Stochastic Modelling and Optimisation
Pairs Trading: A Kalman Filtering Approach

T. Dunlop, J. Llorens, M. Keys, R. Knudsen

March 4, 2018

Abstract

Pairs Trading is a statistical arbitrage strategy that takes advantage of financial markets that are out of equilibrium. In this report, we implement Elliott et al. (2005)'s mean-reverting Gaussian Hidden Markov Chain Model for Pairs Trading, propose different trading rules and illustrate the profitability of our approach on simulated and real-world data. Utilising a EM Smoothing approach for parameter estimation and Kalman Filtering for spread prediction, we show that our strategies are profitable when not accounting for transaction cost. We conclude by suggesting directions for further research to improve the trading rules.

1 Introduction

In financial markets, there is a natural interest in trading strategies that yield above-average returns. One class of strategies that goes by the name "statistical arbitrage" seeks to exploit financial markets that are out of equilibrium and capitalise on the reversion. A popular strategy within this class is Pair Trading. The basic idea is simple. One chooses two securities that are historically "close" (e.g. stocks of Pepsi and Coca Cola) and assumes that their distance (e.g. price spread, return spread) will follow a specific process. Whenever the estimation of this process indicates that the distance is out of equilibrium, one buys one security and short-sells the other to benefit from the reversion to equilibrium.

Various research over the last 20 years has developed methodology for Pairs Trading and documented its profitability on historical data sets. Gatev et al. (1999, 2006) show that a simple strategy entering trades when "prices diverge by more than two historical standard deviations" (p.804) can yield substantial long-term above-average returns, even after accounting for transaction costs. Elliott et al. (2005) propose a more sophisticated mean-reverting hidden Gaussian-Markov chain model for the spread that can be estimated using an offline-smoothing algorithm following Shumway and Stoffer (1982) or using dynamic filtering following (Elliott and Krishnamurthy, 1997). Recent research indicates, however, that the profitability of simple model setups has fallen since the early 2000s Do and Faff (2010, 2012).

This motivates our research to explore and implement (Elliott et al., 2005) offline-smoothing algorithm, propose different trading rules that seek to maximise the profit and illustrate the profitability on simulated and real-world data. Much work has been done on algorithms that identify optimal pairs (see e.g. Gatev et al., 2006). Hence, we focus on the mathematical model of the spread (Elliott et al., 2005) and trading rules. Finally, we provide a Dynamic Programming formulation of choosing the trading rule based on (Elliott et al., 2005)'s model that can be used as a starting point for future research.

Hence, the remainder of this report is structured as follows. Section two provides an in-depth analysis of the model and trading rules. Section three reports the profitability of the proposed trading rules when applied to historical data. Section four elaborates on optimality of the trading rules and proposes a Dynamic Programing formulation. Section five concludes.

2 Model and estimation

2.1 Pairs Trading

As explained above, Pairs Trading is based on the belief that markets may not always be in equilibrium but that they move to a rational equilibrium over time. Hence, if we were able to know at a certain point in time whether the market is in equilibrium or not, we could take advantage from deviations from that equilibrium level. In our model, the main idea is to take a long position in a security and a short position in another security which is co-integrated with the other in the sense of Engle and Granger (1987). In other words, we are betting on the fact that the spread of both securities will eventually revert to its mean value.

For instance, if we let S_k^{high} denote the price of the 'high' security and S_k^{low} denote the price of the 'low' security, we can observe the spread $y_k = S_k^{high} - S_k^{low}$ between both securities. Intuitively, if we think y_k is 'too high', we would take a long position on the 'low' security and a short position on the 'high' security, since we believe that y_k will decrease over time and hence we would make a profit when unwinding the trade. Otherwise, we could make the reverse trade. But what does 'too high' actually mean?

In our model, we will assume that there is a 'true mean spread' x_k, which we will define as a latent state. We may allow the dynamics of x_k to depend on the previous state x_{k-1} and some random noise. The observed spread y_k may depend on x_k plus some measurement error. The goal is to infer the true state from the security price information that we get in every period. Here, we will assume that the dynamics of x_k are linear and that both sources of uncertainty follow a Gaussian distribution. In other words, we will use the Kalman filter approach to estimate x_k, which will form the basis of our trading strategy.

State process (model for spread):

$$x_k = A + Bx_{k-1} + C\epsilon_k, \text{ where } \epsilon_k \sim \mathcal{N}(0,1), \text{iid} \tag{1}$$

For our trading strategy to make sense, we need to assume that our model for x_k follows a mean-reverting process. For example, if τ is the step length, and $A = a\tau, B = 1 - b\tau, C = \sigma\sqrt{\tau} \geq 0$, then $X(t)$ (the limit process of x_k) follows an Ornstein-Uhlenbeck process:

$$dX(t) = (a - bX(t))dt + \sigma dW(t) \tag{2}$$

where $\{W(t)|t \geq 0\}$ is a standard Brownian motion.

Observation process:

$$y_k = x_k + D\omega_k, \text{ where } \omega_k \sim \mathcal{N}(0,1) \tag{3}$$

In order to make a trading decision, we will use the best estimate of the spread given all the past information up to time $k-1$. This is defined as the conditional expectation of x_k given the series of observations up to period $k-1$, which we denote by $\mathcal{Y}_{k-1}$:

$$\hat{x}_{k|k-1} = \mathbb{E}[x_k|\mathcal{Y}_{k-1}] = A + B\hat{x}_{k-1} \tag{4}$$

where $\hat{x}_k = \mathbb{E}[x_k|\mathcal{Y}_k]$.

If our best estimate of the true spread $\hat{x}_{k|k-1}$ is lower than the observed spread y_k, then we believe the observed spread is 'too high'. Therefore, we should enter a short position on the 'high' security and a long position on the 'low' security, thus making a profit when the spread reverts to the true one.

A basic decision rule u_k could be written as follows:

$$u_k = \begin{cases} \text{short } r_k \text{ units of } \textit{high} \text{ and long 1 unit of } \textit{low} \text{ (long spread trade)} & \text{if } y_k - \hat{x}_{k|k-1} > 0 \\ \text{short } r_k \text{ units of } \textit{low} \text{ and long 1 unit of } \textit{high} \text{ (short spread trade)} & \text{if } y_k - \hat{x}_{k|k-1} \leq 0 \end{cases}$$

where $r_k = \frac{S_k^{low}}{S_k^{high}}$ because we want to build a self-financing portfolio. In other words, there is no exogenous infusion (or withdrawal) of money as we will short sell the high stock in the same dollar amount used to take a long position in the low stock.

However, one may argue that when the difference is too close to zero, we may not be sure enough to go long or short the spread trade. Instead, we may choose to trade only when we are confident enough that the difference between y_k and $\hat{x}_{k|k-1}$ is high enough. Since we are using the Kalman filter approach, we can recover both the best estimate of the hidden spread and the estimated uncertainty around that estimate. Therefore, we can argue that a 'better' strategy may be to trade only when the absolute standard residual is greater than a certain threshold, which we denote by α.

This trading rule could be formulated as

$$u_k = \begin{cases} \text{short } r_k \text{ units of } \textit{high} \text{ and long 1 unit of } \textit{low} \text{ (long spread trade)} & \text{if } \frac{y_k - \hat{x}_{k|k-1}}{\hat{R}_{k|k-1}} \geq \alpha \\ \text{short } r_k \text{ units of } \textit{low} \text{ and long 1 unit of } \textit{high} \text{ (short spread trade)} & \text{if } \frac{y_k - \hat{x}_{k|k-1}}{\hat{R}_{k|k-1}} \leq -\alpha \\ \text{no trade} & \text{otherwise} \end{cases}$$

where $\hat{R}_{k|k-1} = \mathbb{E}[(x_k - \hat{x}_{k|k-1})^2|\mathcal{Y}_k]$.

Finding the optimal values for α is not a straightforward exercise. Vidyamurthy (2004) provides some ideas for setting the optimal threshold. Additionally, for every trading position we decide to enter in, we need to decide when to exit. We have different possibilities: the next trading time, after T days, or when the spread corrects sufficiently (see Reverre, 2001).

Finally, we need to estimate A, B, C, D from the data. This process is described in section 2.3.

2.2 Estimating Hidden States - Kalman Filtering

In estimating a hidden state there are three approaches one can take, namely forecasting, filtering, and smoothing. The differences between these depend on what data we are conditioning on. Let k be the period of the state we are estimating and N the number of observations we are conditioning on, then if $k = N$ we are filtering, if $k < N$ we are smoothing, and if $k > N$ forecasting. In this section we discuss the former.

The Kalman Filter is a procedure for estimating the underlying state of a hidden markov model in a sequential manner. Specifically, it attempts to estimate the unknown states in the linear model defined by equations (1) and (3) which describe the evolution of a dynamical system. The Kalman Equations are given below:

$$\hat{x}_{k+1|k} = A + B\hat{x}_{k|k} \tag{5}$$
$$\Sigma_{k+1|k} = B^2\Sigma_{k|k} + C^2 \tag{6}$$
$$\mathcal{K}_{k+1} = \Sigma_{k+1|k}/(\Sigma_{k+1|k} + D^2) \tag{7}$$
$$\hat{x}_{k+1} = \hat{x}_{k+1|k} + \mathcal{K}_{k+1}(y_{k+1} - \hat{x}_{k+1|k}) \tag{8}$$
$$\Sigma_{k+1} = \Sigma_{k+1|k} - \mathcal{K}_{k+1}\Sigma_{k+1|k} \tag{9}$$

Where $\hat{x}_k = \mathbb{E}[x_k|\mathcal{Y}_k]$, $\hat{x}_{k+1|k} = \mathbb{E}[x_{k+1}|\mathcal{Y}_k]$, $\Sigma_k = \mathbb{V}(x_k|\mathcal{Y}_k)$, and $\Sigma_{k+1|k} = \mathbb{V}(x_{k+1}|\mathcal{Y}_k)$.

The Kalman Filtering algorithm is a two stage process. We first obtain the apriori state and covariance estimates based on information from the previous state by equations (5) and (6). Then we compute the optimal Kalman Gain defined by (7) which is a relative measure of certainty in our observed data and uncertainty in our prediction. Our aposteriori estimates of the state and variance are then given by a weighted sum of the a priori estimate and the observed value. The larger the Kalman Gain, the more weight is put on the current observed data, whilst the smaller the gain, the more weight put on our prediction. It can be shown that this weighted sum of the apriori prediction and the observation has a smaller uncertainty than basing state estimates on just one of the two. Also note that at each recursion, we only depend on estimates calculated in the previous iteration and the new observed data y_{k+1}, which means we do not have to parse through all the data $y_1, ..., y_k$ when recieving a new observation y_{k+1}, which is a significant computational advantage.
For initialization, Elliott et al. (2005) recommends $\hat{x}_0 = y_0$ and $\Sigma_0 = D^2$. Giving as input to the kalman filter the parameters of the model and initial values, we are able to sequentially estimate the hidden state for each observed value that we have. If we wish to forecast hidden states and their uncertaintities in future time periods for which we have no observable data, then we simply recurse forward using (5) and (6), initializing with the final output of the Kalman Filter, $\hat{x}_{N|N}$ and $\Sigma_{N|N}$ to obtain $\hat{x}_{N+h|N}$ and $\Sigma_{N+h|N}$.
There are several theoretical results motivating the use of the Kalman Filter in applications. One can show that if our series do follow exactly the dynamics described by (1) and (3), our estimates of D and C are exact, and ω_k and ϵ_k are iid, then our aposteriori state estimate is the minimum mean squared estimator (Anderson and Moore, 1979). In reality, our systems do not follow the modelled dynamics exactly, and obtaining estimates of our noise variances is not trivial given the embedded latent structure that prevents us from obtaining parameter estimates with ideal properties. This raises the question of how well this filtering process works in real applications with relatively simple models. It is well known that poor estimates of these parameters can evoke filter divergence and thereby seriously impair performance (Jwo and Cho, 2007) so one has to be particularly careful here.

2.3 Parameter Estimation - EM Algorithm

In this section we present the procedure for estimating the parameters of the state space model presented in this paper, which in turn allows us to perform filtering and forecasting on new observations. We implement an offline Expectation-Maximisation (EM) algorithm which utilises Kalman Smoothing in order to impute the data of the complete log likelihood which depends on an unobseravble variable x. Following this, we can derive closed form updates for our parameters. This approach to EM was first introduced by Shumway and Stoffer (1982).

Ideally, we are interested in estimators of the form $\theta = \arg\max \mathcal{L}(y|\theta)$, where $\theta = (A, B, C^2, D^2)$. That is, estimators that maximise the likelihood of the observed data. However, given the latent structure imbedded in equations (1) and (3) and their corresponding distributions, closed form MLEs to the parameters θ are not available. An iterative approach here is feasible however. Let the complete data likelihood of our model defined by equations (1) and (3) be denoted by $\mathcal{L}(x, y|\theta)$. By the Markov property, this joint likelihood can be factorised as

$$\mathcal{L}(x, y|\theta) = p(x_0) \prod_{k=1}^{N} p(x_k|x_{k-1}) \prod_{k=0}^{N} p(y_k|x_k) \tag{10}$$

We know that $y_k|x_k \sim \mathcal{N}(x_k, D^2)$ and $x_k|x_{k-1} \sim \mathcal{N}(A + Bx_{k-1}, C^2)$. Assume that $x_0 \sim \mathcal{N}(m, p^2)$, then the complete data log likelihood can be written as

$$log\mathcal{L}(x, y|\theta) = \tag{11}$$

$$-\left(log(p) + Nlog(C) + Nlog(D) + \frac{(x_0 - m)^2}{2p^2} + \sum_{k=1}^{N} \frac{(x_k - (A + Bx_{k-1}))^2}{2C^2} + \sum_{0=1}^{N} \frac{(y_k - x_k)^2}{2D^2} \right) \tag{12}$$

Define the auxillary function $Q(\theta, \theta') \triangleq \mathbb{E}[log\mathcal{L}(x, y|\theta)|y_1, ..y_N, \theta']$, then simple algebra yields

$$Q(\theta, \theta') =$$

$$-\left(\underbrace{log(p) + Nlog(C) + Nlog(D) + \frac{1}{2p}}_{=\eta} + \mathbb{E}\left[\sum_{k=1}^{N} \frac{(x_k - (A + Bx_{k-1}))^2}{2C^2} + \sum_{i=1}^{N} \frac{(y_k - x_k)^2}{2D^2} |\mathcal{Y}_N \right] \right) + c \tag{13}$$

$$= -\left(\mathbb{E}\left[\sum_{k=1}^{N} \frac{x_k^2 - 2x_k(A + Bx_{k-1}) + (A + Bx_{k-1})^2}{2C^2} + \sum_{i=1}^{N} \frac{y_k^2 - 2x_k y_k + x_k^2}{2D^2} |\mathcal{Y}_N \right] \right) - \eta + c \tag{14}$$

$$= -\Bigg(\sum_{k=1}^{N} \frac{\mathbb{E}[x_k^2|\mathcal{Y}_N] - 2A\mathbb{E}[x_k|\mathcal{Y}_N] - 2B\mathbb{E}[x_k x_{k-1}|\mathcal{Y}_N] + 2AB\mathbb{E}[x_{k-1}|\mathcal{Y}_N] + B^2\mathbb{E}[x_{k-1}^2|\mathcal{Y}_N] + A^2}{2C^2}$$
$$+ \sum_{k=1}^{N} \frac{y_k^2 - 2y_k\mathbb{E}[x_k|\mathcal{Y}_N] + \mathbb{E}[x_k^2|\mathcal{Y}_N]}{2D^2} \Bigg) - \eta + c \tag{15}$$

$$= -\Bigg(\sum_{k=1}^{N}\Bigg[\frac{\Sigma_{k|N} + \hat{x}_{k|N}^2 - 2A\hat{x}_{k|N} - 2B(\Sigma_{k,k-1|N} + \hat{x}_{k|N}\hat{x}_{k-1|N}) + 2AB\hat{x}_{k-1|N}}{2C^2} + \frac{B^2(\Sigma_{k-1|N} + \hat{x}_{k-1|N}^2) + A^2}{2C^2}\Bigg] + \sum_{k=1}^{N}\frac{y_k^2 - 2y_k\hat{x}_{k|N} + \Sigma_{k|N} + \hat{x}_{k|N}^2}{2D^2}\Bigg) - \eta + c \tag{16}$$

Where c denotes some constant and $\hat{x}_{k|N} \triangleq \mathbb{E}[x_k|y_1,..,y_N]$, $\Sigma_{k|N} \triangleq \mathbb{V}(x_{k|N}|y_1,...,y_N)$, and $\Sigma_{k-1,k|N} \triangleq \mathbb{C}ov(x_{k-1},x_k|y_1,...,y_N)$ are smoothed estimates of the unobserved data conditioned on past, current and future observed data.

Given fully imputed data, we are able to maximise the auxillary function to obtain closed form solutions for our estimate updates. It follows that our parameters at each update are defined by $\theta_{j+1} = \arg\max Q(\theta,\theta_j)$. We perform these calculations below.

$$\begin{aligned}\frac{\partial Q(\theta,\theta_j)}{\partial A} &= -\sum_{k=1}^{N}\frac{-2\hat{x}_{k|N} + 2B\hat{x}_{k-1|N} + A_{j+1}}{2C^2} = 0\\ &\Rightarrow A_{j+1} = 2\sum_{k=1}^{N}(\hat{x}_{k|N} - B\hat{x}_{k-1|N})\end{aligned} \tag{17}$$

$$\begin{aligned}\frac{\partial Q(\theta,\theta_j)}{\partial B} &= -\sum_{k=1}^{N}\frac{-2(\Sigma_{k,k-1|N} + \hat{x}_{k|N}\hat{x}_{k-1|N}) + 2A\hat{x}_{k-1|N} + 2B_{j+1}(\Sigma_{k-1|N} + \hat{x}_{k-1|N})}{2C^2} = 0\\ &\Rightarrow B_{j+1} = \sum_{k=1}^{N}\frac{\Sigma_{k,k-1|N} + \hat{x}_{k|N}\hat{x}_{k-1|N} - A\hat{x}_{k-1|N}}{\Sigma_{k-1|N} + \hat{x}_{k-1|N}}\end{aligned} \tag{18}$$

$$\begin{aligned}\frac{\partial Q(\theta,\theta_j)}{\partial C} &= -\frac{N}{C_{j+1}} + \frac{1}{C_{j+1}^3}\sum_{k=1}^{N}(\Sigma_{k|N} + \hat{x}_{k|N}^2 - 2A_{j+1}\hat{x}_{k|N} - 2B_{j+1}(\Sigma_{k,k-1|N}\\ &\quad + \hat{x}_{k|N}\hat{x}_{k-1|N}) + 2A_{j+1}B_{j+1}\hat{x}_{k-1|N} + B_{j+1}^2(\Sigma_{k-1|N} + \hat{x}_{k-1|N}^2) + A_{j+1}^2) = 0\\ &\Rightarrow C_{j+1}^2 = \frac{1}{N}\sum_{k=1}^{N}(\Sigma_{k|N} + \hat{x}_{k|N}^2 - 2A_{j+1}\hat{x}_{k|N} - 2B_{j+1}(\Sigma_{k,k-1|N}\\ &\quad + \hat{x}_{k|N}\hat{x}_{k-1|N}) + 2A_{j+1}B_{j+1}\hat{x}_{k-1|N} + B_{j+1}^2(\Sigma_{k-1|N} + \hat{x}_{k-1|N}^2) + A_{j+1}^2)\end{aligned} \tag{19}$$

$$\begin{aligned}\frac{\partial Q(\theta,\theta_j)}{\partial D} &= -\frac{N+1}{D_{j+1}} + \frac{1}{D_{j+1}^3}\sum_{k=0}^{N}(y_k^2 - 2y_k\hat{x}_{k|N} + \Sigma_{k|N} + \hat{x}_{k|N}^2) = 0\\ &\Rightarrow D_{j+1}^2 = \frac{1}{N+1}\sum_{k=0}^{N}(y_k^2 - 2y_k\hat{x}_{k|N} + \Sigma_{k|N} + \hat{x}_{k|N}^2)\end{aligned} \tag{20}$$

Substituting (18) into (17) we obtain

$$A_{j+1} = 2\sum_{k=1}^{N}\left(\hat{x}_{k|N} - \hat{x}_{k-1|N}\sum_{i=1}^{N}\frac{\Sigma_{k,k-1|N} + \hat{x}_{k|N}\hat{x}_{k-1|N} - 2A\hat{x}_{k-1|N}}{\Sigma_{k-1|N} + \hat{x}_{k-1|N}}\right) \tag{21}$$

Which solves

$$A_{j+1} = \frac{\sum_{k=1}^{N} \hat{x}_{k|N} \sum_{k=1}^{N} (\Sigma_{k-1|N} + \hat{x}_{k-1|N}) - \sum_{k=1}^{N} \hat{x}_{k-1|N} \sum_{k=1}^{N} (\Sigma_{k-1,k|N} + \hat{x}_{k-1|N}\hat{x}_{k|N})}{N \sum_{k=1}^{N} (\Sigma_{k-1|N} + \hat{x}_{k-1|N}) - (\sum_{k=1}^{N} \hat{x}_{k-1|N})^2} \tag{22}$$

By substituting (22) into (18) we can get a update for B in terms of smoothers entirely. After some lengthy algebraic manipulations, we arrive at

$$B_{j+1} = \frac{N \sum_{k=1}^{N} (\Sigma_{k,k-1|N} + \hat{x}_{k|N}\hat{x}_{k-1|N}) - \sum_{i=1}^{N} \hat{x}_{k|N} \sum_{k=1}^{N} \hat{x}_{k-1|N})}{N \sum_{k=1}^{N} (\Sigma_{k-1|N} + \hat{x}_{k-1|N}) - (\sum_{k=1}^{N} x_{k-1|N})^2} \tag{23}$$

Detailed derivations of the updates for A and B are available upon request. We are now left with determining how to compute the smoothed estimates. We adopt a Kalman Smoothing approach, which involves initiliazing starting values for $x_{N|N}$ and $\Sigma_{N|N}$ and then recursing backwards using the following equations.

For $k \leq N$, define:

$$\mathcal{J}_k = \frac{B\Sigma_{k|k}}{\Sigma_{k+1|k}} \tag{24}$$

$$\hat{x}_{k|N} = \hat{x}_{k|k} + \mathcal{J}_k(\hat{x}_{k+1|N} - (A + B\hat{x}_{k|k})) \tag{25}$$

$$\Sigma_{k|N} = \Sigma_{k|k} + \mathcal{J}_k^2(\Sigma_{k+1|N} - \Sigma_{k+1|k}) \tag{26}$$

$$\Sigma_{k-1,k|N} = \mathcal{J}_{k-1}\Sigma_{k|k} + \mathcal{J}_k\mathcal{J}_{k-1}(\Sigma_{k,k+1|N} - B\Sigma_{k|k}) \tag{27}$$

$$\Sigma_{N-1,N|N} = B(1 - \mathcal{K}_N)\Sigma_{N-1|N-1} \tag{28}$$

To initiliaze this process, it is standard to use $\hat{x}_{N|N}$ and $\Sigma_{N|N}$ obtained from the final iteration of a Kalman Filter (Holmes, 2013; Elliott et al., 2005). That is, at each iteration of the algorithm, our initializing value for the Kalma Filter becomes more reliable and therefore so does the imputation process of the complete log likelihood. For further details of the above results, the reader may refer to Murphy (2014) or Shumway and Stoffer (2000).

We summarise the algorithm below.

EM Smoothing Algorithm

Input $y = (y_0, y_1, ..., y_n)$
Initialise $\theta_0 = (A_0, B_0, C_0, D_0)$, $\hat{x}_0 = y_0$ and $\Sigma_0 = D^2$
for $j = 0, 1, ...$ **do**
  Apply Kalman Filter with Σ_0, $\hat{x}_0$ and θ_j
  E-Step: Apply Kalman Smoothing with $\hat{x}_{N|N}$ and $\Sigma_{N|N}$ to compute $Q(\theta, \theta_j)$
  M-Step: Compute parameter updates as $\theta_{j+1} = \arg\max Q(\theta, \theta_j)$
  Update initial filter values for the following iteration as $\hat{x}_0 = \hat{x}_{0|N}$ and $\Sigma_0 = \Sigma_{0|N}$
  Continue until $\mathcal{L}(y|\theta_{j+1}) - \mathcal{L}(y|\theta_j) \leq \textit{tolerance}$ or $j =$ max iterations
Return θ

There are several remarks about this algorithm worth mentioning. Firstly, each iteration of the algorithm is guaranteed to monotonically increase the observed likelihood $\mathcal{L}(y|\theta)$. This fact is the theoretical basis for the EM algorithm. However, the algorithm is not guaranteed to converge to a global maximum, and typically the estimates are not equal to the maximum likelihod estimates which is what we are trying to achieve. Typically, the EM algorithm is not extremely sensitive to initial parameter values (Holmes, 2013), but as we have mentioned, the Kalman Filter can be sensitive to initial starting values, and given that this is a part of the E-step in this algorithm one is urged to be cautious here. Lastly, this algorithm is offline. If we wish to update our parameter estimates as we recieve new data we are required to reprocess all the previous data again. Fortunately, the algorithm is fast and so we implement such paramater re-estimation at a fixed time interval depending on the frequency of the data.

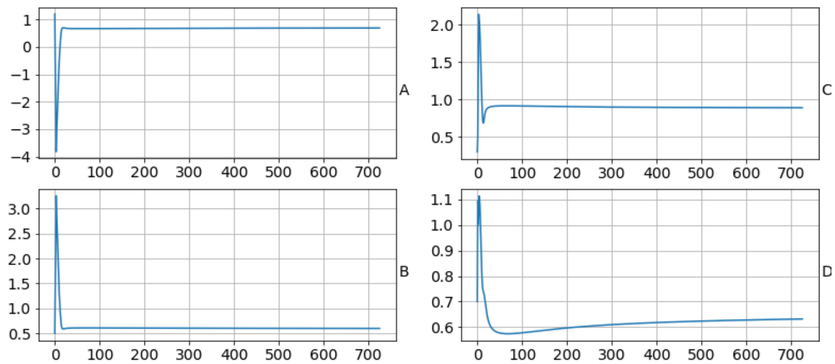

Figure 1: Illustrative convergence of the EM algorithm

3 Data and empirical findings

The following pairs of securities have been selected to train and evaluate our trading strategy.

- MDYG vs VOT: SPDR SP 400 Mid Cap Growth ETF vs Vanguard Mid-Cap Growth.
- CVX vs VLO: Chevron Corporation vs Valero Energy Corporation
- BHP vs BBL: BHP Billiton Limited vs BHP Billiton plc Sponsored ADR

For each series, we have used 252 data points, corresponding to one year of trading periods (from 03/01/2017 to 28/02/2018). We use the first 100 observations to estimate the parameters, and use the rest of the data set to evaluate the strategy based on cumulative profit. We also allow our algorithm to reestimate the parameters every k days (for illustration, we have chosen $k = 50$).

For the purposes of comparison, we have also used simulated stock price data from the following process, which is a simplified version of the log-normal model when $\mu = r = 0$:

$$dS_t = S_t \sigma dW_t,\ S_0 = a,\ t \in [0, T], a > 0, W_t \sim \text{ std Brownian motion} \tag{29}$$

We use the (exact) solution to this SDE to generate a random trajectory for the stock price:

$$S_t = S_{t-1} e^{\sigma \sqrt{\frac{T}{N}} \mathcal{N}(0,1)} \tag{30}$$

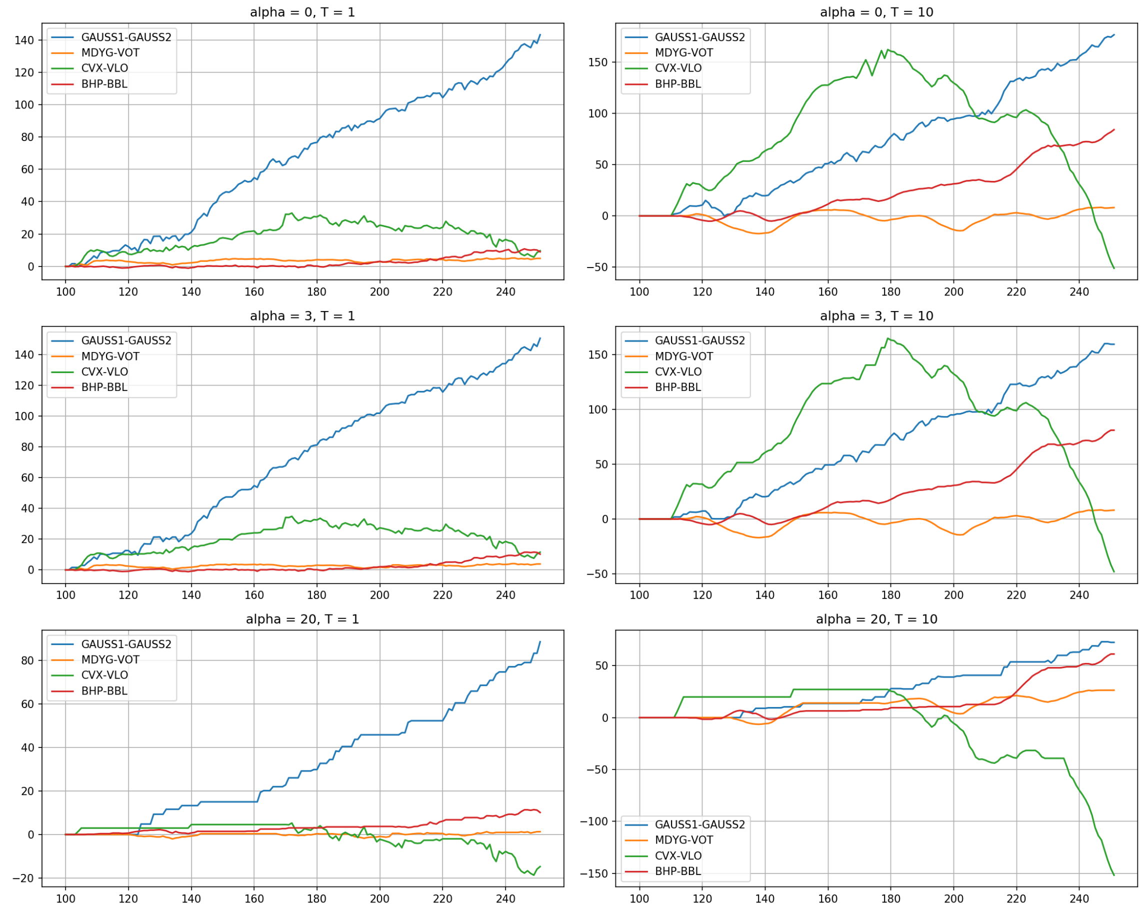

Figure 2: Cumulative profits for chosen securities using different strategies

As we would expect to see, when the threshold α increases, the trading strategy becomes more conservative and it enters a trade on a lower number of days. Recall that it only enters the trade when the standardised deviation between the observed and predicted spread is high enough. However, we do not necessarily observe better performance as we increase the threshold.

As discussed above, we have also a free parameter T for the number of days before we unwind a trade. When we choose a higher value for T, the strategy performs worse (but still good) for the simulated data, performance improves for BHP-BBL but profit becomes more extreme for CVX-VLO. This could probably be due to different time frames for mean reversion for the different series.

We may observe that if the returns of the securities are normally distributed, the trading strategy based on the Kalman filter approach is highly profitable. However, when we evaluate our strategy on real data, we observe that performance is still reasonable but not as good as with simulated data. This goes in line with our intuition, since we do not expect stock returns to be normally distributed in reality.

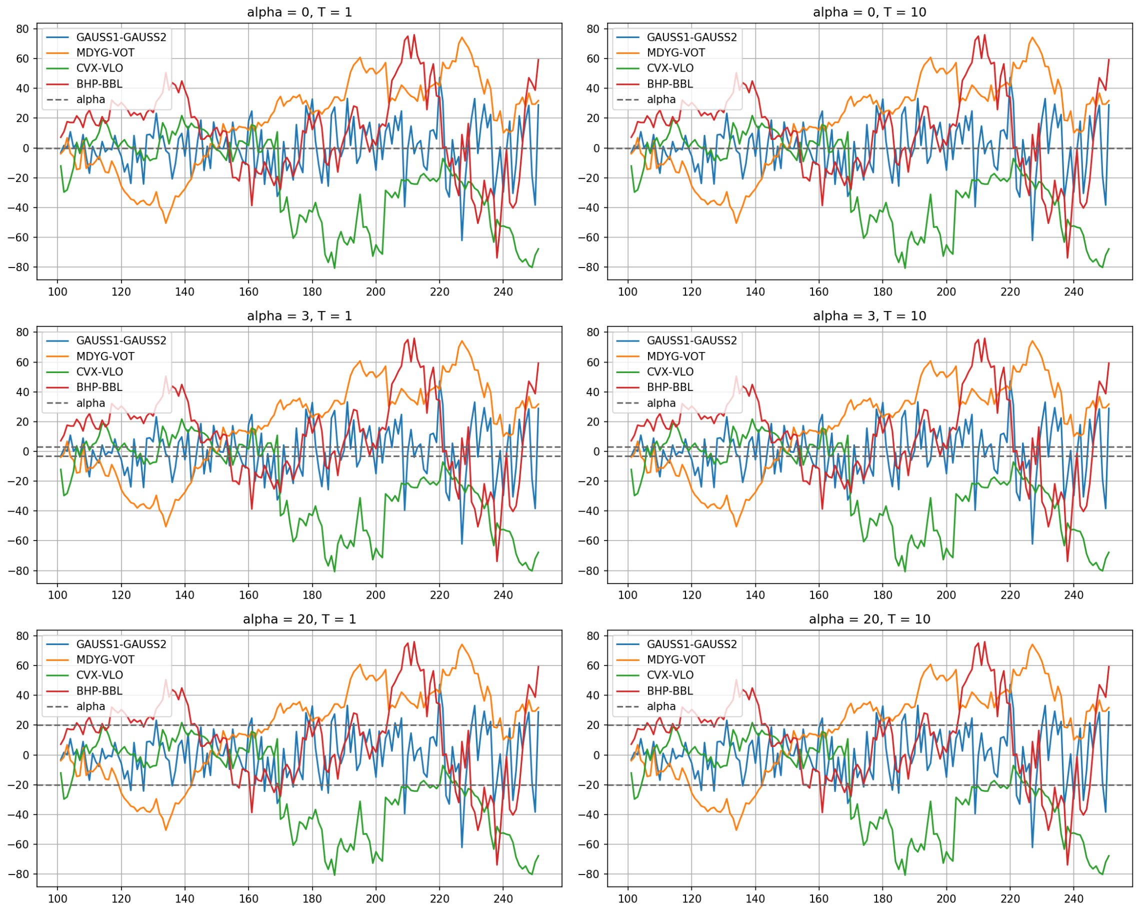

Figure 3: Standardised difference between observed spread and estimated hidden spread

In this figure, we observe the series of the standardised difference between the observed value of the spread versus its predicted value using past price information. This figure might give more intuition about what is happening behind the profit plots. The dashed lines represent the values of α and $-\alpha$. Hence, when the series is outside the dashed lines, we enter a long/short position on the spread trade, and when the series is between α and $-\alpha$, the algorithm chooses not to trade.

We can see that for the simulated data, the series oscillates more frequently between values outside and inside the α-interval, which leads to better performance. In contrast, we can see that the extreme performance we get in the CVX-VLO series may be due to the fact that the algorithm is choosing to short the spread trade a numerous amount of days.

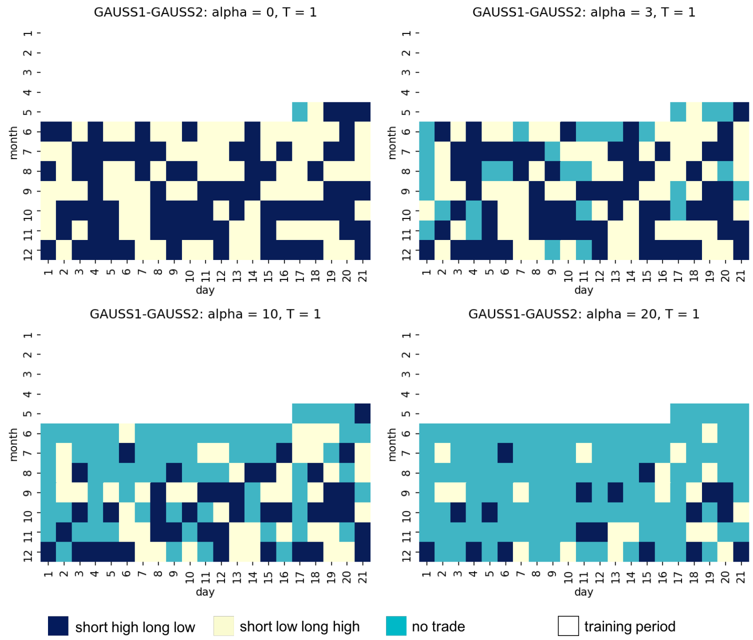

Figure 4: Strategies of the simulated Gaussian securities

For further illustration, this figure shows the evolution of our strategy for the simulated data. As we would expect to see, the algorithm is choosing not to trade more frequently as we increase the value of the α parameter, which can be interpreted as our tolerance for the uncertainty around our prediction for the spread.

4 Dynamic Programming Formulation

As a tool for further analysis, we formulated the pairs trading scheme in the dynamic programming framework. Ultimately, this is what we try to solve with heuristic strategies.

Primitives:

- s_k - price/return of 'high' stock at time k.
- y_k - price/return spread at time k.
- $\ell_k^{(i)} \in \{ \overset{long}{1}, \overset{no\,trade}{0}, \overset{short}{-1} \}$ - status of trade that was made in period i at time k.
  i.e. $\ell_k^{(i)} = 1$ means the 'high' stock was longed in period i and not unwound by period k.
- $u_k^{(i)} \in \{ \overset{hold}{0}, \overset{unwind}{1} \}$- decision made in period k about trade from period i.

State:

- $S_k = [s_0, \dots, s_k]$ - vector of price/return of the 'high' stock at time k
- $Y_k = [y_0, \dots, y_k]$ - vector of price/return spreads
- $L_k = [\ell_k^{(0)}, \dots, \ell_k^{(k-1)}]$ - current open trades.

Action:

- $U_k = [u_k^{(0)}, \dots, u_k^{(k-1)}]$ - vector of decisions to unwind previous trades.
- $\ell_k^{(k)}$ - decision on whether to short, long, or hold in current period.

Action Constraints:

- We enforce that only open trades can be unwound and, for ease of notation, refer to all constraints as Δ.

$$
\begin{aligned}
\Delta = \{U_k, \ell_k^{(k)} \ : \ & u_k^{(i)} \in \{0, 1\}, \\
& u_k^{(i)} \leq |\ell_k^{(i)}|, \\
& \ell_k^{(k)} \in \{-1, 0, 1\} \quad \forall i = 0, \dots k-1\}
\end{aligned}
$$

Dynamics:

- We assume the spread follows the model developed earlier in the report:
  $y_{k+1}|Y_k \sim \mathcal{N}(A + B\hat{x}_k, C^2 + D^2)$
- If a trade is unwound in this period, we update the vector L_k
  $\ell_{k+1}^{(i)} = \ell_k^{(i)} - \ell_k^{(i)} u_k^{(i)}$
- s_{k+1} is updated with an unknown distribution.

Cost:

For ease of notation, let $\phi_k = (S_k, Y_k, L_k, U_k, \ell_k^{(k)})$

$$
g_k(\phi_k) = -\sum_{i=1}^{k-1} u_k^{(i)} \ell_k^{(i)} (y_k + s_k - \frac{(y_i + s_i)}{s_i} s_k)
$$

Dynamic Programming Algorithm:
In the last step, we close out any remaining open bets:

$$J_N(S_N, Y_N, L_N) = -\sum_{i=1}^{N-1} \ell_k^{(i)}(y_k + s_k - \frac{(y_i + s_i)}{s_i} s_k)$$

Then update the strategy through backwards recursion:

$$J_k(\phi_k) = \min_{U_k, \ell_k^{(k)} \in \Delta} \mathbb{E}\left[g_k(\phi_k) + J_{k+1}(\phi_{k+1})\right]$$

However, the solution for this problem is non-trivial. One needs to assume a distribution for the two price trajectories of the securities, and also we need to assume how they move together (e.g. assume that they have a co-integration factor in common). Additionally, even if we modelled the distribution of both securities with their co-integration factor, it would still be relatively hard to solve the problem given the characteristics of the constraint set. Some attempts have been made in the literature in continuous time (see e.g. Mudchanatongsuk et al., 2008), which model the difference between the log-returns of two assets as an Ornstein-Uhlenbeck process and compute the optimal position as a function of the deviation from the equilibrium (as we do with the heuristic rules). Technically, this is done by solving the corresponding Hamilton-Jacobi-Bellman equation. Finally, note that those attempts also require making assumptions about the form of the utility function of wealth, so as to obtain a measure of risk-aversion. For example, it is common to use a CARA utility function (Constant Absolute Risk Aversion).

5 Conclusions

In this project we have explored how to apply the Kalman filter to Pairs Trading, a typical investment strategy employed by various hedge funds. We have presented the model for the spread with the Kalman filter equations, and derived a natural trading strategy using the (standardised) residual between the observed spread and the predicted spread given past price information. We have shown that we can estimate the parameters of the Kalman equations by use of the EM Algorithm, and finally we have applied our strategy on real and simulated data to evaluate its performance. We have found that this simple algorithm is capable of delivering positive cumulative profit for different series of security prices which are 'close' to each other. We have observed that the algorithm performs better with simulated data from a price trajectory that follows a martingale. However, we observe that performance is not as great with real data but it still delivers relatively good results.

The relative simplicity and ease of interpretation of our Kalman Filtering approach implies that the model comes with limitations too. As we have discussed in section 3.3, the Kalman filter is assuming that the dynamics of the spread process are linear with Gaussian noise. However, evidence about financial returns suggests that this is not the case. Therefore, one should not be surprised to observe that the trading strategy performs relatively worse with real data. As suggestions for further research in this area, we point out to the use of more general Hidden Markov Models that are able to capture non-linear relationships and non-gaussian shocks. In particular, we believe that an extension of the model able capture the fact that financial returns tend to have fat tails would perform better than the Kalman filter approach.

We are also aware that we could extend the model so as to be able to determine the optimal amount of each asset that one would long/short at each point in time. In addition, it would be interesting to make extensions of the model that not only determine the optimal positions to take but which are also able to determine which pairs to use out of a given number of securities.

References

Anderson, B. D. and Moore, J. B. (1979). Optimal filtering. *Englewood Cliffs*, 21:22–95.

Do, B. and Faff, R. (2010). Does simple pairs trading still work? *Financial Analysts Journal*, 66(4):83–95.

Do, B. and Faff, R. (2012). Are pairs trading profits robust to trading costs? *Journal of Financial Research*, 35(2):261–287.

Elliott, R. J. and Krishnamurthy, V. (1997). Exact finite-dimensional filters for maximum likelihood parameter estimation of continuous-time linear gaussian systems. *SIAM Journal on Control and Optimization*, 35(6):1908–1923.

Elliott, R. J., Van Der Hoek*, J., and Malcolm, W. P. (2005). Pairs trading. *Quantitative Finance*, 5(3):271–276.

Engle, R. F. and Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pages 251–276.

Gatev, E., Goetzmann, W. N., and Rouwenhorst, K. G. (2006). Pairs trading: Performance of a relative-value arbitrage rule. *The Review of Financial Studies*, 19(3):797–827.

Gatev, E. G., Goetzmann, W. N., and Rouwenhorst, K. G. (1999). Pairs Trading: Performance of a Relative Value Arbitrage Rule. NBER Working Papers 7032, National Bureau of Economic Research, Inc.

Holmes, E. E. (2013). Derivation of an em algorithm for constrained and unconstrained multivariate autoregressive state-space (marss) models. *arXiv preprint arXiv:1302.3919*.

Jwo, D.-J. and Cho, T.-S. (2007). A practical note on evaluating kalman filter performance optimality and degradation. *Applied Mathematics and Computation*, 193(2):482–505.

Mudchanatongsuk, S., Primbs, J. A., and Wong, W. (2008). Optimal pairs trading: A stochastic control approach. In *American Control Conference, 2008*, pages 1035–1039. IEEE.

Murphy, K. (2014). Machine learning, a probabilistic perspective.

Reverre, S. (2001). *The complete arbitrage deskbook*. McGraw-Hill Professional.

Shumway, R. H. and Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis*, 3(4):253–264.

Shumway, R. H. and Stoffer, D. S. (2000). Time series analysis and its applications. *Studies In Informatics And Control*, 9(4):375–376.

Vidyamurthy, G. (2004). *Pairs Trading: quantitative methods and analysis*, volume 217. John Wiley & Sons.