

CS3414. Homework Project V.

In this assignment you will try to interpolate historic USD/EUR foreign exchange data. The idea is that if your interpolation is clever, and you have managed to capture the key trends, you can make money by *extrapolation* into the future. That is if your interpolating/extrapolating function can predict the future USD/EUR rate, you can make money by either buying USD or selling it at the right time. You will be given historic data for a few past years, and you will use least square fitting to find the best functional dependence of USD/EUR as a function of time. You will then make a (virtual) currency purchase according to your predictions and see if you gain or lose on the transaction.

You CAN work in groups of up to three people. There will be no penalty for being part of a group. The group submits ONE report with ALL names on it, each member will get the same number of points. The group that made most virtual "money" and therefore has developed the best interpolation scheme will receive a REAL prize.

1 Background

As the cost of a currency changes, one can make money by buying it when it is cheap and selling it when it goes up in price (very complicated, isn't it?). On the figure below you see a historical chart of EURO currency measured in US \$ (USD). Simply put, if you buy at time t_1 and sell at time t_2 ,

$$\text{your profit} = [R(t_2) - R(t_1) - C] \times M, \quad (1)$$

where M is the amount of money you risk, $R(t)$ is the exchange rate, C is a small commission that the clearing house takes per transaction. Assume $C = 0.003$ here. Also assume that your $M = 100,000$. Of course, $\text{your profit} < 0$ if $R(t_2) < R(t_1)$. You can still make money, but you will need to sell at t_1 and buy at t_2 . So, all you have to do is predict how $R(t)$ will look like tomorrow, or 10 days later, or 3 month later. Note that if you expect the future value of EUR to be lower, then you want to sell at t_1 and buy at t_2 , in which case:

$$\text{your profit} = [R(t_1) - R(t_2) - C] \times M, \quad (2)$$

2 To Do:

Basically, you need to find a meaningful fit function to the data you have, and use it to "predict" the future. You will place your bets according to your prediction. So, perform

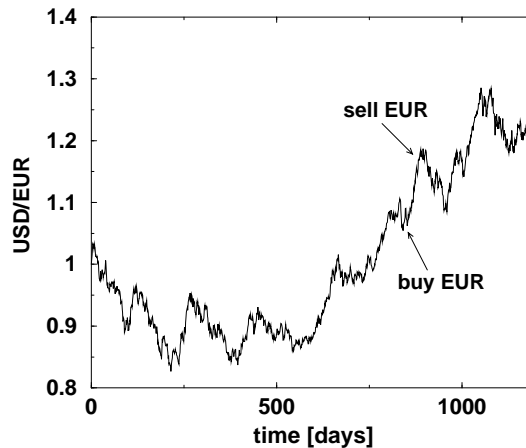


Figure 1: Historic chart of USD/EUR exchange rate. Cost of 1 EUR in USD. If you had bought and sold at points indicated, you would have made money.

either a linear or a non-linear least square fit (approximation) to the data set given.

The data set is given as a two dimensional column $\{ t_k, y_k \}$, just like in Fig.1. I strongly suggest that you plot one of the data sets to see that for yourself. Also, get familiar with the fitting routines first, by generating some simple data sets and using fits to recover the input. Similar to what we did in class. For example, generate data that changes periodically with time, with some small noise, and fit it to a `Sin[]` function. Make sure you understand what you are doing. You must present these results in your printout.

3 Results

Results should be presented as a combined plot of the original data and your fit function. A mathematical equation for your fit function must be presented. The detailed buy/sell instructions should be specified (see below). You should also explain why, in your opinion, the non-linear fit algorithm did not always give you best fit.

Algorithm design/implementation.

There is really not much to do in terms of the code, you may slightly modify the `least.square.FOREX.math` shown in class. The key is to find best function, not to write a code. If you want to see a working code that was used in real life, go <http://people.cs.vt.edu/onufriev/software.html> and look up

the DSR fit code. However, it has way more than you will need for the project. If you really hate Mathematica, you could try and write your own C++ code, but the non-linear least square fitting algorithm is rather complex, so if you absolutely hate Mathematica's implementation, you should use the critical part that is available from reliable sources. You can use GSL, Numerical Recipes or a NetLib source. Best is to stick to Mathematica. Look up `least.square.nonlinear.math` and `least.square.FOREX.math` on our class web-site to see what functions to use in Mathematica. Mathematica's front-end is another useful source for syntax questions. You can read in an external file via `data = ReadList["data", Number, Number]`. Mathematica uses Levenberg-Marquardt method to do least square fits. This is also the default in Mathematica's `NonlinearFit[]`. The method combines the steepest descent with a variation of a multi-dimensional Newton's method for minimization. The details are not important for you here. But note that the method has all of the generic pitfalls of finding a global minimum in many dimensions.

4 The contest.

Along with the best fit graph, functional form, and *detailed reasoning* of how you have arrived at it, you should also make a bet. That is you place all of your money $M = 100,000$ at the time $t_1 = 1195$ in present (this is the last data point available to you, EUR/USD = 1.2070) and predict a time t_2 in the future where you it is best to sell (Or buy. You have to specify whether you sell or buy at t_1). Note that t_1 is fixed, you are not allowed to change it. After the homework has been collected, the "future" will be released, and you will compute how much money you have just made (or lost...), according to Eq. 1 if you your instructions were to buy first or Eq. 2 if you sell first. Of course, you can't change your instructions past the deadline of this project. Bring these results to class once the "future" has been released. **On the first page, near your name, you MUST give a short-hand for your prediction, in bold font. For example: "SELL 1316" or "BUY 1211", meaning "buy at $t_1 = 1195$ and sell at $t_2 = 1316$ " or "sell at $t_1 = 1195$ and buy at $t_2 = 1211$ ".**