alpha) = T(\alpha_k) = T_k \) and \( \alpha_1 = 0. \)

Applying the method of \( V \)-fold cross-validation to the sequence
\( T_{\text{MAX}} \succ T_1 \succ T_2 \succ T_3 \succ \ldots \succ \{t_0\}, \) the optimal tree is then determined.

Performance gain: sequence of optimal subtrees vs all possible subtrees.
Cross-Validation Live Example: BMW

- \( \Delta \text{Div. Yield} < 0.006 \) (28,32)
  - Class BUY (9,23)
  - CPI < 103.350 (6,4)
    - Class BUY (1,4)
    - Class SELL (5,0)
  - \( \Delta \text{Ext. Debt} < 0.092 \) (3,19)
    - Class BUY (0,16)
    - Class SELL (3,3)
  - Class SELL (19,9)
    - \( \Delta \text{Nat. Reserves} < -0.031 \) (19,6)
      - Class BUY (0,2)
      - Ext. Debt < 103331.500 (19,4)
        - Div. Yield < 1.915 (17,1)
          - Class SELL (16,0)
          - Class SELL (1,1)
        - Class BUY (2,3)
Data

Two different setups – with and without *Implied Volatility* variable – are considered.

- XETRA DAX companies;
- November 27, 2000 – October 20, 2003;
- normalized time scale: weekly observations;
- stock prices, technical and fundamental indicators from the past
## Available Variables

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Type</th>
<th>Frequency</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close Price</td>
<td>-</td>
<td>1 day</td>
<td>Closing price of a stock</td>
</tr>
<tr>
<td>Momentum</td>
<td>Technical</td>
<td>1 day</td>
<td>$M_t = P_t - P_{t-T}, \ T = 20$</td>
</tr>
<tr>
<td>Stochastic</td>
<td>Technical</td>
<td>1 day</td>
<td>$\frac{P_t - P_{t-T}}{P_H - P_L}, \ P_H = \max(P_t), \ P_L = \min(P_t)$</td>
</tr>
<tr>
<td>MA</td>
<td>Technical</td>
<td>1 day</td>
<td>$MA(T) = \frac{\sum_{i=t-T}^{t} P_i}{T}, \ T = 12$</td>
</tr>
<tr>
<td>MA St. Error</td>
<td>Technical</td>
<td>1 day</td>
<td>Standard deviation of MA</td>
</tr>
<tr>
<td>MACD</td>
<td>Technical</td>
<td>1 day</td>
<td>$(1 - \frac{n_1}{n_2}){MA(n_1) - MA(n_2 - n_1)}$</td>
</tr>
<tr>
<td>ROC</td>
<td>Technical</td>
<td>1 day</td>
<td>$\frac{P_t - P_{t-T}}{P_L - P_H}, \ T = 10$</td>
</tr>
<tr>
<td>TRIX</td>
<td>Technical</td>
<td>1 day</td>
<td>Triple exponentially smoothed MA</td>
</tr>
<tr>
<td>BV</td>
<td>Fundamental</td>
<td>1 month</td>
<td>Book Value</td>
</tr>
<tr>
<td>CF</td>
<td>Fundamental</td>
<td>1 month</td>
<td>Cash Flow</td>
</tr>
<tr>
<td>Dividends paid</td>
<td>Fundamental</td>
<td>1 month</td>
<td>-</td>
</tr>
<tr>
<td>EBITDA</td>
<td>Fundamental</td>
<td>1 month</td>
<td>Earnings Before Income Tax and Depreciation</td>
</tr>
<tr>
<td>EPS</td>
<td>Fundamental</td>
<td>1 month</td>
<td>Earnings per Share</td>
</tr>
<tr>
<td>Outst</td>
<td>Fundamental</td>
<td>3-6 months</td>
<td>Number of outstanding stocks</td>
</tr>
<tr>
<td>Sales</td>
<td>Fundamental</td>
<td>1 month</td>
<td>-</td>
</tr>
<tr>
<td>ImplVola</td>
<td>Fundamental</td>
<td>1 day</td>
<td>Implied volatility</td>
</tr>
</tbody>
</table>

### Recursive Stock Picking with Decision Trees

*Recursive rules:*

- $X_1 < 0.028$
  - CLASS -1 (42,42)
  - CLASS -1 (13,35)
- $X_2 < 0.057$
  - CLASS -1 (29,7)
  - CLASS 1 (16,6)
  - CLASS 1 (13,1)
Major Steps

Model testing implies the following three stages:

- learning period (November 27, 2000 – June 24, 2001);
- calibration period – $\bar{R}$ (June 24, 2001 – October 22, 2001);
- trading period (October 22, 2001 – October 20, 2003).

Transaction costs are accounted in the amount of 10 b.p. per active transaction.

At the end of every period all active positions are closed.
Equally-Weighted Portfolio

Portfolio returns are calculated basing on individual stock returns. If \( N \) is the number of stocks in the portfolio, then portfolio return \( \Pi \) is defined as follows:

\[
\Pi = \sum_{s=1}^{N} \omega_s \hat{R}_s
\]

where \( \omega_s \) is the weight associated with \( s \)-th stock in the portfolio and \( \hat{R}_s \) is the realized stock yield following the recommended position.

Given that no obviously relevant weighting scheme is available in this context, \( \forall s : \omega_s = \frac{1}{N} \).
Wealth Curves – I: Vola Excluded

Active CART Portfolio (Vola Excluded) Performance Benchmarking

Date
Cumulative Yield
Active CART Portfolio (Vola Excluded) Performance Benchmarking

DAX30
LIBOR 90 days
DJIA
FTSE100
Active CART

Recursive Stock Picking with Decision Trees
Wealth Curves – II: Vola Included

Recursive Stock Picking with Decision Trees
Weekly Returns – I: Vola Excluded

Active CART Strategy Weekly Yields: Vola Excluded

Recursive Stock Picking with Decision Trees
Weekly Returns – II: Vola Included

Active CART Strategy Weekly Yields: Vola Included

Recursive Stock Picking with Decision Trees
Sharpe Ratio Methodology

Annualized mean portfolio return and its standard deviation given the time period of $T$ weeks:

$$\bar{\Pi}_{ann} = \frac{T}{T} \sum_{t=1}^{T} \Pi_i$$

$$\bar{\sigma}(\Pi)_{ann} = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (\Pi_i - \bar{\Pi}_{ann})^2 \sqrt{52}}$$

Given the risk-free rate $R_f$, the **Sharpe ratio** is then defined as follows:

$$SR = \frac{\bar{\Pi}_{ann} - R_f}{\bar{\sigma}(\Pi)_{ann}}$$
## Simulation Results – III: Financial Indicators

<table>
<thead>
<tr>
<th>Vola Excluded</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sharpe ratio</td>
<td>0.88</td>
</tr>
<tr>
<td>Mean relative weekly</td>
<td>19.99%</td>
</tr>
<tr>
<td>Skewness</td>
<td>0.63</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>6.69</td>
</tr>
<tr>
<td>Risk-free rate (Avg. LIBOR)</td>
<td>0.04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vola Included</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sharpe ratio</td>
<td>1.59</td>
</tr>
<tr>
<td>Mean relative weekly</td>
<td>25.55%</td>
</tr>
<tr>
<td>Skewness</td>
<td>1.00</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>5.68</td>
</tr>
<tr>
<td>Risk-free rate (Avg. LIBOR)</td>
<td>0.04</td>
</tr>
</tbody>
</table>
Cross-Validation Live Example: BMW

Recursive Stock Picking with Decision Trees
BNS Live Example: BMW

Recursive Stock Picking with Decision Trees
BNS – Best Node Strategy

Divides nodes into pure/impure i.e. reliable/unreliable groups.

- allows to prune not only the whole branch, but one leaf if necessary;
- does not employ cross-validation;
- needs two extra parameters to be calibrated

\( \bar{p} \) – the minimum allowed probability of the dominating class in a pure node (usually 75% or more)

\( \bar{n} \) – the minimum allowed quantity of observations in a pure node (usually 10% or more of the learning sample size)