Informeta, L.L.C.

Trading United States Treasury Yields

Simulation Design and Follow-up Results

Matthew Johnson
matthew.johnson@informeta.net

Ron Coleman, Ph.D.
ron.coleman@informeta.net

1 Civic Center Plaza
Poughkeepsie, New York 12601
www.informeta.net

1. Executive Summary

This document details the final results from our exploration of the use of the predictive features of *Mentys*™ to trade daily U.S. Treasury constant maturity rates (TCMR).[1] The figures given below were obtained using first differences of the original data provided by a premier Wall Street analytics Company, and without augmenting with textual news from *The New York Times*.

We executed controlled tests using *Mentys*™ to simulate an artificially intelligent process that learns simple trading parameters to maximize mean returns on the 3M TCMR.[2] We also tested a random walk strategy (base line) as well as two different feed-forward (FF) artificial neural network (ANN) architectures.

The Table 1 below summarizes the out-of-sample potential spanning 50 months of historical 3M yields. The data assume application of $100K per trade with one basis point transaction costs.

Table 1. Validation performance on 3Mo TCMR

Method	**Buy-Hold**	**Base Line**	**FF1 ANN**	**FF2 ANN**	**Mentys™ (10% accept ratio)**	**Perfect**
Mean ret (%)	1.01	0.32	-1.27	-0.78	1.96	2.12
Ret/Risk	2.26	0.12	-0.19	-0.11	0.29	0.72
N Trades	17	281	32	26	160	280
Avg Invst ($K)	65	65	38	31	292	65
Max Invst ($K)	100	100	100	100	500	100
Drawdwn ($K)	0	10	15	15	47	0
P&L ($K)	17	83	-41	-21	310	588

Table 2. Validation performance on 3Mo TCMR (β=0*)*

Method	**Buy-Hold**	**Base Line**	**FF1 NN**	**FF2 NN**	**Mentys™ (10% accept ratio)**	**Perfect**
Mean ret (%)	1.01	0.32	-1.27	-0.78	2.22	2.12
Ret/Risk	2.26	0.12	-0.19	-0.11	0.29	0.72
N Trades	17	281	32	26	151	280
Avg Invst ($K)	65	65	38	31	294	65
Max Invst ($K)	100	100	100	100	500	100
Drawdwn ($K)	0	10	15	15	47	0
P&L ($K)	17	83	-41	-21	332	588

---

[1] "Trading United States Treasury Yields", Informeta, L.L.C., Confidential, v1.3, xx-xx-xxxx.
[2] "Forecasting United States Treasury Yields", Informeta, L.L.C., Confidential, v1.3, xx-xx-xxxx.

The remainder of this document details the design of the artificially intelligent process, built around the predictive features of *Mentys*™, used to simulate the daily trading of U.S. Treasury constant maturity rates.

2. Data

The data analyzed in this study comprised approximately nine (9) years of historical TCMR for ten contract terms up to and including the thirty-year (30Y). Approximately half of the records were used to construct the knowledge bases, while the remainder was used for forecasting.

Although no supplementary data were used to augment the TCMR, we did preprocess the data in two significant ways. In an effort to bootstrap the system, we employed the technique of autocorrelation. Furthermore, since all the data were time-series, we also applied a first differences transformation.

3. Trading Simulation Design

The trading simulation is a three-stage process that progresses from the construction of knowledge bases through simulated trading based on forecasts generated by those knowledge bases. The executables used were built from `simpute4.cpp` revision 22 and `tsdetail.cpp` revision 17. A summary analysis of the results obtained is given in Tables 1 and 2 above.

3.1. Assumptions

Several assumptions are made to this end. All trades are made "at closing." We assume high liquidity, so any desired buy/sell transaction is possible. Transaction costs are computed at a fixed rate of 0.01 %. Our initial account balance is $1M and capital is ready-available for every trade. The base investment per trade is $100K, and no portfolio management is used to adjust this amount up or down.

3.2. Stage One

The first stage of the process begins with the construction of the knowledge bases. For this study, *Mentys*™ learns a portfolio of five (5) knowledge bases that differed in size only, ranging from ranging from 800 to 1000 records and covering roughly 3 to 4 years of historical data. All of the knowledge bases use a granularity factor of zero (0).

Using each knowledge base, *Mentys*™ generates a sequence of forecasts over a predetermined time interval (namely, 1250 trading days). By employing a portfolio of distinct knowledge bases, we attempt to hedge our bets and curb over-fitting. We then use the resulting forecast sequences to trade along multiple pathways simultaneously. [3]

3.3. Stage Two

In stage two, we determine the reliability thresholds associated with each knowledge base. Each forecast sequence is partitioned into two segments: the calibration segment (in our study this included 250 records), and the validation segment (consisting of 1000 records).

---

[3] The binary executable that runs this process is named **`simpute4.exe`**.

The reliability thresholds, which we obtain from the calibration segment of the test data, are the percentiles of the observed posterior probabilities corresponding to six acceptance ratios: 1%, 5%, 10%, 15%, 20%, and 25%. For each acceptance ratio, we derive an acceptance threshold, α, and a contrarian threshold, β. (For more detail, see section **Trading Parameters** below.) At the 5% acceptance ratio, 5% of the probabilities observed in calibration exceed α while 5% fall below β. In validation, we expect the proportions to differ slightly.

3.4. Stage Three

In the final stage, we input to the trading machine the portfolio of forecast sequences and the trading parameters determined during calibration. The trading machine is reinitialized with each new set of parameters. For each parameter configuration, the forecast sequences are run consecutively through the machine, producing a series of simulated trades. These trades are analyzed separately for each set of parameters under which they were produced, and a final summary is generated.[4]

3.5. Trading Machine

The trading simulator is built around a simple automaton. This trading machine is designed to make simulated trades based on a sequence of forecasts for a single instrument. Conceptually, transition between two states in the machine depends upon a trade signal, which takes one of the values *buy*, *sell*, *hold*. A graphical depiction of this automaton is given below.

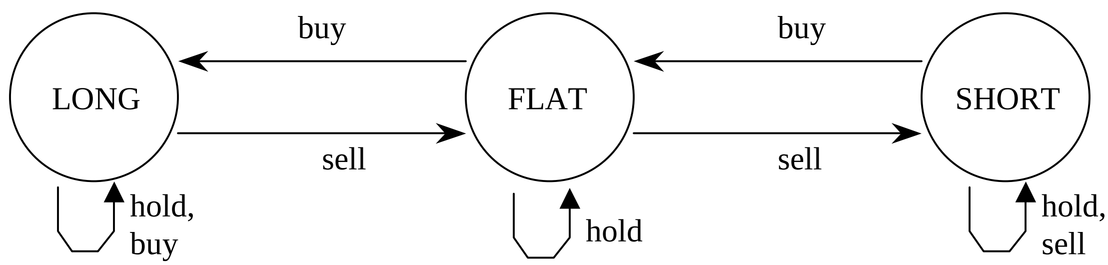

Figure 1. Automaton, from the trading simulator

In practice, input to the trading machine is not a simple trade signal, but rather a forecast for the next price of the instrument. The translation of forecast to trade signal is governed by: (1) a set of trading parameters set during initialization, and (2) the particulars of the forecast, including current instrument price, predicted next price, and forecast reliability.

The trading machine also performs simple accounting to manage the transaction details. During each iteration, the return, drawdown, investment amount, and account balance are updated.

3.6. Trading Parameters

The trading parameters that guide the transformation of forecast to trade signal are listed below:

---

[4] The binary executable that runs this process is named **`tsdetail.exe`**.

- *PM strategy* – If we already have long/short position, this indicator is used to decide whether to hold, pile-in, or pile-out on confirmation.
- *Pile fraction* – The proportion of the base trade amount to pile-in/-out, as applicable.
- *Max hold time* – The maximum number of iterations to maintain a long/short position.
- *Min hold time* – The minimum number of iterations to maintain a long/short position.
- *Loss tolerance* – A pair of values that indicate how long to maintain a long/short position in the face of negative return, as well as how large a loss we are willing to incur.
- *Margin of error* – If the magnitude of the predicted change in price does not exceed this threshold, we treat the forecast as "no change."
- *Reliability acceptance threshold,* α – The probability above which a forecast is accepted as accurate, and hence believed to be correct.
- *Reliability contrarian threshold,* β – The probability below which a forecast is rejected, and believed to be contrary to the correct value, hence we accept the inverse forecast.
- *Flat-contrarian flag,* **r** – Indicates whether to act on contrarian signals when flat.

Since we use a portfolio of knowledge bases, one set of trading parameters contains several (α,β) pairs, one for each knowledge base.