InferTrade
PREDICTIVE RESEARCH TOOLS

# Predictive Relationship: Level Relationship

## Contents

# 1 Relationship Description

**In a level relationship, the current value of the signal directly affects future changes in price:**

$$\Delta p_{t+1} \propto kS_t + c \tag{1}$$

where $\Delta p_{t+1}$ represents the future price change, $S_t$ represents the current value of the research signal at current time $t$, $k$ is a scalar, and $c$ is a constant.

Intuitively, assuming a positive coefficient for k, this means that when the signal is high it is good to buy the market, as the price will tend to drift higher. When the signal is low, it is good to sell the market as the price will tend to drift lower.

An example of a time series that might show a predictive level relationship would be a sentiment index:

*positive sentiment –> index value is high –> price goes up because investors are bullish*

We would typically not expect a signal that is a price forecast to show a predictive level relationship, as a price forecast is typically relative to the current price (a difference relationship). Similarly, a technical positioning indicator as a signal may be more likely to show a change relationship, whereby changes in the value of a signal drive the price, because a stable level of interest may not move the market.

# 2 Trading Strategy Description

A predictive level relationship could be reflected in many kinds of rules. InferTrade uses a 120 period (6 months for daily data) rolling regression of the signal $r_t$ against next day's observed percentage price change $\Delta p_{t+1}$ as a benchmark to detect level relationships. This section describes in detail this simple strategy allocation recipe that InferTrade used to detect this relationship, but any kind of rule that takes bigger positions when the signal is higher (assuming k is positive) would generate a predictive edge. We provide the below explanation so that you can replicate the results you have obtained on an independent platform and/or modify further to match your trading and investment needs.

The rolling regression level relationship trading rule recommends taking a position sized as a percentage of the capital you have allocated to trading this particular market and will show higher returns than usual if a significant level relationship is present between the price and signal series.

To calculate how much to invest, InferTrade first performs a standard rolling linear regression of the observed daily price changes $\Delta p_t = \frac{p_t}{p_{t-1}} - 1$ against the previous day's research levels $r_{t-1}$ for a rolling window of length $L$. This creates an equation for estimating the future percentage price change to occur from today to tomorrow $\Delta p_{t+1}$ using today's research level $r_t$:

$$\Delta p_{t+1} = k_1 r_t + k_2 + \epsilon_{t+1} \tag{2}$$

with the error symmetric such that

$$E_t^{model}[\Delta p_{t+1}] = k_1 r_t + k_2 \tag{3}$$

where $E_t^{model}[\Delta p_{t+1}]$ is the predicted next day percentage price change, $r_t$ is the research value at time $t$, and $k_1$ and $k_2$ are the level and static coefficients. $\epsilon_t$ represents the in sample regression error between the model predicted and observed price change for time t, such that $\epsilon_t = E_{t-1}^{model}[\Delta p_t] - \Delta p_t$.

To give a suggested portfolio allocation $z_t$ for the market, e.g. "invest 5% of strategy allocation into the security", the regression forecast's expected next day % price move is divided by the square of the Standard Root Mean Squared Percentage Prediction Error $\varepsilon_{rms_t}^2$ in our above regression:

$$z_t = \begin{cases} \frac{\Delta p_{t+1}}{\varepsilon_{rms_t}^2} & \text{if } t > L, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $z_t$ is the suggested allocation as a percentage of the capital which you have allocated to trading this market at time $t$.

Worked example for liquid market: current share price is \$50, regression coefficients are 0.00005 ($5 * 10^-5$) for level and 0.0 for the constant, with \$1,000 allocated to this single-stock strategy. The current research series - a sentiment index - is +5.0, with a RMS prediction error from the regression of 5%. In this situation the strategy recipe recommends an investment of $0.00005 * 5.0/(0.05)^2 = 10\%$ be allocated. So \$1,000 * 10% = \$100 would be allocated, so we would buy $\frac{\$100}{\$100} = 2$ shares. We see from this example the expected gain over the next day is 2.5 basis points (minus trading costs), versus 500 basis points of daily risk, so in practice this kind of strategy is most useful where the signal both has a predictive edge and is stable (auto-correlated) in recommendation over time, so that the 'edge' from multiple time steps can compound linearly without incurring additional bid-offer spreads whilst the 'risk' aggregates scales with the square-root of time (for random walk).

(Note that if the magnitude of the error term is smaller than the magnitude of the price change prediction, i.e. $|\varepsilon_{rms_t}| < |\Delta p_{t+1}|$, dividing $\Delta p_{t+1}$ by $\varepsilon^2_{rms_t}$ would result in $z_t > 1$. Limits can then be applied to cap the allocation. E.g. between 0% and 100% if short selling is prohibited. In practice it is generally unlikely that in liquid markets a genuine statistical trading signal this strong will be found.)

# 3 Fixed Strategy Parameters

Below is a table summarizing the parameters specific to this trading rule.

| Parameter Name | Default Value | Description | Symbol |
|---|---|---|---|
| Regression Length | 120 | This is the number of previous days used to estimate the regression coefficients. | $L$ |

For estimating the regression coefficients, a rolling window is used, so that at time t, the regression data uses research data from time $t - 121$ to $t - 1$ to calculate 120 research changes as the regression's independent variable. Price data from $t - 120$ to $t$ is used to calculate 120 next day price changes as the regression's dependent variable.

Note that rule parameters are fixed to their default values for this and any other relationships optimised on simplified mode.

# 4 Glossary

- **Bullish:** Positive outlook on the market. Expectation of positive returns.
- **Bearish:** Negative outlook on the market. Expectation of negative returns.
- **Allocation:** The allocation is the fractional amount of the portfolios value used to determine the size of the trading position.
- **Parameter:** Value used by the trading rule in the calculation for trading position
- **Trading Rule:** Strategy to determine when to buy, hold or sell a position.

# Further Links

1. InferTrade: `https://www.infertrade.com`
2. Privacy Policy/Legal notice: `https://www.infertrade.com/privacy-policy`
3. InferStat Ltd: `https://www.inferstat.com`

# Appendix

## Calculating Regression Error

This PDF omits any explanation of how the regression coefficients $k_1$ and $k_2$ are calculated as we are using a standard regression for calculation of both the coefficients and standard error. For InferTrade we used the open source *sklearn* Python library. The same can be achieved in Microsoft Excel by using the *FORECAST.LINEAR* function.

As a worked illustration of the error calculation, assume that InferTrade predicted that a security would increase in price by 0.1% between yesterday and today, and the price actually decreased by 4.9%. In this example, $\Delta p_t = +0.1\%$, and $\Delta o_t = -4.9\%$, giving $\Delta P_t - \Delta p_t = 0.1\% - 4.9\% = -5\% = -0.05$. This value is then squared to give $(\Delta p_t - \Delta o_t)^2 = (-0.05)^2 = 0.0025$. To obtain $\varepsilon^2_{rms_t}$ for day $t$, this calculation is repeated for every day from day $i = t - L$ to day $i = t$, with values summed before dividing by the regression length $L$.