
*Research article***Tick based clustering methodologies establishing support and resistance levels in the currency exchange market****Karl Tengelin and Alexandros Sopasakis***

Department of Mathematics, Lund University, Lund 22100, Sweden

* **Correspondence:** Email: alexandros.sopasakis@math.lth.se; Tel: +46462220000; Fax: +46462224010.

Abstract: We establish support and resistance levels from data in intraday currency exchange market activity based on machine learning methods. Specifically we design two semi-supervised classification neural networks. The first one is based on a variant of the K-means method while the second is based on a Gaussian mixture model with expectation maximisation. Each performs classification from tick data on very short time windows and produces the corresponding support and resistance price levels. We test the methodology on actual market data for the EUR-USD currency exchange. As a sanity check we also perform mock trades based on actual market data. We evaluate the results for statistical significance using a number of performance metrics while also comparing against traditional methods.

Keywords: tick data; support and resistance levels; clustering methods; Gaussian mixture model; K-means; EUR-USD exchange rates

JEL Codes: C45, C63, C81, C83

1. Introduction

Currency investment strategies tend to be more profitable than equivalent strategies in other asset classes as found by Taylor (2005) and Levich and Potì (2015). It is noted that for currency markets predictability often violates the theoretical upper bound attributed to absence of “good deals” (Taylor, 2005). In other words good deals are available and for brief time intervals “predictability is higher than one would expect under the efficient market hypothesis” (Taylor, 2005). On the other hand, as is well known, Forex trading returns were significantly reduced during the 1990s (Neely and Weller, 2013). However since 1998 Forex returns have been superior to equity returns. This is especially true in the case of Forex technical trading for emerging markets (Neely and Weller, 2013).

We examine here short-term investment strategies in currency markets through machine learning type technical trading. Technical trading implies that we study historical patterns in actual market

data via machine learning methods and subsequently implement our findings in real time such data. Specifically we establish so called support and resistance levels, from Sullivan et al. (1999), based on semi-supervised Gaussian mixture models and K-means methods. A support level is defined by Neely and Weller (2013) as the price for which the asset is considered to be reluctant to fall below. Clearly this is an anthropogenic definition since the asset has no such will of its own. In other words traders have some kind of intuition or feeling in terms of what the lowest value an asset can take during a given time interval and call that value a support level. Traders often consider investing in the asset if the price comes close to a support level since they believe that the asset will soon rebound thus making a profit for them. A support level is also known as a channel rule following Taylor (2005). Correspondingly the resistance level is a level which the asset is considered to be reluctant to rise above. Typically investors sell the asset as it comes closer to a resistance level. Sullivan's definition of support and resistance levels is similar to Taylor's although "the minimum and maximum values [of the asset] over a certain number of n days must be within a certain distance of each other" (Neely and Weller, 2013). In our case the length between minimum and maximum values or rather support and resistance levels is not computed within days but instead just mere seconds as will be seen in the Results section later. We also point out that we do not take into account of any transaction costs and its effects on perceived opportunities which is discussed in more detail in Frenkel and Levich (1975). The asset is allowed to touch these support and resistance levels but should bounce back for the duration of the predefined trading window. The length of that window is naturally very hard to predict and the focus of many studies. We do not address the question of the length of the window in this work but instead focus on building tools which find the support and resistance levels based on data analysis by machine learning methods.

Tick data is used to describe the flow of orders registered in the order book. The order book contains dated prices of a given asset. Tick values do not distinguish between an order to buy (otherwise called bid) or an order to sell (otherwise called ask). Tick data is commonly used to represent activity in a specific market. A pip, representing the minimal incremental change of a currency, should not therefore be confused with a tick.

Our interest lies with the top of the order book which consists of the lowest ask-level and the highest bid-level at the current time. These values are of interest since they are directly linked to an eventual trade for that asset. Another quantity of interest is the bid-ask spread. The bid-ask spread represents the difference between the highest bid and lowest ask prices. In this manuscript asset price is computed by averaging the bid-ask spread.

To further motivate our approach and shed some light in the information content within ticks we also present here some basic statistics between tick and price. In particular we present information about correlation as well as variance in price in order to explicitly show that indeed activity in the form of ticks can be another indicator about price fluctuations. Specifically the Pearson correlation is calculated to be 0.75 between tick activity and price variance for the EUR versus USD exchange rate market for data taken during one single day of trading in January 2019. The Spearman correlation with an even better value of 0.83 similarly shows a positive relationship between these quantities. These correlation coefficients therefore indicate that if the number of ticks increases then so should the variance of the EUR/USD exchange rate price.

Our aim is to use machine learning methods in order to discover hidden features in historical exchange rate and tick activity information. Succeeding in this task will then allow us to establish support and resistance levels in future exchange rates.

We begin in Section 2 by providing some of the background for the machine learning methods to be employed here. We then discuss results and applications in the currency exchange market data in Section 3. We follow up with a discussion of these results in Section 4.

2. Methodology

2.1. Semi-supervised classification and metrics

A number of methodologies exist capable of performing classification from data. Auto-encoders in general are well suited (Norlander and Sopasakis, 2019) to such a task. In this section we outline two of most prominent methods: K-means and Gaussian Mixture Models (GMM). One of the main reasons for choosing either of those methods is that they are both essentially unsupervised methodologies. As a result these methods are capable of automatically extracting the optimal number of classes from patterns within the data in a seemingly unsupervised way. The only reason that they are not truly unsupervised is that we must manually provide the number of possible clusters in the data once some initial tests, which we outline below, are performed.

2.2. K-Means type classification

One of the most well known classification methods is K-means clustering. It is based on Lloyd's algorithm (Norlander and Sopasakis, 2019; Lloyd, 1982) and is considered to be a hard type of classification methodology in the sense that each data point either belongs or not to a specific cluster. This is in contrast to, for instance, Variational Auto-encoders (Norlander and Sopasakis, 2019) where data belong to a cluster based on a probability. K-means clustering starts with a specified number of cluster centers and adds successive data to each such cluster based on nearest Euclidean distance. After each new addition the cluster centers are updated and the process repeats until all data has been assigned to the corresponding clusters. Such an example of clustering from exchange rate data can be seen in Figure 1.

We instead apply Lloyd's algorithm in only one dimension for our data. Specifically we update the center of mass per cluster by processing information only from the price of the asset. The algorithm we therefore propose (see Supplementary material) calculates the new center of mass x_c for each cluster from $x_c = \sum_{l=0}^N x_l m_l / M$ where $M = \sum_{l=0}^N m_l$ and m_l denotes the x-coordinate of data point l with mass m_l . Note that m_l denotes activity and x denotes price in the case that the data represents exchange rates. The resulting classification for a set of exchange rate data between USD and EUR is shown in Figure 2.

The method above requires that the number of clusters be provided for each data set. To identify the correct number of clusters we test our data under four different metrics: ELBO method (Evidence based Lower Bound) (Alind, 2020), Silhouette score (Rousseeuw, 1987) Calinski-Harabasz (Calinski and Harabasz, 1974) and finally the Davies-Bouldin method (Thomas et al., 2013). The methods give a numerical value corresponding to how well the data matches the different number of clusters. We tested our data for number of clusters ranging between 4 and 20 and present the results in Figure 3. It is common praxis for traders to use between 4 and 20 clusters in their market analysis although the optimal number of clusters is considered a trade secret. Some traders use just 10 clusters most of the time. Clearly however less than 4 clusters can be too few to provide a clear picture and more than 20 tends to complicate the analysis.

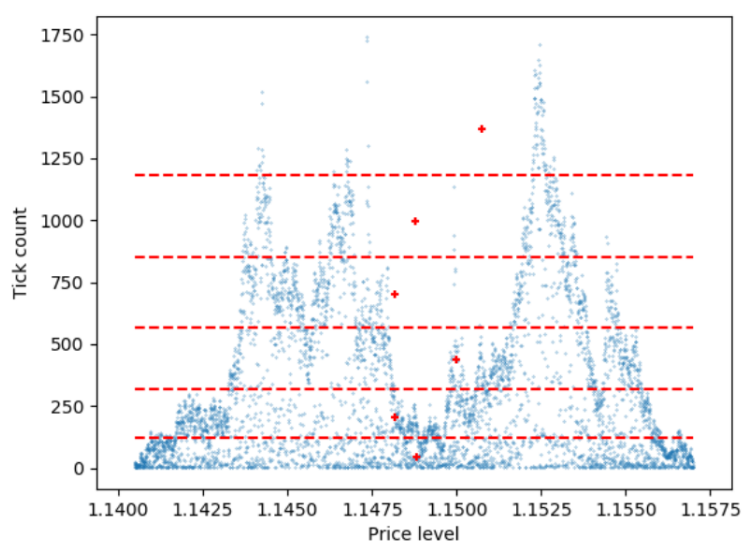


Figure 1. Clustering of USD versus EUR exchange rate data based on the classic Lloyd's algorithm (Lloyd, 1982). Clusters are produced by considering both the price and corresponding tick data for 1 hour during day trading in 2018 (see also Figure 5 in Tengelin (2020) for details). The horizontal lines indicate the cluster bounds after the process is complete. Based on this and many other similar results, not presented here, the algorithm does not seem to uncover clear ranges for support and resistance levels (compare also with improved method in Figure 2).

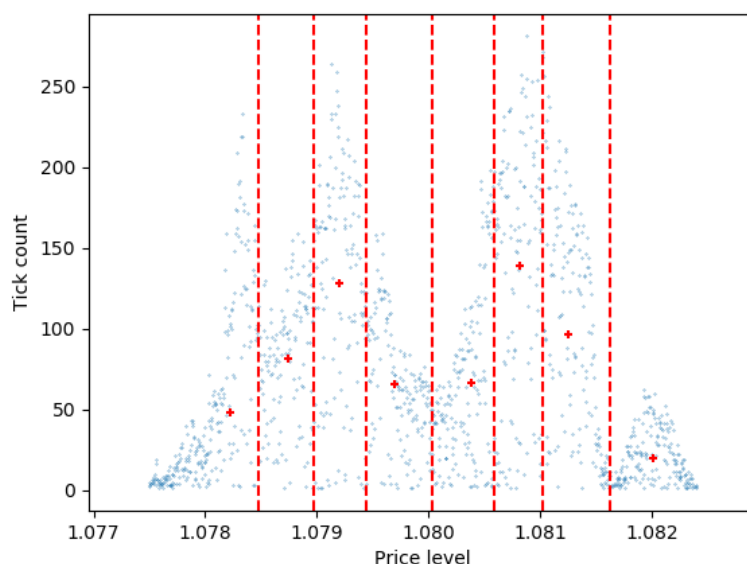


Figure 2. Clustering is performed via a modified Lloyd's algorithm for exchange rates between USD and EUR. These 8 clusters are produced by analysing the tick data for 1 hour starting at 11 am on May 14, 2020. The horizontal lines separate the different clusters. Support and resistance levels are now more meaningful by simple inspection (compare also with classic Lloyd's method in Figure 1).

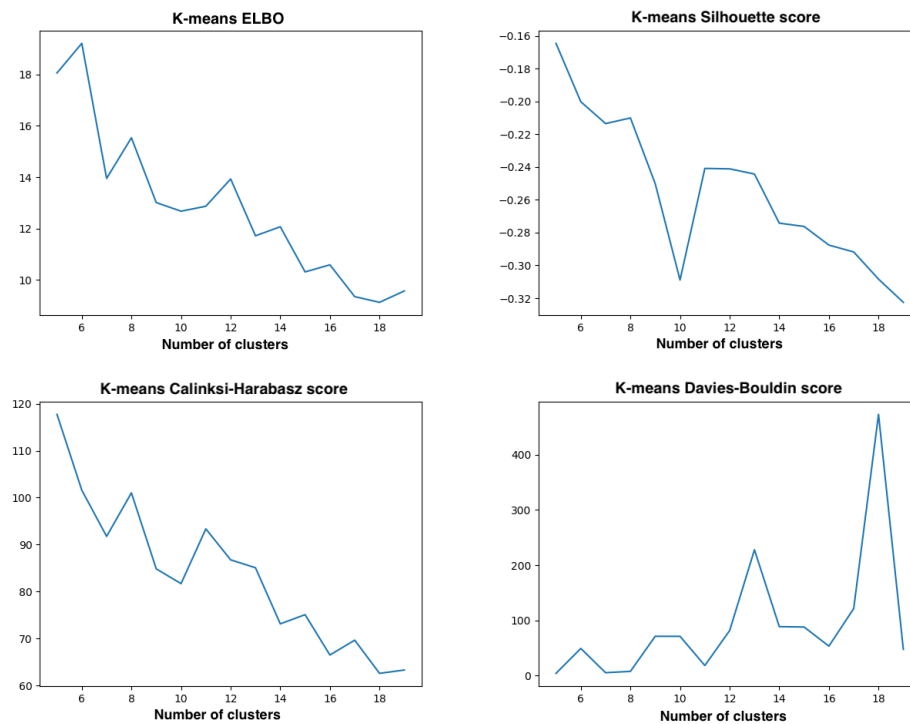


Figure 3. The four metrics above show which is the most likely number of clusters for this USD/EUR exchange rate data. The last metric shows the optimal number of clusters as a minimum instead of a maximum. The suggested number of clusters for the first 3 figures are indicated to be in order: 6, 5, 5. In the last figure any of 5, 7 or 8 clusters are suggested to be best. Based on these results we therefore choose to classify this dataset with 5 clusters.

These metrics indicate that we should use respectively 6, 5, 5 and 5 number of clusters. As a result we decided to use 5 clusters for our data.

2.3. Gaussian mixture models classification and metrics

An alternate method to perform clustering is to use GMM. The GMM method relies on the Expectation Maximisation (EM) principle in order to assign data to clusters. Specifically GMM relies on the idea that each data point $\mathbf{x} = \{x_i, \dots, x_K\}$ should belong to a specific Gaussian distribution whose each member follows,

$$f_k(x) = \frac{1}{\sqrt{|\Sigma|}2\pi} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right) \quad (1)$$

where T here denotes the transpose $\mu = \{\mu_i, \dots, \mu_K\}$ is the mean and $\Sigma = \{\Sigma_i, \dots, \Sigma_K\}$ is the covariance of each of the distributions. Note that neither of those will be initially known. A GMM approach strives to compute the optimal mean μ for each such distribution as well as its corresponding covariance Σ which specifies the width of that distribution. Finally also the mixing probability vector π between these distributions must also be computed since it will specify how large the Gaussian function should be. One way to establish those is to use a VAE (Norlander and Sopasakis, 2019) in order to automatically discover those but that method is computationally intensive and may be prone to instability. A much

easier way is to instead use the Expectation Maximisation (EM) method (Dempster et al., 1977) The EM method is an iterative algorithm which optimises a log-likelihood type function (Dempster et al., 1977); Tengelin (2020) in order to obtain the best such μ and Σ . Specifically the EM method optimizes the logarithm of (1),

$$\ln f_k(\mathbf{x}) = -\frac{1}{2} \left(\ln 2\pi + \ln \Sigma_k + (\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k) \right) \quad (2)$$

By differentiating with respect to μ and Σ . The method iterates starting from random values until it identifies the best mean and covariance vectors which in turn specifies each of these clusters for the given data. The expectation maximization part guarantees that the complexities of the above objective function are solvable (Dempster et al., 1977). The solutions obtained correspond to the maximum likelihood estimates.

As a first step in order to prepare our data towards representation by the GMM method we first normalise them. We present the result of such a data normalization in Figure 4.

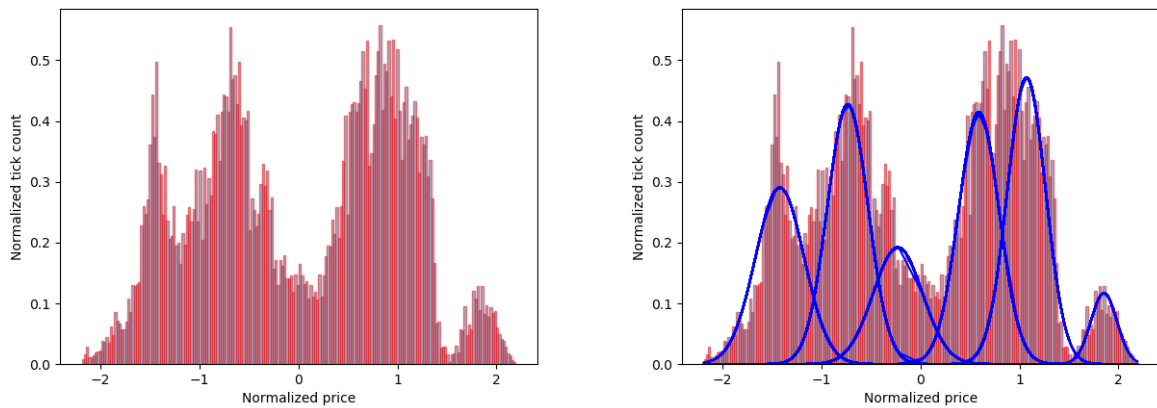


Figure 4. (Left) Normalising the tick versus price dataset before analysis by the GMM method. (Right) The resulting fitting of the normalised data by 6 Gaussian distributions via the GMM method.

In order to compute the mixing probabilities we impose the condition $\sum_{k=1}^K \pi_k = 1$ which corresponds to the normalization step we above.

To evaluate the resulting clustering and overall performance of the GMM method we propose two different metrics. Specifically we compute the W-statistic by Shapiro and Wilk (1965),

$$W = \frac{(\sum_{i=1}^n a_i x_{(i)})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (3)$$

where $x_{(i)}$ is the smallest value in the sample of size n . Here \bar{x} is the sample mean and each of the a_i is computed from,

$$(a_1, \dots, a_n) = \frac{e^T V^{-1}}{(e^T V^{-1} V^{-1} e)^{1/2}} \quad (4)$$

Here V denotes the covariance matrix between all order statistics and e is a vector $e = (e_1, \dots, e_n)^T$ where each e_k is the expected value of each order statistic sample independently drawn from a standard

normal distribution. A high value of this W-statistic (3) indicates stronger evidence that the sample distribution is similar to a Gaussian distribution.

An alternative to the W-statistic is the Kolmogorov-Smirnov (KS) score. The KS score is used to indicate similarity between the sample distribution and the Gaussian distribution. The corresponding test statistic D is defined via,

$$D_n = \sup_x |F_n(x) - F(x)| \quad (5)$$

where $F(x)$ denotes the CDF for the normal distribution $N(\mu, \sigma)$ and $F_n(x)$ is the empirical distribution function based on n observations X_i ,

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n I_{[-\infty, x]}(X_i), \quad (6)$$

where $I_{[-\infty, x]}$ is the indicator function defined to be 1 if $X_i \leq x$ and 0 otherwise.

The W-statistic seems to perform better (Razali and Wah, 2011) than the D-statistic for most problems. For large data samples however, similar to those we use in this work, the D-statistic has been shown to be a better choice (Samuels, 2015). Nevertheless we present results from both of the above statistics for completeness and to aid comparisons.

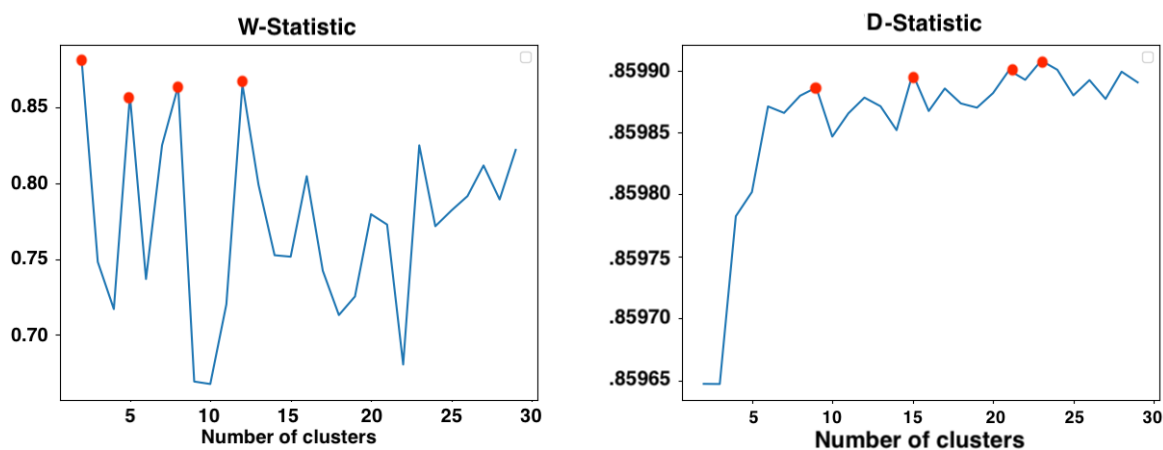


Figure 5. For both statistics a high value indicates the preferred number of clusters for the GMM method. The W-statistic on the left indicates that any of 3, 5, 8 or 13 clusters are sufficient to describe this data. The D-statistic on the right, based on the KS-score, indicates 8, 15, 20 or 23 clusters. Clearly however most all clusters above 5 would present sufficiently good options. Dataset used is that same as the one in Figure 3.

Based on the results from Figure 5 we observe that for this dataset both statistics would be satisfied if we use 8 clusters. Even 13 clusters would work.

3. Data and results

In order to test our conclusions we use both synthetic as well as real market data. The real data is obtain by directly downloading from the major indexes of the forex currency exchange market. The data is specifically comprised of several columns and thousands of rows. We only used three columns for our pattern analysis in this paper consisting of:

time in milliseconds; bid price; ask price.

This information is processed in order to produce support and resistance levels based on the methodology described in Section 2. The synthetic data is formatted in exactly the same way as the actual data imported from forex. We process these datasets with our own python codes using the library Pandas.

In this work a pip has a value of 0.0001 and we work with price changes in the range of 10 pips which therefore represent a price change in the range of 0.001.

We apply each of the methods presented in Section 2 in order to establish support and resistance levels for the USD/EUR exchange rate market. We compare the results against each other but also against the so called Trendline method.

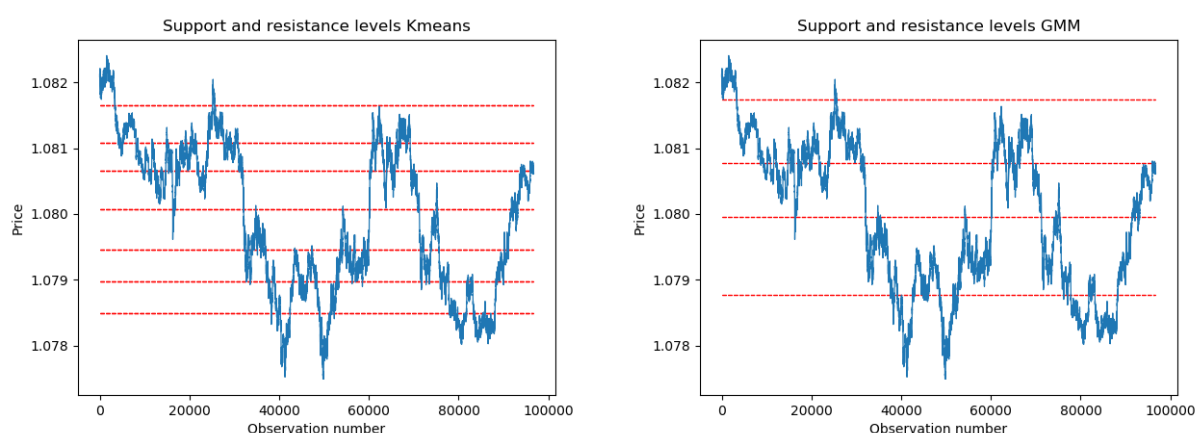


Figure 6. Partitioning with respect to price based on short-term historical tick data. (Left) The K-means method. (Right) The GMM method.

Note that the Trendline method is an ad-hoc procedure to establish subjective support and resistance levels based on short-term visual observations of historical price data. We present below two examples of support and resistance levels established by the K-means method and the GMM method in Figure 6.

We also present, as an example a result of the Trendline classification in Figure 7 where we have drawn the respective support and resistance levels after visually inspecting price fluctuations over 1.6 million observations.

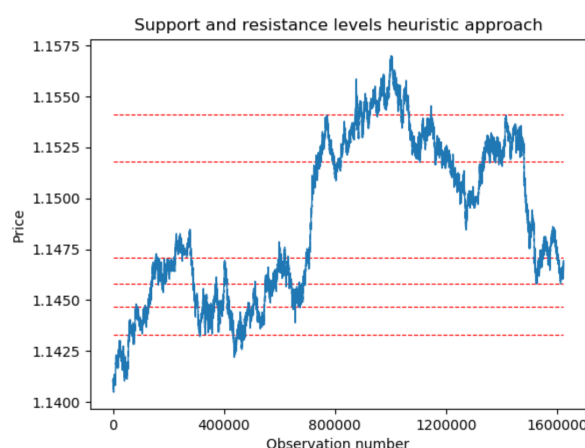


Figure 7. The Trendline method. Accounting for short-term historical price observation the user draws by hand their believed support and resistance levels. Visually at least the results look reasonable.

3.1. Target and stop loss under mock trades

To evaluate the success or failure of our data classification we apply classic methods from trading practices. Specifically we compare the number of times for which we reach a “target” or a “stop loss” during a series of trades. A target is a predefined price for an asset for which a trader aims to alter their position after purchasing the asset. Clearly the trader would like to purchase low and sell at a higher value in order to realise a profit. That higher value is an example of a target. A stop loss (SL) is very different in scope and its purpose is as a safety for the trader in order to limit their losses. As an example therefore the trader would, upon purchase, define a lower price which if reached the trader will sell the asset. That lower price is the SL price and it protects the seller from incurring much bigger losses.

The results in this section comes from initiating mock trades within specific time windows. For this work we chose the size of the time windows to vary randomly around 12 hours. Other length time windows have also been tried and similar results are found as those which will be presented below. In some of the results which we present below we encounter situations for which neither a target nor a SL is established within the window of our analysis. In that case we mark this inconclusive result as a null event and we mark it as such in our analysis below.

In Figure 8 we present a visual representation of how the mock trades are performed. Those cases where the price moves “through” a support and resistance level by more than 10 pips within the time window we count it as a stop loss. If the price on the other hand “bounces” at a sup/res-level by more than 10 pips that result is noted as hitting target.

We present in Table 1 results from a P-value test in order to evaluate the statistical significance of our results. We compute the P-value both for the entire 5 months investigated as well as for each month separately.

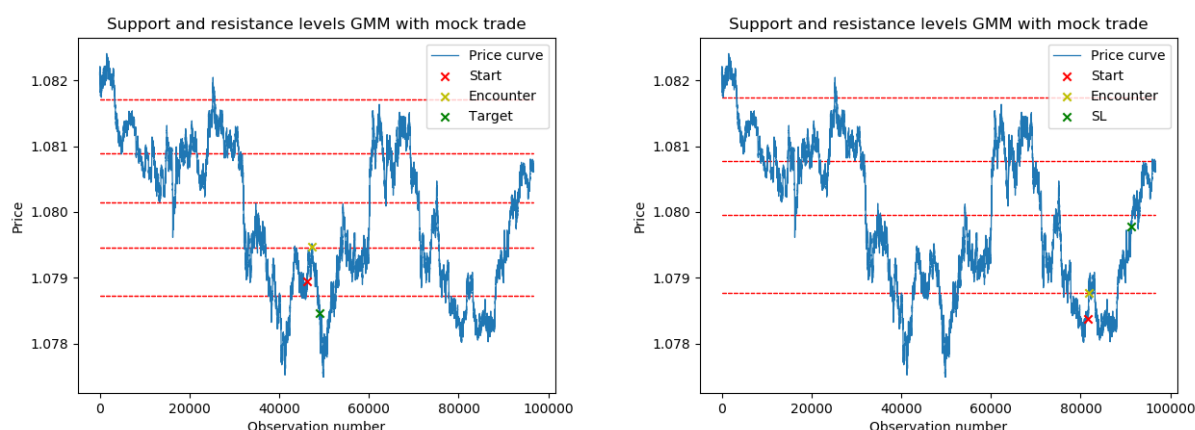


Figure 8. Illustrates how the target and stop loss-scores are determined. The left graph illustrates a start point (where the target vs stop loss investigation starts) , an encounter-point where the price reaches a support/resistance level and the target point, which is reached after moving 10 pips away from the encounter point. The right graph illustrate the same points apart from the target point which has been exchanged with a SL-point.

Table 1. The number of mock trades initiated during the first five months of 2019 in order to test the success or failure for each of the two methods. Failures are associated here with a Null event. These mock trades are simulated in real time as information from actual market data arrives.

Month	GMM			K-means		
	Target	SL	Null	Target	SL	Null
Jan	84	57	7	44	44	2
Feb	48	35	2	41	38	6
Mar	33	46	6	36	43	6
Apr	42	36	7	48	32	5
May	42	37	6	41	40	4
Total	249	211	28	210	197	23
Total (%)	51.0	43.2	5.7	48.8	45.8	5.3

Table 2. P-values for the GMM and the K-means method for the mock trades shown in Table 1 corresponding to the first five months of 2019. In the last row we present the combined P-value score for the whole period.

Month	P-val. GMM	P-val. K-means
Jan	0.014	0.542
Feb	0.094	0.411
Mar	0.943	0.816
Apr	0.286	0.046
May	0.326	0.500
Total	0.042	0.276

For completeness the P-values corresponding to the mock trades simulated in Table 1 are computed and presented in Table 2.

If we start by looking at Table 1 one could argue that since the percentage of times we hit target is higher than the percentage of times we hit stop loss we have succeeded in making a profitable model in both cases (GMM and K-means). However if we look at the results in Table 2 we get the notion that this might not be the case. What Table 2 says is that, especially in the case of the K-means model, the results might as well be random. In the case of the GMM the results are a bit more intriguing. We discuss these findings in context of the literature in the following section in more detail.

4. Discussion

We presented the K-means and the GMM method both of which extract classification information in a self-supervised way based on price versus tick data from the EUR/USD exchange rate market. Subsequently we test their performance during mock trades in detecting support and resistance levels from which we can establish our strategy

The classification performed by either of those methods seems to find support and resistance levels which can compare favourably against classic trading methods such as the Trendline method. Also the fact that both of the methods we present are able to independently produce similar support and resistance levels for the same dataset is further evidence that the results are not so random after all and are reproducible. We should point out here that in general the number of clusters employed by each method can be very different. If however it so happens that the number of clusters used by each methods are the same then we do observe the similarities discussed above.

It is however the mock trades which give practical information about the implementational success of the methods. The mock trades shown in Table 1 are based on initiating trades at the same time as information from actual market data arrives. These trades are indicative that the method is successful since most of the time they hit their target values. On the other hand the P-values in Table 2 are not within the usual 5% or 10% level of significance. The K-means method in particular seems to produce rather large P-values. GMM displays slightly better P-values but still not always within the typical ranges of 5–10%. In fact, as can be seen in Table 1, we seem to have the worst P-values for both methods during the month of March. This is also in agreement in terms of March being the only month in Table 1 for which we reach a stop loss more times than we reach a target.

A high P-value however does not necessarily imply that the methods have failed. On the contrary, we note that in total both of our models reach a target value more times than they reach a stop loss or a null event (last line of Table 1). Based on these findings therefore both models would be profitable although clearly the GMM shows superior results. Given the volatility of the exchange rate markets as well as a number of exogenous factors which we have not accounted for in our data it is perhaps not surprising that we do not observe better results in our testing. There were in total 459 mock trades which were not sufficiently many in order to obtain a complete picture of how results vary and more importantly which factors are responsible for that variation. As a result this aspect of the work is of particular interest and will be carried out in a follow up publication.

Clearly trading strategies evolve and adapt to take advantage of any such possible advantages in the foreign exchange markets. This is essentially the premise of the adaptive market hypothesis (Neely and Weller (2013)) which we also adhere to. As a result we expect that the small potential benefits of

the investment modeling strategies observed here will inevitably need to be adapted in the near future in order to remain competitive. These ideas are outlined in Neely and Weller (2013) and include diversification among different forex trading strategies. In general however adaptive strategies should outperform nonadaptive strategies based on the Adaptive Market Hypothesis (Levich and Potì (2015); Neely and Weller (2013)). We note also here that we did not include transaction costs or demand and supply elasticities in the forex market. Had we done so it is very likely that these costs would account for at least a portion of the observed profit opportunities (Frenkel and Levich (1975)). We leave the investigation of how large this portion is to future research work.

Acknowledgments

We would like to thank all three anonymous referees for their inciteful comments which greatly enriched the manuscript.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

- Alind G (2020) Elbow Method for optimal value of k in KMeans, GeeksforGeeks. Available from: <https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans>.
- Arsenault MO (2017) Kolmogorov-Smirnov test. Towards data science. Available from: <https://towardsdatascience.com/kolmogorov-smirnov-test-84c92fb4158d>.
- Caliński T, Harabasz JA (1974) A Dendrite Method for Cluster Analysis. *Commun Stat Theory Methods* 3: 1–27.
- Dempster AP, Laird NM, Rubin DB (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. Available from: https://www.ece.iastate.edu/~namrata/EE527_Spring08/Dempster77.pdf.
- Frenkel JA, Levich RM (1975) Covered interest arbitrage: unexploited profits? *J Polit Econ* 83: 325–338.
- Levich RM, Potì V (2015) Predictability and ‘good deals’ in currency markets. *Int J Forecasting* 31: 454–472.
- Lloyd S (1982) Least Squares Quantization in PCM. *IEEE Tran Inf Theory* 28: 129–137.
- Neely CJ, Weller PA (2013) Lessons from the evolution of foreign exchange trading strategies. *J Bank Financ* 37: 3783–3798.
- Norlander E, Sopasakis A (2019) Latent space conditioning for improved classification and anomaly detection. arXiv:1911.10599 in Computer Science and Machine Learning.

- Razali NM, Wah YB (2011) Power comparisons of Shapiro-Wilk, Komogorov-Smirnov, Lilliefors and Anderson-Darling tests. *J Stat Model Anal* 2: 21–33.
- Rousseeuw P (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Available from: <https://www.sciencedirect.com/science/article/pii/S0377042787901257>.
- Samuels P (2015) How do we know which test to apply for testing normality? Available from: https://www.researchgate.net/post/How_do_we_know_which_test_to_apply_for_testing_normality.
- Shapiro SS, Wilk MB (1965) An analysis of variance test for normality (complete samples). *Biometrika* 52: 591–611.
- Sullivan R, Timmermann A, White H (1999) Data-Snooping, Technical trading rule performance, and the bootstrap. *J Financ* 54: 1647–1691.
- Taylor SJ (2005) *Asset price dynamics, volatility and prediction*, Princeton, NJ: Princeton University Press.
- Tengelin K (2020) Tick data clustering analysis establishing support and resistance levels of the EUR-USD exchange market. Master Thesis, Lund University Publications in Lund, Sweden.
- Thomas JCR, Mora M, Sandos M (2013) New Version of Davies-Bouldin Index for Clustering Validation Based on Cylindrical Distance. In 2013 32nd International Conference of the Chilean Computer Science Society (SCCC), IEEE, 49–53.



AIMS Press

© 2020 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)