Program Traffic Locator - A Knowledge Based Expert System

Sankar R. Nair
New York Stock Exchange
11 Wall Street
New York, NY 10005

Abstract
A Program Basket contains a specified set of stocks in a certain proportion. It is a difficult problem to recognize a program basket when the orders comprising the basket are interspersed with the large number of individual retail orders arriving at the Exchange. A knowledge based expert system was designed to identify and "learn" the basket properties so that it can be used to identify baskets using approximate matching.

Introduction:
On a very active day more than 500,000 orders can arrive at the NYSE's Common Message Switch (CMS). Many of these orders are Program Orders; meaning they are orders belonging to "program baskets". Program Traffic Locator (PTL) is a knowledge based expert system to identify program baskets.

A program basket is a set of orders in different stocks with the proportion of shares in each stock approximately the same for the same type of basket. Usually a basket type models an "Index". An Index is a calculated number for a group of stocks based upon price, weightings and other parameters.

Program baskets are used by Stock Index Fund managers to buy or sell stocks. A transaction in an Index Fund, by definition must contain all the underlying stocks in the necessary proportion. To implement a trading strategy in an Index correctly, the orders in all underlying stocks must be executed simultaneously; making it necessary to send out the orders as close to each other in time as possible.

Program baskets are also used for asset allocation; for example institutional managers may decide to move money in or out of stocks depending on market conditions. This can cause buying or selling of program baskets in a relatively short period of time.

Another use of a program basket is in Index Arbitrage. Index Arbitrage is triggered by the price differential between index trading in the futures market and the underlying equities. This is a "risk free" process as long as the underlying stocks and the index option can be bought/sold before their prices change. Thus speed is a very crucial factor and all basket orders once sent must be executed as early as possible. There are several indices in which Index Arbitrage is done; the most common ones are the S&P500 Index, the Major Market Index, the S&P100 index, and the NYSE index. Frequently baskets also correspond to subsets or supersets of these indices.

Program Baskets can also be used in certain proprietary trading strategies by brokerage firms and large investors where one set of stocks is used to mimic the behavior of another set of more active stocks and some underlying index in them.

The order stream arriving NYSE is a "scrambled" combination of retail and program traffic from different member firms coming through different lines. Also composition of stocks in baskets vary considerably from firm to firm. Frequently firms create their own variations of baskets to suit their trading needs. Thus there are no "exact" baskets, and no exact algorithm can be used to identify the program baskets.

Still a program basket has the following characteristics: A program basket contains 15 or more stocks. This is a somewhat arbitrary criterion used by NYSE.

All orders belonging to the basket must arrive as close to each other as possible in time, to be effective for the trading strategy.

All orders must be on the same "side" (ie: buy,sell,sell-short etc)

All orders usually come from the same communication line of a given firm; there are exceptions to this rule; PTL will "learn" the exceptions with "experience".
All orders usually have the same account type field; account type information denotes whether the order is principal (for the sending firm's account) or agency (for a client) etc. Account type is also supposed to indicate whether an order is Index Arbitrage or Program; but there are errors in this including blank account type fields. Thus PTL does not use this to identify baskets.

Another complication is that baskets consisting of round lots (orders with shares in multiples of 100), partial round lots (more than 100 shares, but not a multiple of 100) and odd lots (shares less than 100). Oddlot orders are processed differently on the NYSE floor and the information on them is stored in a different database.

Another problem occurs when the order stream gets fragmented with large time gaps in between. PTL is capable of identifying such situations.

**How PTL works:**

PTL's knowledge database contains templates of baskets used in the past by various firms. There are two knowledge databases, one temporary and one permanent.

- A newly identified basket's template always goes into the temporary database; if it is identified a second time in the order stream, it moves into the permanent database.

PTL first breaks down the arriving order stream into pseudo-baskets based on a set of rules. Some of the parameters in the rules are derived from experience and hence is part of the Knowledge Data base. Such parameters are typical inter-order times, firmnames with very large inter-order times, firmnames which use multiple lines and multiplex among lines while transmitting the same basket, etc.

Once pseudo-baskets are separated out, an effort is made to correlate them with the existing templates. If there are baskets adjacent in time to a given basket on either side, groups of adjacent baskets are added together in all possible ways and compared to the templates. If any particular combination gives better correlation with any of the templates, then that combination is designated as a basket. All pseudo-baskets and their combinations which "match" with existing templates are identified as baskets.

PTL can be run in two modes - the "regular" mode and the "learning mode". In the learning mode, its knowledge database is constantly modified based on the new basket types encountered for each firm. When PTL is run in the "learning mode", the pseudo-baskets which donot match with existing templates are added to the temporary data base and also identified as pseudo-baskets in the output of the program. In the learning mode, if a pseudo-basket is encountered which matches with a template in the temporary database, that template moves up to the permanent database and is deleted from the temporary database.

A match between a template and a basket is always an approximate one. Two baskets are identical if they contain the same stocks in the same proportion. Thus a template contains the names of the stocks and their relative proportions. The matching constant between two baskets (or one basket and a template) is the ratio of stocks common to both to the total number of stocks multiplied by the square of the correlation coefficient of the vectors of distributions. The threshold for acceptance is currently set as 90%.

The template T consists of two columns; in any row, \( T_{1i} \) contains the stock symbols, and \( T_{2i} \) contains the normalized proportion of shares of \( T_{1i} \) in the underlying basket. \( \Sigma T_{2i} (i=1..n) = 1.0 \)

The signature of the pseudo-baskets in the incoming order stream is a similar matrix \( S \) of two columns with \( m \) rows.

The matching constant is defined as:

\[ r_1 \times r_2, \]

where \( r_1 \) is number of stock symbols common in \( T_{1i} \) and \( S_{1j} \) divided by total number of unique stock symbols in \( U(T_{1i}, S_{1j}) \) and \( r_2 \) is the square of the correlation coefficient of subsets of vectors \( T_{2i} \) and \( S_{2j} \) with \( T_{1i}=S_{1j} \).

When the matching constant exceeds the threshold value, the pseudo-basket is called a basket of the same "type" as the template.

Additional matching is done of the result baskets against typical index basket templates stored in a different database. This helps to identify the underlying derivative product (S&P500 Index, Major Market Index, etc) of each basket. The threshold of acceptance here is somewhat loose and is set at 70%.

The temporary data base is periodically flushed to get rid of templates which have not matched with anything for quite some time (currently set at 3 months). The permanent data base is also periodically flushed to get rid of basket types firms have not used recently (currently set at past 6 months).

**Implementation:**

PTL is coded in IBM's APL2/PC and runs on a IBM PS2 Model 70. The Knowledge Database also resides on the PC. The order data for each day is available on an IBM VM Mainframe system. Each day the order data is consolidated to select only the necessary information fields to run PTL and downloaded to the PC.
The knowledge database is object libraries created and maintained using an auxiliary processor of the APL2/PC, which allows random access, delete and replace.

PTL is implemented using very compact and optimized code. Reading in of the huge input data stream is buffered for optimum performance. Still it takes approximately 6-8 hours of computing time on the PC to run a typical day with the learning mode enabled. The learning mode is enabled only once a week. In a typical run the steps are:

1. Logon to mainframe, download data to PC and logoff.
2. Run PTL on the PC.
3. Logon to mainframe, upload results, reformat and append to an IC/1 query file maintained on the VM.
4. Backup the knowledge data base to the LAN server.

All these functions are done automatically under program control by APL2. At present distributed parallel processing of the data stream by different machines on the LAN under program control is being developed. This is expected to reduce the processing time considerably.

**Results:**

Data about program trading is important to the Exchange for regulatory purposes, capacity planning, and in general to serve its customers better. Therefore member firms are required to report their program trading activities to the Exchange.

Excellent correlation of results have been obtained between results obtained from PTL and data actually reported by firms sending in Program Orders. On many occasions PTL has identified new trends in program trading which has helped the Exchange in planning message capacities of floor systems. PTL has also been helpful in pointing out misreporting of program orders in some instances.

The table below illustrates the process of "learning" by PTL about a certain program firm which mixes retail orders with program orders. In this sequence PTL learns about 372 and 20 stock baskets sent by this firm, and keeps in its temporary database the template of a 453 stock basket. Comparison of member submissions to PTL on a very active day for a selected firm is also shown below.

<table>
<thead>
<tr>
<th>Date</th>
<th>&quot;Actual&quot; Baskets</th>
<th>Baskets by PTL</th>
<th>Temporary Database</th>
<th>Permanent Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning begins</td>
<td>5/2</td>
<td>453x1</td>
<td>-</td>
<td>453</td>
</tr>
<tr>
<td>Nothing on this day</td>
<td>5/3</td>
<td>none</td>
<td>-</td>
<td>453</td>
</tr>
<tr>
<td>Detects 370-stock baskets</td>
<td>5/4</td>
<td>20x1,370x5</td>
<td>370x3</td>
<td>453,20</td>
</tr>
<tr>
<td>Detects 372-stock baskets</td>
<td>5/5</td>
<td>372x2</td>
<td>372x2</td>
<td>453,20</td>
</tr>
<tr>
<td>Detects 20-stock baskets</td>
<td>5/6</td>
<td>20x2,372x2</td>
<td>20x2,372x2</td>
<td>453</td>
</tr>
<tr>
<td>Test Ends</td>
<td>5/9</td>
<td>20x6,372x1</td>
<td>20x6,372x1</td>
<td>453</td>
</tr>
</tbody>
</table>

**Learning Process**

Notes:
1. "Actual" baskets located with manual assistance
2. 453-stock basket awaits confirmation, having occurred only once
**Basket Arrival Times**

*Member Submissions versus PTL for a Firm*

- **O** - PTL
- **X** - Member Submissions

**Acknowledgements:**

Author acknowledges input from Raman Srivastav and Alan Rudnick in the development of PTL, Joel Beier in the preparation of this paper and the work by David Greenzweig of Securities Industries Automation Corporation (SIAC) in the development of PTL's input database.