Master's thesis proposal

Krzysztof Wojdalski

General idea

The general idea is to implement and train reinforcement learning agent on FX market data. I use Temporal Difference, Monte Carlo Control, and Q-learning algorithms. Several tweaks take place e.g. to address exploitation/exploration dilemma.

Reinforcement learning is effective for problems that other, deterministic methods do not fit well , implementation is overly complicated or require too much computation power.

Let's take a simple Tic Tac Toe game for example. The traditional way to address the problem of finding right policy to maximize win/loss ratio is to set up a deterministic policy for 3^9 states (some of them are infeasible, but let's ignore it for now). 3 since we can have 2 different marks or blank space (3 different states for each field). 9 because by default Tic Tac Toe game is played on 3×3 matrix. Solving Tic Tac Toe in this way is error-prone and not generalizing well, i.e. it would require a lot of effort. Instead, it is possible to avoid countless ifs and do nearly as good, with reinforcement learning. It utilizes dynamic programming principles to find such a policy that maximizes the chance of winning, or more correctly - value function.

The hypothesis of reinforcement learning superiority is based on the assumption that agent, if well-tweaked, might see relationships between variables that are not obvious for an individual. In my opinion it is much less efficient or impossible to immune against non-desired outcomes in certain market conditions with a set of condition given a priori.

To sum up, in my master's thesis I would like to test whether or not such methods might be beneficial for entities that deploy them.

The implementation of a reinforcement learning algorithm for Tic Tac Toe game in R:

- Krzysztof Wojdalski - GitHub / Tic Tac Toe

References for algorithms/concepts:

- Q-learning - wiki
- Temporal difference learning - wiki
- SARSA

Longer version

Financial markets have been interested in computer science methods since the 1970s. Although there are ways to gain abnormal, positive returns by following traditional (fundamental) ways of investing, such as buy-and-hold, more sophisticated (in the mathematical/statistical sense) methods gain on popularity. One of the most emerging categories is artificial intelligence-based trading. Such an approach has been employed due to the belief algorithm can be at least as good as human in the decision process.

As of now, the majority of systems described in the literature aim to maximize trading profits or risk-adjusted measures, such as the Sharpe ratio. Many attempts have been made to come up with a consistently profitable system and inspiration has come from different fields ranging from fundamental analysis, econometric modelling of financial markets, to machine learning. A few attempts were successful and those that seemed most promising often could not be used to trade actual markets due to associated practical disadvantages. Among others these included large drawdowns in profits and excessive switching behaviour resulting in high transaction costs. Professional traders have generally regarded those automated systems as risky in

comparison to the returns they were able to deliver by themselves. Even if a trading model was shown to produce an acceptable risk-return profile on historical data there was no guarantee that the system would keep on working in the future. It could cease working precisely at the moment it became unable to adapt to changing market conditions.

My upcoming master's thesis aims to deal with the above problems to obtain a usable, fully automated and intelligent trading system. To accomplish this, a risk management layer and a dynamic optimization layer are added to known (previously mentioned) machine learning algorithms. The middle layer manages risk in an intelligent manner so that it protects past gains and avoids losses by limiting or even shutting down trading activity in times of high uncertainty. The top layer dynamically optimizes the global trading performance in terms of a trader's risk preferences by automatically tuning the system's hyper-parameters.

While the machine learning system is designed to learn from its past trading experiences, the optimization overlay is an attempt to adapt the evolutionary behaviour of the system and its perception of risk to the evolution of the market itself.

This research departs from a similar principle by developing a fully layered system where risk management, automatic parameter tuning and dynamic utility optimization are combined. Merging the machine learning algorithm with the dynamic optimization is adaptive reinforcement learning.

Chapter 1 consists of the introduction to the problem. It outlines the whole concept of the AI-related fields in finance. It brought up historical background of finance and computer sciences, and its interdependency.

Chapter 2 starts with the outline of selected papers from quantitative finance. It includes both classic models, such as CAPM, the gold standard in equity research, and modern ones. The part is descriptive as it regards implicit pros and cons of financial models. The literature review is specifically about algorithmic trading and the methodology of other similar researches, e.g. Sakowski et al. (2013). Second chapter also includes the explanation of machine-learning algorithms underlying (or inspiring) the trading system.

Chapter 3 starts with the detailed explanation of the research. The main hypotheses are as follows:

- Algorithms based on artificial intelligence can be fruitful for investors by outperforming benchmarks in both risk and return;
- Better performance turns out to be true in high-frequency trading and on longer period intervals;
- Algorithms can learn how to spot overreacting on markets and choose the most under/overpriced security by exploiting time series analysis tools.

It contains the description of the methodology - all formulas and steps that directed to final results. It looks at each layer of the trading system. In Subchapter 3.1 the modifications to the standard algorithm are set out and in Subchapter 3.2 and Subchapter 3.3 the risk management and optimization layers are explained.

The used algorithms are based on dynamic optimization approach. Besides value function based on Differential Sharpe Ratio, there will be several indicators, e.g. RSI, which serves as a base for decision taking of the algorithm. The methodology will include transactional costs so that the optimization is going to be implemented in a real-like environment.

The value function will be based on several statistics, such as the Sharpe and the Differential Sharpe Ratio to capture both risk and return. The output of my algorithm will be a set of the agents' actions in the form of $\{-1, 0, 1\}$. Moreover, I will enclose all elements of Reinforcement Learning-model, i.e.:

- environment - a data frame of cumulated returns, and risk measures, such as the Sharpe ratio, MD, MDD, the Sortino ratio and other measures
- states - a vector of environment in time t
- actions - a vector of -1s, 0s, and 1s
- rewards - in this case it is $DiffSharpeRatio_t - DiffSharpeRatio_{t-1}$
- policy - a set of rules for agents. For instance, if EUR/USD gained more than 0.001 between t and $t-1$, then go up.

In the last part of the chapter the performance of the trading system (RL-based agents) is demonstrated and examined against several benchmarks, such as:

- Buy-and-hold strategy what means holding long-position in selected currency pairs.
- Random actions - this part of the algorithm will generate random values in a domain of $\{-1, 0, 1\}$. These values will serve as a position in the underlying pairs. The benchmark will not include any transactional costs as this obvious that this extreme case would have an enormous cumulated transactional cost (position would change in $\frac{2}{3}$ of states).

The final section outlines conclusions. It compares the results with similar works and suggests possible directions to extend the research. It addresses such questions as: * What can be additionally implemented? * What were limitations and what must be done to avoid them in the future?

Data

Datasets used for the purpose of this workpaper are from the following databases:

- Thomson Reuters Tick Database for FX market - including all ticks for Reuters D3000 interbank market. I have the access to the last 3 months.
- Tick database from a vendor aggregating quotations from liquidity providers (TradAir, yet to be discussed)
- As the time interval is uneven, it is needed to fill the gaps with the last known value from the past.

Data is divided into training and test datasets (70/30) so that the conclusions are based on actual agent performance on data it is not familiar with.

Formula for the Differential Sharpe Ratio

- The differential Sharpe ratio - this measure is a dynamic extension of the Sharpe ratio. By using the indicator, it can be possible to capture a marginal impact of return at time t on the Sharpe Ratio. The procedure of computing it starts with the following two formulas:
-
$$A_n = \frac{1}{n} R_n + \frac{n-1}{n} A_{n-1}$$
$$B_n = \frac{1}{n} R_n^2 + \frac{n-1}{n} B_{n-1}$$

  At $t = 0$ both values equal to 0. They serve as the base for calculating the actual measure - an exponentially moving Sharpe ratio on η time scale.

$$S_t = \frac{A_t}{K_\eta \sqrt{B_t - A_t^2}}$$

  where:

- $A_t = \eta R_t + (1 - \eta) A_{t_1}$
- $B_t = \eta R_t^2 + (1 - \eta) B_{t_1}$
- $K_\eta = \frac{1 - \frac{\eta}{2}}{1 - \eta}$

Use of the differential Sharpe ratio in algorithmic systems is highly desirable due to the following facts:

- Recursive updating - it is not needed to recompute the mean and standard deviation of returns every time the measured value is evaluated. Formula for A_t (B_t) enables to very straightforward calculation of the exponential moving Sharpe ratio, just by updating for R_t (R_t^2)
- Efficient on-line optimization - the way the formula is provided directs to very fast computation of the whole statistic with just updating the most recent values
- Interpretability - the differential Sharpe ratio can be easily explained, i.e. it measures how the most recent return affect the Sharpe ratio (risk and reward).