

We create a trading agent endowed with an information set based on past prices of the S&P500 and with a set of tools to compute marginal CDFs and copula-based joint distributions and conditional probabilities. Formally, let $\mathcal{S} := \{i : i \text{ quotes in the S\&P500}\}$ denote the set of stocks in the S&P500, and let $\mathcal{P}_t := \sigma(\{p_{s,i}\}_{s \leq t, i \in \mathcal{S}})$ denote the information set of the agent, based on the sigma algebra of the past and contemporaneous prices of the S&P500 stocks.

The agent aims to perform pairs trading on the S&P500. In traditional pairs trading, a trading agent would focus on finding a correlated pair (with the hope that such correlation profile would hold still over time) and trade the spread hoping for mean reversion. In this paper, we will endow the pairs trading agent with a new technology. Our agent will pairs trade a selected stock (target asset) vs. its replica (synthetic control). In particular, it will choose a “*target*” stock (with price denoted p_t^{trgt}) and mimick it with a “*synthetic*” stock, built as a sparse linear combination of other stocks from the S&P500 universe $p_t^{\text{synth}} = \sum_{i \in \mathcal{S}} w_{t,i} p_{t,i}$.

The technogology to compute the synthetic control is based on an ℓ_1 -regularized least squares program where we minimize the mean squared error of the target stock price series with respect to a linear combination of other stocks in the S&P500 on a lookback window of 252 days and imposing a constraint on the weights so that they add up to one. To avoid inflating transactions costs with excessive rebalancing, we don’t recalculate the weights after each price realization, but instead, recompute the weights every 100 days. This delivers the price series $\{p_t^{\text{trgt}}\}_t, \{p_t^{\text{synth}}\}_t$.

The agent then computes the returns of the the target and synthetic price series $\{r_t^{\text{trgt}}\}_t, \{r_t^{\text{synth}}\}_t$ and use the latter to compute a joint distribution of returns, $F_t(R^{\text{trgt}}, R^{\text{synth}})$, by means of a parametric copula. The t -subscript in the joint distribution denotes the fact that, to obtain the most accurate representation of the joint distribution, we recalibrate the copula after each return realization.

The agent then uses the joint distribution of returns to compute a mispricing index between the pair. Such mispricing index MI_t is obtained as a conditional probability; in particular, as the probability that the target (synthetic) return is lower than its realized value conditional on the synthetic (target) return being equal to its current value. Formally:

$$\begin{aligned} MI_t^{\text{trgt}|\text{synth}} &= \mathbb{P} \left[R^{\text{trgt}} \leq r_t^{\text{trgt}} \mid R^{\text{synth}} = r_t^{\text{synth}} \right] \\ MI_t^{\text{synth}|\text{trgt}} &= \mathbb{P} \left[R^{\text{synth}} \leq r_t^{\text{synth}} \mid R^{\text{trgt}} = r_t^{\text{trgt}} \right] \end{aligned}$$

Subsequently, the agent computes a cumulative mispricing index (CMI) by accumulating these mispricing indices in an autoregressive way. The accumulation of the mispricing indices (conditional probabilities) is done in excess of 0.5 (i.e. the noninformative probability level, or simply,

the probability of a fair coin toss).

$$\begin{aligned} CMI_0^{\text{trgt|synth}} &= 0; & CMI_t^{\text{trgt|synth}} &= CMI_{t-1}^{\text{trgt|synth}} + (MI_t^{\text{trgt|synth}} - 0.5); \\ CMI_0^{\text{synth|trgt}} &= 0; & CMI_t^{\text{synth|trgt}} &= CMI_{t-1}^{\text{synth|trgt}} + (MI_t^{\text{synth|trgt}} - 0.5); \end{aligned}$$

In the last step the trading agent launches simultaneous trades on the target-synthetic pair to trade away the accumulated mispricing when it exceeds certain thresholds. In particular, the trading rule maps the CMIs to trading positions (short, neutral, long) by means of a parametric trading rule with opening thresholds (D_u, D_l) and stop loss thresholds (S_u, S_l) . Formally, stack the thresholds in $\boldsymbol{\theta} = (D_u, D_l, S_u, S_l) \subseteq \Theta$, then the trading rule $f : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times \Theta \rightarrow \{-1, 0, 1\}$ performs the contemporaneous mapping $(CMI_t^{\text{trgt|synth}}, CMI_t^{\text{synth|trgt}}, \boldsymbol{\theta}) \mapsto TR_t$.

Alternatively, we can determine the trading rule optimally through reinforcement learning. In this case, we would interpret $s_t = (CMI_t^{\text{trgt|synth}}, CMI_t^{\text{synth|trgt}}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$ as the state, and $a_t = TR_t \in \{-1, 0, 1\}$ as the action. Then the agent dynamically learns the policy function f_t by maximizing a reward function, for which we will use mean variance preferences: $\pi_t = \prod_{s=0}^t \gamma^s (1 + r_s) / \sigma$ (revise this utility function).