FX Arbitrage Detection Through Linear Programming

Melique Daley

1 Introduction

Arbitrage is a rare event when there's price differences between two or more markets. We can use these price differences by finding a combination of deals that take advantage of these imbalances. In the context of foreign exchanges, we can successively convert a currency into multiple currencies then convert it back to the original currency. When arbitrage is present this amount would be higher than the original amount.

Not only are arbitrages rare but when present they disappear very quickly making them hard to find. We can detect them by forming the problem as an optimization problem and using linear programming. This report briefly discusses theory of linear programming, constructing a linear program to detect arbitrage in foreign exchange and pros and cons of this method.

2 Data Description

We examine a year worth of exchange rates from May 21st, 2020 to May 21st, 2021 of 8 currencies: USD, CAD, EUR, AUD, GBP, JPY, CNY, and INR. The data is stored in a dictionary with the keys being the dates and the value being a Pandas DataFrame. For example, the exchange rates on May 21st, 2020 are,

	USD	CAD	EUR	AUD	GBP	JPY	CNY	INR
USD	1.000000	1.391818	0.909091	1.519091	0.817664	107.654545	7.104818	75.504091
CAD	0.718485	1.000000	0.653168	1.091444	0.587479	77.348138	5.104703	54.248530
EUR	1.100000	1.531000	1.000000	1.671000	0.899430	118.420000	7.815300	83.054500
AUD	0.658288	0.916218	0.598444	1.000000	0.538259	70.867744	4.677020	49.703471
GBP	1.222997	1.702189	1.111815	1.857843	1.000000	131.661163	8.689170	92.341261
JPY	0.009289	0.012929	0.008445	0.014111	0.007595	1.000000	0.065996	0.701355
CNY	0.140750	0.195898	0.127954	0.213811	0.115086	15.152329	1.000000	10.627167
INR	0.013244	0.018434	0.012040	0.020119	0.010829	1.425811	0.094098	1.000000

3 Linear Programming

A linear program (LP) is the problem of maximizing or minimizing an affine function subject to a finite number of linear constraints. An affine function is a function $f : \mathrm{R}^n \to \mathrm{R}$ of the form $f(x) = a^T x + \beta$ where a and x are vector and β is a constant. Moreover, the following are linear constraints [1],

$$f(x) \leq \beta \quad \text{or} \quad f(x) \geq \beta \quad \text{or} \quad f(x) = \beta \tag{1}$$

An example of a LP is,

$$\begin{aligned} \max \quad & x_1 \\ \text{subject to} \quad & x_2 + x_3 \leq 3 \\ & x_1 - x_4 \geq 1 \end{aligned} \tag{2}$$

This LP has 4 variables $x_1, x_2, x_3, x_4 \in \mathbb{R}$. The objective function is x_1 and we are trying to maximize it. Moreover, the LP has 2 constraints that must be satisfied.

4 Model

Consider two currencies such as CAD and USD at any particular time. There are 2 exchange rates between them, 1 CAD will buy r_1 USDs and 1 USD will buy r_2 CADs. Observe, an arbitrage opportunity exists if $r_1 r_2 > 1$ since we can simultaneously convert 1 CAD into r_1 USDs and then r_1 USDs into $r_1 r_2 > 1$ CADs. We net $r_1 r_2 - 1$ CADs without any risk [2]. We can go beyond 2 currencies with the help of linear programming. To form the LP, our variables and constants are,

$$\begin{aligned} a_{ij} &:= \text{the exchange rate from currency } i \text{ to currency } j \\ x_{ij} &:= \text{the amount of currency } i \text{ to convert to currency } j \\ y_k &:= \text{the net amount of currency } k \text{ after all transactions} \end{aligned} \tag{3}$$

Let n denote the amount of currencies. The LP for this problem is as follows,

$$\begin{aligned} \max \quad & y_1 \\ \text{subject to} \quad & y_k = \sum_{i=1}^{n} a_{ik} x_{ik} - \sum_{i=1}^{n} x_{kj}, k = 1, \ldots, n \\ & y_1 \leq 1 \\ & x_{ij} \geq 0, i \in \{1, \ldots, n\}, j \in \{1, \ldots, n\}, i \neq j \\ & y_k \geq 0, k \in \{1, \ldots, n\} \end{aligned} \tag{4}$$

were the constraint $y_1 \leq 1$ is needed so the LP is bounded.

Consider this example of made up exchange rates,

	USD	*EUR*	*GBP*	*AUD*	*JPY*
USD	1.00000	0.63900	0.537	1.0835	98.89
EUR	1.564	1.00000	0.843	1.6958	154.773
GBP	1.856	1.186	1.00000	2.014	184.122
AUD	0.9223	0.589	0.496	1.00000	91.263
JPY	0.01011	0.00645	0.05431	0.01095	1.0000

Its corresponding LP is,

$$
\begin{aligned}
\max \quad & y_1 \\
\text{subject to} \quad & y_1 = 1.564x_{21} + 1.856x_{31} + 0.9223x_{41} + 0.01011x_{51} - x_{12} - x_{13} - x_{14} - x_{15} \\
& y_2 = 0.63900x_{12} + 1.186x_{32} + 0.589x_{42} + 0.00645x_{52} - x_{21} - x_{23} - x_{24} - x_{25} \\
& y_3 = 0.537x_{13} + 0.843x_{23} + 0.496x_{43} + 0.05431x_{53} - x_{31} - x_{32} - x_{34} - x_{35} \\
& y_4 = 1.0835x_{14} + 1.6958x_{24} + 2.014x_{34} + 0.01095x_{54} - x_{41} - x_{42} - x_{43} - x_{45} \\
& y_5 = 98.89x_{15} + 154.773x_{25} + 184.122x_{35} + 91.263x_{45} - x_{51} - x_{52} - x_{53} - x_{54} \\
& y_1 \leq 1 \\
& y_k \geq 0, k \in \{1, \ldots, 5\} \\
& x_{ij} \geq 0, i \in \{1, \ldots, 5\}, j \in \{1, \ldots, 5\}, i \neq j
\end{aligned} \tag{5}
$$

Solving the LP we get an optimal value of 1 with optimal variables,

$$
y_1 = 1 \quad x_{12} = 365.90933 \qquad x_{23} = 233.81606 \quad x_{35} = 197.10694 \qquad x_{51} = 36,291.724
$$

All other variables being 0.

This solution leads to a \$1 arbitrage as follows

$$
\begin{aligned}
\text{USD to EUR:} \quad & (365.90933)(0.639) = 233.81606 \\
\text{EUR to GBP:} \quad & (233.81606)(0.843) = 197.10694 \\
\text{GBP to JPY:} \quad & (197.10694)(184.122) = 36,291.724 \\
\text{JPY to USD:} \quad & (36,291.724)(0.01011) = 366.90933
\end{aligned}
$$

Observe, we have a \$1 in arbitrage which is very tight and vanishes when taking in account transaction cost and other fees. Moreover, a large amount of capital is needed to achieve \$1 in arbitrage. Note that the LP was bounded by 1, we can change the upper bound and it has the effect of scalar multiplying the variables x_{ij}.

5 Conclusion

Running the model on the yearly data we found there are no arbitrages which is expected since the exchange rates are for the day and if there was arbitrage it would disappear very quickly. We can overcome this by having real-time data being fed into the model and continuously solving the model. Solving the model should not be computationally hard since the LP can be solved in polynomial time. Attached is the data and code used for this report.

Bibliography

[1] J. Könemann B. Guenin and L. Tunçel. *Linear Programs*, pages 5–6. Cambridge University Press, New York, NY, 2014.

[2] Javier Peña Gérard Cornuéjols and Reha Tütüncü. *Linear Programs*, pages 53–54. Cambridge University Press, New York, NY, 2018.