Lifetime Portfolio Selection:
Using Machine Learning

Gordon Irlam
(gordoni@gordoni.com)

July 14, 2020

Abstract

The first half of this paper provides a primer on machine learning and relates it to problems in financial planning. Machine learning proves capable of handling real world complications and details which classical financial planning approaches cannot handle; e.g. taxes, reversion to the mean, and time varying yield curves. The second half of this paper provides a case study of the use of reinforcement learning for making asset allocation, annuitization, and consumption decisions in retirement planning. Machine learning typically delivers within a few percent of the theoretical optimal solution on highly abstract problems. Machine learning is found to outperform other approaches for more complex financial planning problems whose optimal solution is not known. The value of SPIAs and inflation-indexed bonds is highlighted by the machine learning approach.

Introduction

Financial planning problems

Financial planning is hard. You have to make decisions today, in a highly complex, highly uncertain environment, with the full impact of those decisions not known until decades later. This is true of asset allocation, consumption planning, pension planning, tax planning, long-term care planning, annuitization decisions, and reverse mortgages. Worse, each piece of the puzzle affects every other piece.

The classical approaches

Modern Portfolio Theory represents a rich body of theory that at first glance appears to allow you to make asset allocation decisions. Unfortunately, applying Modern Portfolio Theory to financial planning has several related problems. First the answer depends on how much risk is appropriate for you, and that varies in an unknown way, depending on both the size of your financial assets, and your expected future income. Second, Modern Portfolio Theory is unable to handle bonds, or any other asset class that exhibits mean reversion. Finally, Modern Portfolio Theory is based on optimizing over a single time period, not the multiple time periods commonly required in financial planning.

Samuelson (1969) and Merton's (1969) work on what is known as Merton's portfolio problem provides a solution to asset allocation and consumption planning problems over multiple time periods. Merton's portfolio problem is a mathematical problem with a precise mathematical solution. Unfortunately it is unable to handle the effects of labor or guaranteed income, which is typically the largest asset held by most individuals.

Bellman's work on dynamic programming performed in the 1950's is also worthy of mention (Bellman 2003). Dynamic programming is a numerical technique that delivers the theoretical optimal result, but can do so only for simple scenarios. Dynamic programming is able to handle guaranteed income. But trying to extend it to realistically handle bonds, stocks with mean reversion, or partial annuitization decisions is computationally extremely challenging.

And as far as incorporating the effect of taxes, all of the classical approaches fall a long way short.

Rules of thumb

Possibly because of the inadequacy of the classical approaches to financial planning problems, financial planning often falls back on rules of thumb: follow a glide path while working, employ a fixed asset allocation in retirement, use actuarial withdrawal rules, annuitize by age 80, and so on. Monte Carlo simulation is helpful in exploring the range of possible outcomes, but Monte Carlo software does not optimize strategies, and for the most part there is little by way of optimization in financial planning software. The rules of thumb though helpful only provide a partial solution, and as we shall see later often significantly under-perform a more sophisticated approach.

Research Question

The classical approaches to developing optimal financial plans were all developed in the 1950's and 1960's. Might we be able to do better today by employing modern machine learning / artificial intelligence techniques? The answer appears to be a resounding yes.

Literature Review

Early work on using machine learning for asset allocation alone includes Neuneier (1996 and 1998) and Sato and Kobayashi (2000). Weissensteiner (2009) extended the problem to include determining the optimal consumption amount. However Weissensteiner used neuron self-organizing maps rather than artificial neural networks, Q-Learning rather than a more modern reinforcement learning algorithm, and did not incorporate guaranteed income or taxes.

In a review of machine learning and financial planning Mulvey (2017) found "the area of finance has been relatively immune to the ML technology" indicating a lack of uptake of the early research by the financial community. More recently Kolm and Ritter (2019) provided an overview of the relationship between reinforcement learning and multiple time period financial decision making opening up additional avenues for research.

This paper applies more modern reinforcement learning techniques, as exemplified by Proximal Policy Optimization (Schulman et al. 2017), to the asset allocation and consumption problem including the effects of guaranteed income and taxes.

Machine learning

Machine learning is comprised of three main sub-fields:

- supervised learning – learning to recognize patterns given a labeled training data set; e.g. credit scoring given a collection of training data on individuals including a label that indicates whether they have defaulted on their credit
- unsupervised learning – learning patterns in data that lacks pre-existing labels; e.g. recommendation systems for movies, music, or shopping in which individuals that appear to have similar tastes are grouped together but no pre-existing labels for these tastes are present
- reinforcement learning – learning to take actions in an environment so as to maximize a reward signal; e.g. a computer playing a game of Go or StarCraft

Of the three, reinforcement learning (Sutton and Barto 2018) appears to have the most relevance to financial planning. It is also unfortunately the most complicated.

Reinforcement learning concepts

As mentioned, reinforcement learning involves taking actions in an environment so as to maximize some reward signal. The environment in reinforcement learning is something that can be observed, and interacted with through a discrete series of actions that make up an episode. After each action the environment generates a numerical reward, along with a second signal indicating whether the episode is done and a new episode needs to be started.

In the financial planning context the environment will be some sort of financial model of how a client's income and assets might evolve over time, typically, until they die.

An observation might consist of a subset of the following parameters: marital status, the client's age and health, current wage income, current or expected Social Security, pensions, and other guaranteed income, the size and possibly makeup of their investment portfolio, the value of their other assets including their home, liabilities, future expected expenses, when they plan to retire, the availability of a 401(k) scheme, the degree of any bequest motive, an assessment of whether stocks are fairly valued, the current interest rate yield curve, the inflation rate, and how the client feels about variability and uncertainty in consumption, i.e. their risk aversion.

An action might consist of a subset of the following: consuming a particular dollar amount in the current time period, an asset allocation to adopt, the type and duration to use for bonds, the amount of SPIAs to purchase, the amount to invest or withdraw from tax free or tax deferred accounts, long-term care purchases, and whether to obtain a reverse mortgage.

The reward will be the utility derived from consuming the amount specified by the most recent action. Utility is a concept from economics. It is the well-being or satisfaction derived from consumption. If the utility of consuming $50,000 was 1, then the utility of consuming $100,000 might be 1.5. This is because satisfaction saturates at higher consumption levels. For retirement planning, where we are typically not concerned with maximizing pre-retirement well-being, the pre-retirement rewards will all be zero.

A policy is a mapping from observations to actions. The goal of reinforcement learning is to find a policy that attempts to maximize the expected sum of all the rewards from each episode. In the financial planning context the goal of reinforcement learning is to find a policy that maximizes well-being over the lifecycle.

Both the values making up the observations and the actions in reinforcement learning may be made up of discrete or continuous values. A yes/no decision to buy a SPIA or Long-Term Care insurance policy are a good example of discrete values, while portfolio size, consumption, and the optimal SPIA purchase amount are examples of a continuous values. In financial planning the policy function primarily maps from continuous observations to continuous actions. Problems like financial planning where the action space is continuous are referred to as continuous control problems.

Reinforcement learning algorithms

An algorithm is a detailed step-by-step procedure for solving a problem that can be implemented using a computer. A difficulty with mastering reinforcement learning is there are many different algorithms, with more being developed every year, and they defy an orderly categorization scheme.

Many reinforcement learning algorithms follow an actor-critic model in which there is a policy function (the actor) which maps from an observation to action to take, and a value function (the critic) which maps from observation to a number, the reward-to-go, that specifies how well the policy function is expected to perform over the remainder of the episode. Both the policy function and the value function are described by a large set of weights. In the financial planning context the policy function might map from age and portfolio size to the recommended asset allocation and consumption, while the value function maps from age and portfolio size to a dollars per year value representing how well the policy function performs.

Perceptrons are highly stylized mathematical models of neurons. A single perceptron takes a series of numerical inputs, i_1, i_2, i_3, …, multiplies each by the weight for that input, w_1, w_2, w_3, …, sums the products and passes the result though an "activation" function, f, producing the output value:

$$o = f(w_1 * i_1 + w_2 * i_2 + w_3 * i_3 + \ldots)$$

The role of the activation function is to apply a non-linearity to the result. It may be as simple as:

$$f(x) = \max(0, x)$$

or as complex as the hyperbolic tangent function.

A feed-forward network is one in which information flows in a single direction; there is no feed-back. Historically multi-layer feed-forward networks of perceptrons forming an artificial neural network have been used for the policy and value functions. The first layer is the input layer that simply comprises the observation. Each subsequent layer consists of a set of perceptrons, each of which take the results of the prior layer as inputs, apply their own weights, and pass the result through the activation function, to produce a set of outputs. The final layer, the output layer, often lacks an activation function.

For continuous control problems the layers are typically fully connected, meaning that each perceptron in a layer receives inputs from every one of the outputs from the layer below. The layers of the network excluding the input and output layers are termed the hidden layers. An artificial neural network with 2 or more hidden layers is termed a deep neural network. For playing games and image processing tasks anywhere from 20 to 200 hidden layers might be usual. For financial planning because of the smoothness of the surfaces being modeled, a far more modest 2 hidden layers are typically appropriate. For the artificial neural networks constructed in the second half of this paper both layers are comprised of 256 perceptrons.

A key concept in reinforcement learning is the notion of advantage. If at some point the expected value of future rewards is 10, but after performing a timestep you receive a reward of 4 and the expected value of future rewards is now 8, the advantage would be 4 + 8 – 10, or 2. More formally, advantage is the difference between the observed reward at the end of some timestep plus the value of the value function, and the value of the value function prior to that timestep. A large advantage signals a large divergence between the current value function and the true value function for the current policy function.

Both the policy and value functions are iteratively refined. The weights are initially assigned random values. The policy function is then used over multiple episodes to generate a large batch of experiences. For each timestep in the batch the advantage is computed. These advantage values then get used to update the policy graph weights to maximize the return of the policy. At the same time the distance between the observed rewards over the remainder of an episode and the value function’s predicted reward-to-go is typically used to update the weights of the value function to maximize its accuracy.

The weights are updated by performing “hill climbing” on the function being optimized. Hill climbing is an optimization technique that attempt to find a maximum value by looking at the slope of the surface being optimized, figuring out the direction which points the most uphill, taking a small step in that direction, then looking at the slope again, and taking another small step. Over many iterations the weights are adjusted, the policy function gradually converges towards the optimal policy, and the value function converges to the expected reward-to-go for that policy function.

An important idea in reinforcement learning is the concept of exploration versus exploitation. It turns out it is necessary to add a degree of noise into the training procedure to ensure the policy algorithm doesn’t get stuck exploiting a local maximum, but explores other nearby actions.

Another concept is that of bias. Reinforcement learning may produce results that are biased in one direction or another. This can be the result of the credit assignment problem. It isn’t always possible for the reinforcement learning algorithm to know that poor results being experienced now are the result of overly aggressive asset allocation decisions many time periods earlier. As a result it may produce recommended asset allocations that are 0 to 20% more aggressive than warranted. When necessary this bias is measured and manually corrected.

Reinforcement learning in practice

In general machine learning requires a lot of training data. The availability of data can be a limiting factor in training machine learning models. For financial planning though it is possible to simulate a financial model and generate as much data as required on the fly.

Training a reinforcement algorithm takes a lot of computer time. To train the models used in the next section required training for a grand total of almost 7 billion simulated years on 9 four core servers for 3 days.

Once trained, the inference process is very fast. Inference involves looking up the recommended asset allocation and consumption policy given an observation. It takes around 10 milliseconds to look up a financial plan for a client, and the vast majority of this time involves marshaling the data into a format the policy function expects. Application of the policy function itself takes less than a millisecond.

It often isn’t enough though to tell a client what they should do. Often it is important to give the client a sense of what to expect in the future. For this Monte Carlo simulation of the recommendations may be necessary. This is a slower process that can take anywhere from 10 seconds to 10 minutes. The performance bottleneck here is not the speed of the policy function, but the speed of the financial model being simulated.

The results from reinforcement learning are frequently very good, but they are not optimal. Limits on the quality of the results are set by the size of the neural network model in terms of the number of weights, and by the amount of training performed.

Applying machine learning to retirement planning

In this section a financial model for retirement planning is developed, validated against a classical approach, refined in ways the classical approach can’t handle, and then compared against other approaches for both a retiree and over the lifecycle.

Results were produced by first using reinforcement learning to train a set of 7 generic models, each capable of handling a broad range of financial scenarios divided up based upon risk aversion, retirement status, and whether SPIA purchases are allowed. The performance of the models was then evaluated by performing Monte Carlo simulation of specific scenarios.

Mortality

For training the pre-retirement model, an initial age of 20 was used. For training the retired model, an initial age of 50 was used. This encompasses training with almost the full range of age possibilities because the financial model increases the age after each reinforcement learning action.

The Social Security Administration AS 120 female cohort life table was used to determine the probability of being alive in each year up until age 121. The financial model performed simulations up to the maximum age for each episode and the utility rewards were weighted by the probability of being alive. This allows the capture of the full range of mortality experiences in a single episode.

Poor or good health was simulated by increasing or reducing the age by several years to produce an effective age before looking up the mortality associated with this age in the life table. Health plays a key role in determining the suitability of the purchase of SPIAs.

Since actions are based on observed remaining life expectancy, not age, there is little need for the use of a separate male life table, or life tables for different cohorts when training.

During evaluation a female with a life expectancy 3 years longer than predicted by the life table was used. This reflects the good health of typical financial planning clients.

Retirement and guaranteed income model

Retirement is assumed to occur at age 67, and $20,000 per year of Social Security received during retirement. During training a broad range of retirement age and income possibilities was considered.

Relative Risk Aversion (RRA)

Risk aversion specifies the degree to which the client is averse to variability and uncertainty in future consumption. A relative risk aversion of γ implies a utility, U, given by:

$$U = C^{1-\gamma} / (1 - \gamma)$$

where C is the consumption amount. For every 1% increase in consumption there is a γ% reduction in marginal utility, that is utility per dollar of incremental consumption.

In this paper 3 RRA values will be considered: 1.5, 3, and 6. The higher the RRA value the more risk averse the client is.

Unlike risk aversion in Modern Portfolio Theory, which is totally dependent on the client’s wealth and fixed income expectations which vary over time, relative risk aversion is independent of wealth and portfolio size, and only very weakly, if at all, dependent on consumption level. This greatly simplifies determining how much risk is appropriate to take. The client simply has to pick a single number. Despite this, determining the most appropriate risk aversion to use is still a difficult decision.

Certainty Equivalent (CE)

Sometimes papers comparing financial outcomes overwhelm with a sea of numbers: worst case, best case, percentiles, average, and median values. Certainty equivalents are a technique for boiling down this sea of numbers to a single number against which meaningful comparisons can be made.

The certainty equivalent of a consumption strategy is the constant fixed consumption that has the same expected utility as the uncertain variable consumption of the strategy.

Certainty equivalents are highly useful and will be used for all comparisons made in this paper.

Validation results

Before diving in to the rich financial model that reinforcement learning is capable of optimizing, it is worth pausing to compare the results of reinforcement learning to dynamic programming.

For these scenarios stock returns are log-normally distributed with a 6.5% arithmetic mean real return and 17.4% volatility (geometric return 5.1%), and bond returns are also log-normally distributed with a 1.0% arithmetic real return and 11.0% volatility. Except for the lower anticipated return on bonds these values were taken from the global historical record reported in the 2019 Credit Suisse year book (Dimson et al. 2019).

In these scenarios the results of a generically trained reinforcement learning model trained against the simple stock and bond model are compared against the optimal results produced using dynamic programming. At each timestep the reinforcement learning observation is the life expectancy remaining, and investment wealth as a fraction of investment wealth plus the present value of future guaranteed income. The action consists of the amount of wealth to consume and the asset allocation to use. The client’s age is 67, they are retired, receive $20,000 per year of Social Security, and a number of different possible initial portfolio sizes are considered. The reinforcement learning model is trained with both taxable and tax-free assets and income. At evaluation time all assets and income are tax free. The results are shown in Table 1. The standard error of reinforcement learning CE measurement is around 0.3%.

RRA	Initial portfolio	Reinforcement learning CE	Optimal CE	Relative performance
1.5	$200,000	$32,615	$32,728	99.7%
1.5	$500,000	$49,802	$49,867	99.9%
1.5	$1,000,000	$77,346	$77,350	100.0%
1.5	$2,000,000	$131,096	$131,058	100.0%
3	$200,000	$31,503	$31,508	100.0%
3	$500,000	$45,976	$46,023	99.9%
3	$1,000,000	$68,039	$68,246	99.7%
3	$2,000,000	$109,305	$110,318	99.1%
6	$200,000	$29,880	$29,987	99.6%
6	$500,000	$40,968	$41,316	99.2%
6	$1,000,000	$57,286	$58,040	98.7%
6	$2,000,000	$87,060	$88,794	98.0%

Table 1. Comparison of the results of reinforcement learning to the optimal results computed using dynamic programming.

In every case the results of reinforcement learning are within a few percent of the optimal solution. This is highly encouraging. Reinforcement learning knows nothing of log-normally distributed returns. It is therefore hoped that by adopting richer stock and bond models reinforcement learning might equally do well at optimizing them.

Stock model

Stock returns adhere to the global historical record reported in the Credit Suisse year book.

Shiller (2015) argues convincingly that stocks at times exceed or fall below their rational valuations. As a result, for every 10% stocks are over-valued or under-valued a 1% reduction or increase in expected return is applied. Stock volatility follows a GJR-GARCH model meaning that volatility has an element of predictability. The GJR-GARCH model is calibrated to the historical returns over the period 1970-2018, and the residuals from this calibration exercise are used to provide the stochastic returns. This means stock returns display fat tails; i.e. skew and kurtosis.

The parameters used here are only a rough estimate of the precise situation. Nonetheless, by using these estimates we are likely to do better than by ignoring them.

Observations of the environment include the stock price to fair price and the trailing volatility. However, to be fair, a 15% standard deviation noise factor is added to the observed stock price to reflect the difficulty in assessing whether stocks are truly fairly valued. When evaluating performance stocks are initially fairly valued.

Bond model

Bond returns correspond to the returns expected from holding bonds in a bond fund.

For the bond model a Hull-White bond model is employed. This means there is a yield curve and the yield curve varies over time in a non-linear fashion depending on the short interest rate. The Hull-White yield curve is calibrated to the average Treasury yield curve over the period 2005 – 2018. The initial short real rate is 1.0%, and the initial inflation rate is 1.6%. Nominal rates are constructed from the real yield curve and the inflation expectations curve.

The Hull-White model means bonds as an asset class are less risky than they first appear. Returns exhibit a degree of mean reversion. When rates go up, producing poor returns, it is more likely that they will fall in the future, producing good returns, and vice versa.

Asset class standard errors

With little more than 100 years of high quality returns data there is considerable uncertainty as to the long run mean returns of the asset classes. The Credit Suisse yearbook reports a 1.6% standard error in the reported mean stock return. To recognize this each episode is given a different expected mean return with a 1.6% standard deviation. On top of this expected mean return the stock volatility model is applied.

Similar adjustments are applied to the average long run short rate of bonds and the inflation rate.

SPIA model

SPIAs are priced using the Society of Actuaries IAM Basic Table and a Money’s Worth Ratio (MWR) of 98% against Treasury bonds after state guarantee association taxes. Annuity Experience Report contract age adjustments are applied to reflect the fact that a recent SPIA purchaser is likely to be in better health than someone of the same age that purchased a SPIA 10 years ago.

An MWR of 98% against Treasuries is reasonable and reflects the competitive SPIA market in the U.S.. When priced against corporate bonds, as might be held by an insurance company, SPIAs have a lower MWR. The bond model currently only includes Treasury bonds, not corporate bonds, hence the use of Treasury bonds in the SPIA pricing model.

SPIAs may only be purchased from retirement through age 90, with a minimum purchase amount equal to 10% of total assets, and a 2% annual increase to compensate for the effects of inflation.

Because the bond model is time varying, the price of SPIAs has to be recomputed at every time period.

Tax model

Both income and wealth are divided into tax free, tax deferred, and taxable amounts. Individual allocations to each asset class are maintained for taxable assets. For simplicity average cost is used as the cost basis method for changes in share holdings.

The tax model is currently based on the U.S. income tax code for 2020. Many simplifications are obviously made, but the level of detail is still substantial. A progressive tax rate structure is used with the standard deduction, special taxation rules applied to Social Security payments, capital gains tax rates, capital loss carry forwards, and the net investment income tax. 401(k) and IRA contributions are allowed and the IRA catch-up amount is available for older workers.

State and local taxes on the other hand are greatly simplified. Based on national averages, a state, local, and property tax rate of 11% of income above the standard deduction is applied. Improving much past this would require diving into the details of how taxes are assessed in every state, a formidable task.

Comparison in retirement

To assess the performance of reinforcement learning Monte Carlo simulations of a 67 year old retiree in good health with a $1,000,000 investment portfolio were performed. To make the problem challenging, the retiree’s relative risk aversion was 6. The retiree was able to invest in nominal Treasury bonds and stocks, but not SPIAs. Table 2 compares reinforcement learning to a range of alternative withdrawal schemes with constant asset allocations. All of these alternative schemes have parameters, such as the initial consumption amount, asset allocation, and assumed rate of return. For each scheme a trial and error search was performed to find the optimal parameter values. This took several days. Many of these schemes were proposed without consideration of guaranteed income or taxes. This needed to be corrected. Consumption amounts were computed by applying the scheme to an estimate of the after tax portfolio size and then adding an estimate of the after tax guaranteed income. For the fixed asset allocations, a nominal bond duration of 20 years was used, reflecting the long duration typically selected by reinforcement learning.

Withdrawal scheme	Asset allocation	CE
Percent rule	Fixed	$44,330
Guyton’s Rule 2	Fixed	$45,003
Target Percentage Adjustment, life expectancy 35	Fixed	$47,708
PMT, life expectancy 35	Fixed	$49,763
Guyton Klinger, life expectancy 35	Fixed	$50,650
PMT, dynamic life expectancy	Fixed	$51,945
Extended RMD	Fixed	$54,086
Blinded reinforcement learning	Dynamic	$55,498
Reinforcement learning	Dynamic	$56,690

Table 2. Comparison of strategies for a retiree, RRA=6.

The reader is referred to Bengen (1994), Guyton (2004), Zolt (2013), and Guyton and Klinger (2006) for a description of some of these rules. PMT uses the payout amount required to deplete the portfolio over the remaining life expectancy assuming some fixed rate of return. Extended RMD bases withdrawals on the IRS required minimum distribution table extended back to age 67. It uses the old 2002 RMD table, not the revised table currently proposed by the IRS to take effect in 2021. Blinded reinforcement learning is reinforcement learning blinded to observation of the stock price, volatility, and real interest rate.

Reinforcement learning outperforms all of the schemes considered.

For this scenario Extended RMD performs very well. But that is just this scenario. For a different initial portfolio size, a different health level, or a different risk aversion, it may not perform as well. And that is the issue. Testing out different schemes and parameter values to find the one that performs the best is time consuming. Reinforcement learning on the other hand appears able to deliver a near optimal strategy in milliseconds.

The small difference between blinded reinforcement learning and reinforcement learning in which it is possible to take advantage of the stock price, volatility, and real interest rate may possibly point to the difficulty of successfully performing tactical asset allocation.

Financial instruments comparison

Reinforcement learning can be used to shed light on the appropriate financial instruments to use in a retirement portfolio. Table 3 shows the results for the previously described retiree scenario. Treasury bonds plus 1% is intended as a generous estimate of the return for corporate bonds.

Instruments	Reinforcement learning CE
Stocks, Treasury bonds	$56,690
Stocks, Treasury bonds + 1%	$58,046
Stocks, Inflation-indexed Treasury bonds	$58,526
Stocks, Inflation-indexed Treasury bonds, SPIAs	$63,527

Table 3. Certainty equivalents produced by reinforcement learning with different financial instruments for a retiree, RRA=6.

There is a strong case to be made for the use of SPIAs in retirement assuming no bequest motive. There is also a weaker case for the use of inflation-indexed bonds. It is also worth noting reinforcement learning selected long duration bonds: 16 years for nominal bonds, and 20-22 years for inflation index bonds.

Comparison over the lifecycle

Consider a 30 year old female in good health and initially with no assets. She earns $80,000 per year before tax, and consumes $50,000 a year after tax. She plans to retire at age 67 and receive an inflation adjusted $20,000 per year in Social Security. She invests in stocks, inflation-indexed Treasury bonds, and once retired SPIAs. Table 4 presents the results of using several different strategies assuming an RRA of 6. The glide path is 90% stocks until 25 years before retirement, sloping down to 40% stocks at retirement and beyond. Optimal values of all strategy parameters were chosen.

Asset allocation	Withdrawal scheme	Annuitization scheme	CE
Fixed	Extended RMD	100% at fixed age	$61,644
Glide path	Extended RMD	100% at fixed age	$62,663
Reinforcement learning	Dynamic	Dynamic partial annuitization	$67,032

Table 4. Comparison of strategies over the lifecycle, RRA=6.

Once again reinforcement learning delivered results significantly in excess of the other strategies.

Discussion

Reinforcement learning is a rapidly evolving field. Already it appears capable of delivering results that are superior to other approaches. And it is likely that the algorithms in use today are only going to be improved upon in the future.

Despite the promise of reinforcement learning, one of the problems with reinforcement learning today is the variability in the recommendations. Near the optimal asset allocation, a 10% change in asset allocation might only result in a 0.5% change in CE. Reinforcement learning does well when the difference in performance between different alternatives is large, but less well when the difference is small. Consequently different neural networks trained on the same data may produce recommended asset allocations that differ by 10% or even 20%. The truth is this difference barely matters to outcomes, but the lack of consistency may appear unprofessional to a client if they are told they should be at 60% stocks one year and 80% the next.

Another issue is the black box nature of the approach. There aren’t any tools available to answer questions like “Why do you recommend I annuitize 50% of my assets at age 70?”. You simply have to trust the results of the machine learning approach.

There is a danger of some people believing the results too much simply because they were produced using “artificial intelligence”. Yes, machine learning does a good job of solving the asset allocation and consumption problem, but it does so with a very broad brush, that provides an outline of the solution, but not the details. It knows nothing of tax lots, specific id, and deciding which lots to sell. Similarly the allocation of assets across taxable, tax-deferred, and tax-free accounts is currently performed using hard coded rules of thumb (stocks gravitate toward taxable; bonds gravitate towards tax-deferred). A savvy financial planner can add value by filling in the details, and deciding when rules of thumb should be broken.

Finally there is the issue of heuristics, versus optimal solutions. How confident can we be in a tool that appears to do very well in practice, but lacks any sort of ironclad guarantees. This is particularly important for scenarios that lie at extremes that might not have been encountered frequently, or at all when training the model. Here, some suspicion of the results is warranted.

Availability

The website www.aiplanner.com provides access to reinforcement learning trained financial planning models. It is quick to set up and run. The website takes as inputs high level data about life expectancy, assets, liabilities, and retirement goals, and produces consumption, asset allocation, and annuitization recommendations, along with the results of a Monte Carlo simulation of the recommendations. AIPlanner is described in more detail in Irlam (2018 and 2020).

Conclusion

Reinforcement learning appears to be the first fundamentally new approach to financial planning in 50 years. It is capable of generating very high quality financial plans for complicated scenarios involving guaranteed income, rich stock and bond models, assets, liabilities, and taxes, for pre and post retirement scenarios. No other tool or technique comes close to being able to do this. Often the plans come within a few percent of the theoretical optimal result. Reinforcement learning outperforms all other considered alternative approaches. In the past financial planning was constrained by the complexity of the financial planning problem. This no longer appears to be the case. The biggest issue in applying reinforcement learning today might be in determining the accuracy of the financial model used compared to the real world.

Acknowledgments

The author is grateful to Phil DeMuth, William Selden, and Joe Tomlinson for feedback on early versions of this paper.

References

Bellman, Richard. 2003. Dynamic Programming. Dover Publications.

Bengen, William P. 1994. “Determining Withdrawal Rates Using Historical Data.” Journal of Financial Planning 7 (4): 171–180.

Dimson, Elroy, Paul Marsh, and Mike Staunton. 2019. Credit Suisse Global Investment Returns Yearbook 2019. Zurich: Credit Suisse.

Guyton, Jonathan T. 2004. “Decision Rules and Portfolio Management for Retirees: Is the “Safe” Initial Withdrawal Rate Too Safe?” Journal of Financial Planning 17 (10): 54–62.

Guyton, Jonathan T. and William J. Klinger. 2006. “Decision Rules and Maximum Initial Withdrawal Rates.” Journal of Financial Planning 19 (3): 48.

Irlam, Gordon. 2018. “Financial Planning via Deep Reinforcement Learning AI.” SSRN 3201703.

Irlam, Gordon. 2020. “Multi Scenario Financial Planning via Deep Reinforcement Learning AI.” SSRN 3516480.

Kolm, Petter N., and Gordon Ritter. 2019. "Modern Perspectives on Reinforcement Learning in Finance." The Journal of Machine Learning in Finance 1 (1).

Merton, Robert C. 1969. “Lifetime Portfolio Selection Under Uncertainty: The Continuous-time Case.” The Review of Economics and Statistics 51 (3): 247–257.

Mulvey, John M. 2017. "Machine learning and financial planning." IEEE Potentials 36 (6): 8–13.

Neuneier, Ralph. 1996. "Optimal asset allocation using adaptive dynamic programming." Advances in Neural Information Processing Systems, pp. 952–958.

Neuneier, Ralph. 1998. "Enhancing Q-learning for optimal asset allocation." Advances in Neural Information Processing Systems, pp. 936–942.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347.

Samuelson, Paul A. 1969. “Lifetime Portfolio Selection By Dynamic Stochastic Programming.” The Review of Economics and Statistics 51 (3): 239–246.

Sato, Makoto, and Shigenobu Kobayashi. 2000. "Variance-penalized reinforcement learning for risk-averse asset allocation." International Conference on Intelligent Data Engineering and Automated Learning, pp. 244-249. Springer, Berlin, Heidelberg.

Shiller, Robert J. 2015. Irrational Exuberance: Revised and Expanded Third Edition. Princeton: Princeton University Press.

Sutton, Richard S. and Andrew G. Barto. 2018. Reinforcement Learning: An Introduction. Cambridge: MIT Press.

Weissensteiner, Alex. 2009. "A Q-Learning Approach to Derive Optimal Consumption and Investment Strategies." IEEE transactions on Neural Networks 20 (8): 1234–1243.

Zolt, David M. 2013. “Achieving a Higher Safe Withdrawal Rate With the Target Percentage Adjustment.” Journal of Financial Planning 26 (1): 51–59.