1. Choice of epsilon : Support Vector Regression seems to be an interesting choice of algorithm when the goal is to predict the return of a financial asset. While non-linearity, as a known stylized fact of financial returns, is a strong argument for using SVR, the epsilon insensitive cost function can also be viewed from an economic point of view, from which any deviation of the actual returns of less than an epsilon can represent an acceptable loss for the investor. However, the choice of epsilon can be particularly problematic, especially from a computational cost point of view. Since our target variable is centered around zero, it would only make sense to search for an optimal epsilon of a very small value. The issue with that is that the lower epsilon, the higher the precision required to approximate the target variable, hence the more SVs needed to encode that, therefore the higher the computational cost. This is illustrated by the figure below, where we plotted the time (in seconds) to train the SVR (for a train set of 70% of the whole dataset), and by the figure next to it, where we plotted the number of Support Vectors encoded by the algorithm as a function of epsilon. Finally, this is also related to the flateness of our estimations: when epsilon is much larger than the average observation, the algorithm never penalizes any wrong prediction, which in turn leads to no Support Vectors at all, which therefore allows the norm w to be set to zero, as seen on the figure below. All these facts gives motivation on the necessity of correctly tuning epsilon, but the fact that its optimal value should be relatively low can be expensive if implementing Grid Search. While research has shown that implementing non-Fixed and symetrical margin (NASM) worked particularly well when implementing SVR on financial data, we decide to maintain a classical fixed and symmetrical margin (FASM), but we decide to search for a margin that fits around the standard deviation of our target variable, as in our case standard deviation for financial returns are used to capture volatility. We can accept to have a higher margin when volatility is high, and, at the contrary, low margin when the volatility is low. This limitation allows us to reduce the computational cost when seeking for the optimal margin to train our algorithm.
2. Choice of C: (gamma = scale of kernel)
3. Relationship between algorithm and outliers : if outliers fit outside the epsilon-tube, they are supposed to be more penalized, the higher they are. They are even more penalized the less C is. Questions: how does it impact the Support Vectors (values and numbers)?
4. How sensitive is the algorithm to very few samples ? in terms of computational time, the fit time is more than quadratic with the number of samples.
5. Normally, the computational time of SVR is independent of D. Show this by changing the number of D.
6. What impacts has the choice of Kernels?
7. What impacts has the choice of cost function?? According to paper, it One can show that the performance of a SV machine depends significantly on the cost function used [M¨ uller et al., 1997, Smola et al., 1998b]