QuantAwards

The Quantitative Finance Competition

2021 winners

- 1st prize: Reinforcement Learning vs 1/N and Mean-Variance Optimization In The Portfolio Allocation Problem' - Matus Jan Lavko - Utrecht University, Economics and Business Economics
- 2nd prize: Selection of securities for collateralized debt obligations contracts: an optimization approach applied to European SME loan portfolios - Matthieu Jonard - UC Louvain, CEMS Master in International Management
- 3rd prize: Who Are the Socially Responsible Investors? A Machine Learning Approach - Sul Kim - Tilburg University, Master in Finance

Background information

More than a century after the seminal work of Louis Bachelier, the quantitative approach to financial markets has become omnipresent.

Nowadays, many investment outfits specialize in research, development, and implementation of systematic trading strategies, while other active asset managers have added quantitative strategies to their business lines. Individual clients may also now delegate management of their portfolios to robo-advisors.

These are just a few manifestations of the changing landscape, and we believe that quantitative portfolio management will become ever more important because of the discipline offered by the scientific approach and full automation of the investment process.

The QuantAwards competition offers students a unique opportunity to showcase their creativity and their understanding of this highly timely subject.

Reinforcement Learning vs 1/N and Mean-Variance Optimization In The Portfolio Allocation Problem

Abstract—**In this paper, I test for the out of sample trade performance of state of the art model-free reinforcement learning agents to compare them against the performance of an equally weighted portfolio and typical mean-variance optimization benchmarks. Motivated by the recent advances in this field and successful applications of the Reinforcement Learning paradigm in continuous state-action space, I develop a study within different contexts. By dividing the European and U.S. broad market index constituents into factor data sets using a Carhart four-factor model, the models face different scenarios defined by these factor environments. Consequently, the RL approach is empirically evaluated based on a number of measures and probabilistic assessments. Training the models only on price data and features constructed from the prices, the performance of the RL approach yields better risk-adjusted as well as probabilistic portfolio as the Mean-Variance specifications, but ties with the equally weighted portfolio. The models are partially able to uncover the nonlinear structure of the SDF and the key drivers of the performance and suggest that these models are useful in practical application as part of a larger trend of AI and big data applications in finance.**

I. Introduction

Advances in computing and engineering bring about intriguing challenges in the realm of nonlinear modeling problems that are unfeasible to solve analytically and require more complex methods. The advent of reinforcement learning and stochastic control has been deployed with great success onto the board games problems, notably TD-Gammon (Tesau & Tesau, 1995) or later ventures of the IBM DeepMind, where the models achieve a high level of winning rates against other programs as well as human professional players[1] (Silver et al., 2016). Growing digitization and automation in the asset management industry provides a natural ground for experiments with stochastic control because of suitable mathematical specifications of the problem. Numerous attempts to apply reinforcement learning have been conducted with promising results especially using direct reinforcement or inverse reinforcement methods (Moody et al., 1998), but until recently problems associated with training brittleness and high level of dimensionality did not allow for deep networks to be utilized during the process as the learning process proved often to be unstable.

In the model-free representation of the environment – in contrast to the model-based – there are two main approaches that allow for the use of deep learning, often combined in parallel. In Q-learning, an experience buffer storing transition states has improved training significantly (Charpentier et al., 2020). In Policy Optimization methods, various parallelism techniques as in Schulman et al., 2017 or other tweaks to the architecture[2] have been presented in order to address the issues described earlier. These algorithms that are on the forefront of research have proven to be useful for benchmark problems and given their promising results this provides a motivation to test them on a domain problem in finance – the Portfolio Allocation problem.

A. *Motivation*

Portfolio construction is an important problem in finance, with an objective to construct an optimally performing portfolio. Traditionally, it has been proposed that this construction consists in two-steps, with a parametric approach to the estimation of moments and optimizing over a grid of feasible portfolios (Markowitz, 1952). It has been established that due to high dimensionality and short sample histories, the estimation is unstable (Merton, 1980), and various methods have been proposed to stabilize this process such as robust specifications with "uncertainty structures" (Goldfarb & Iyengar, 2003), Bayesian approaches (Aguilar & West, 2000) or incorporating equilibrium returns and views about asset returns (Black & Litterman, 1992). These have proven to have ad hoc advantages but often fail due to estimation errors as described in DeMiguel et al., 2009.

The problem of nonlinear effects and interaction effects present in the structure of the cross-section of the returns result in ambiguity and a "zoo" of factors attempting to explain the difference in risk premia (Harvey et al., 2016). My work is hence tangent to the problem of asset pricing in the field of machine learning applications in the domain space. Recovering the Stochastic Discount Factor[3] for asset pricing models using nonlinear effects (Gu et al., 2020) or regularization approaches (Kozak et al., 2020), and further review of deep learning using an array of firm-specific and macro features under a no-arbitrage machine learning problem specification (L. Chen et al., 2019) show that the SDF is well approximated by linear

[1] Such as the IBM Deep Blue playing Garry Kasparov, a world champion in 1996-1997. The symbolic win of Deep Blue in 1997 was a landmark victory for the computer over a champion player.

[2] Gradient clipping, Monte Carlo sampling or delayed networks are among many proposed ideas as will be described later.

[3] SDF is defined as the transformation of a portfolio lying on the capital market line such that the expected excess return is zero. For more general definition see Danthine and Donaldson, 2014.

factor models, although there are robustness advantages to using machine learning methods.

Although my goal is not retrieving the SDF explicitly, an application of this approach can be used for actively managing exposures and risks, found implicitly by the model and incorporated into the decision making within the RL framework with various constraints. This can provide benefits to algorithmic trading strategies complementing traditional tools. A similar effort in Cong et al., 2020 based on a complex model-based multi-armed bandit RL system strives to make RL more interpretable for investing through considering logits as scoring metrics for allocation. In my paper, a different approach is taken based on a coarser price history used for training and no use of firm-specific or macro variables in order to conduct a benchmark study similar to the method described in DeMiguel et al., 2009.

B. Problem Statement

Consider an environment $\mathcal{E}$ consisting of N securities that are contingent on a partially observed continuous state space[4] $\mathcal{S}$ of which $\mathbf{w}_{n,t+1} \subset \mathcal{S}$ are the weights of a portfolio that constitute actions $\mathbf{a}_t$ with state $\mathbf{s}_t := \mathbf{p}_t$, where $\mathbf{p}_t$ is the respective price vector at time t.

A portfolio in time t is a unique set of actions $\mathbf{a}_{t-1}$ and is determined by a reward $r(\mathbf{a}_t, \mathbf{s}_t)$ that we wish to maximize in $[t_0, T)$, where for each t a trade can take place such that:

$$\mathbf{a}_{t-1} = \mathbf{w}_t = \begin{pmatrix} w_{1,t} \\ \vdots \\ w_{N,t} \end{pmatrix} \ , \ \sum_{i=1}^{N} w_{i,t} = 1 \ \ , w_{i,t} \in \mathbb{R}^+ \qquad (1)$$

Given a portfolio $\mathbf{w}_t$, state $\mathbf{s}_t$ we postulate a simple arithmetic return in the form of $r_t = \frac{s_t}{s_{t-1}} - 1$, and derive the simple portfolio return and variance for a vector valued random variable:

$$\boldsymbol{\mu} := \mathbb{E}[\mathbf{r}] \approx \mathbb{E}\begin{pmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_N \end{pmatrix} = \begin{pmatrix} \frac{1}{T}\sum_{t=1}^{T} r_{1,t} \\ \vdots \\ \frac{1}{T}\sum_{t=1}^{T} r_{N,t} \end{pmatrix} \qquad (2)$$

$$\sigma^2 := \mathbb{E}[(\mathbf{r} - \mathbb{E}[\mathbf{r}])^2] \ \ , \mathrm{Var}[\mathbf{r}_i] = \frac{1}{T-1}\sum_{i=1}^{T}(r_t - \mu_r)^2, \quad (3)$$

obtaining the return and volatility of a portfolio as:

$$\mu_p = \mathbf{w}^T \boldsymbol{\mu} \ \ , \sigma_p^2 = \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w},$$

where $\boldsymbol{\Sigma}$ is an element-wise expansion of the left expression in 3.

Having defined these estimated quantities[5], we would like to maximize the reward function across the discrete grid $t_0, \ldots, T$ with distances of length h. We choose this reward to be the Sharpe Ratio as in 4:

$$\mathbf{w} = \underset{\mathbf{w} \in G}{\operatorname{argmax}} \sqrt{h} \frac{\mathbb{E}[\mathbf{r_{t,t+h}}]}{\sqrt{\mathrm{Var}[\mathbf{r}_{t,t+h}]}}, \qquad (4)$$

where we satisfy the set of affine constraints G as defined in Meucci, 2005.

This problem is typically attacked with the use of quadratic programming, but we can exploit this formulation and rewrite the problem without the loss of generality such that it is broadly applicable to the reinforcement learning setting:

$$\max_{\Theta} \sum_{t=1}^{T} \mathbb{E}[\gamma r(\mathbf{a}_t, \mathbf{s}_t)], \qquad (5)$$

where γ stands for a time-discounting factor and Θ the parametric expression of the nonlinear models used to estimate the value or policy functions depending on the architecture of the model used.

II. Methodology

A. Data

I collect raw daily open high low close price and volume data for all equities in the S&P 500 index and its European counterpart assembled by Bloomberg – Bloomberg 500 index (Bloomberg, 2020). In addition I collect a sector dataset according to S&P sector classification, assembled from 9 SPDR sector exchange traded funds (SPDR, n.d.) and two iShares ETF's namely the Telecommunications (iShares, n.d.-b) and Real Estate (iShares, n.d.-a). Due to computational constraints and a hypothesis of a change in the market regime, the data set is restricted to start from 15th of October 2008 until 1st of January 2021 for a total of 2974 observations.[6] In order to prevent survivorship bias and ensure completeness of the sample, securities that do not have at least 2 years before t_0 are purged, resulting in a dataset of 479 securities listed in Europe and 487 U.S. listed stocks. On this set of securities the features for the training of the RL agents are calculated as described in the Appendix. For estimating the factor model I use factors constructed by Bloomberg, aggregating by monthly return on these factors that are constructed using the methodology in Fama and French, 1992. These are the suitably defined risk premia across the section of the assets in my factor model. In addition, a risk-free proxy is used for computing performance metrics and excess returns as the 10-year rate on U.S. Treasury bills.

B. Training Scenarios

Training on the whole index is computationally very expensive for such a large set of securities and is not of interest since the resulting portfolios would contain a large number of stocks. To divide the data sets into clusters I take an approach

[4] Assuming that a particular observation of state $\mathbf{s_t}$ does not describe the state of the environment as modelled by a multivariate stochastic process.

[5] Note that there are many options one can take in order to pursue less variance in the estimates as outlined in Elton et al., 2006.

[6] Note that this includes the estimation period for the sample covariance matrix which is used as a feature during the training, effectively pushing the training by 252 trading days starting on the 2nd of February 2009

based on a linear time-series factor model that is specified as follows:

$$\mathbf{E}[R_{t,i}] - R_t^f = \alpha_{t,i} + \sum_{j=1}^{n_f} \underbrace{\beta_i(\mathbf{E}[F_{t,j}] - R_t^f)}_{\text{systematic risk}} + \underbrace{\epsilon_{t,i}}_{\text{intrinsic risk}}, \tag{6}$$

where it is assumed that intrinsic risks can be hedged such that $\mathbf{E}[\epsilon_{t,i}] = 0$ for a model with j number of factors across i assets such that the no-arbitrage condition holds.[7] For the purpose of the experiment, this rather strict assumption can be relaxed as it is not an exercise in retrieving the SDF, but it is good practice to keep it in mind given the fact that it affects the form of our training data sets.

Following Carhart, 1997, I postulate the existence of a four factor asset pricing model, where a filtration $\mathcal{F}\colon F_j \in \{MKT_t, SMB_t, HML_t, MOM_t\}$ is the set of risk premia describing the investment universe $\mathcal{E}$.

Assuming that the Generalized Gauss-Markov theorem holds true, we can retrieve $\hat{\beta}_{i,j}$ using a least-squares estimate (Kempthorne, 2013).[8] Given the possibility of substitution effects among the factors I employ a shrinkage method that has the advantages in reducing estimation errors out of sample as in Kozak et al., 2020. Although in such a low-dimensional model this could reduce performance of the SDF as a pricing kernel, this approach is taken in order to improve the stability of the estimates since the parameters will be later used for sorting. On a span of 36 months I estimate a linear model using the L_1 and L_2 penalties and solve the optimization problem in the unconstrained Lagrangian form.

After obtaining the estimates, these data sets are constructed using quantile sorting of the coefficients $\hat{\beta}_j$. This β-quantile is defined as:

$$Q_{\boldsymbol{\beta}}(p) := \inf\{\beta \in \mathbb{R}\colon p \leq F(x)\},$$

where $F_{\boldsymbol{\beta}}(X) = \mathbb{P}(\beta \leq x)$ is the empirical CDF obtained from the factor model.[9] I summarize the resulting sorted data sets in Table I together with their corresponding quantiles.

Dataset	Quantile Range	N
Growth 50	$\{Q_{\boldsymbol{\beta}_{\text{HML}}}(0.1)\}$	50
Momentum 50	$\{Q_{\boldsymbol{\beta}_{\text{MOM}}}(0.9)\}$	50
Size 50	$\{Q_{\boldsymbol{\beta}_{\text{SMB}}}(0.1)\}$	50
Growth + Size 100	$\{Q_{\boldsymbol{\beta}_{\text{HML}}}(0.1)\} \cup \{Q_{\boldsymbol{\beta}_{\text{SMB}}}(0.1)\}$	100
Momentum + Size 100	$\{Q_{\boldsymbol{\beta}_{\text{MOM}}}(0.9)\} \cup \{Q_{\boldsymbol{\beta}_{\text{SMB}}}(0.1)\}$	100
Sector ETF	N/A	11

TABLE I: Beta quantile-sorted data sets

[7]This lemma implies an existence of an SDF which is the main object of interest in the asset pricing literature and is the building block of no-arbitrage pricing models (Danthine & Donaldson, 2014).

[8]Note that under the normality assumption on the distribution of residuals it can be proven that this is also the MLE estimator. See chapter 14 in Rice, 2006 for proof.

[9]Note that the definition of the quantile does not require any smoothness or continuity properties to be required of $F(x)$.

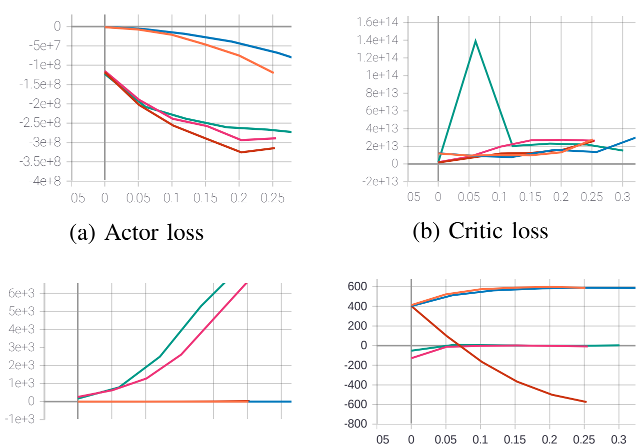
I obtain these sorted portfolios for both the S&P 500 and BE500 indices, resulting in 10 quantile sorted data sets plus the sector ETF data set. This provides for means to test the algorithms on both American and European equities with a single-factor and multi-factor approach, where the idea is to test if the model is able to capture hedging opportunities within different factors – or sectors in the sector ETF data set.

C. Model Training

Training neural networks is often a challenging task given the specifics of stochastic gradient descent optimization techniques. It is also quite computationally expensive which is why I have decided to train the models using Google's cloud service Colab Pro that allowed me to parallelize training on a GPU unit Tesla p100-PCIE-16GB. For the models themselves I have used a high-level package that is built on PyTorch (Liu et al., 2020) and provided an interface to use the Stable Baselines library (Hill et al., 2018) that provides stable implementations of all state of the art algorithms based on OpenAI infrastructure.

A softer grid search has been employed on all of the models in order to search for optimal hyper-parameters. This has been done in order to ensure that a stable configuration of the model is deployed on the data given the difference in the underlying data processes of our data sets.

The model architectures used in the benchmarking study are the Actor Critic (A2C), Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), Soft Actor Critic (SAC), and Twin Delayed DDPG (TD3). These algorithms allow for continuous state-space and involve tweaks to the Q-learning and Policy Gradient paradigm in one form another [10].



(a) Actor loss (b) Critic loss

(c) Entropy coefficient (d) Entropy loss

Fig. 1: Sample SAC grid search log

Obtaining training statistics for all of the model runs allows for examination of the relevant metrics and selection of the

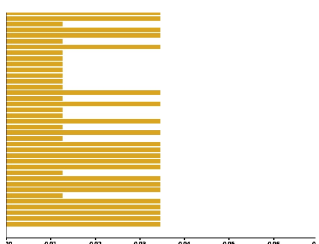
[10]The descriptions of the algorithms together with the source code and the full paper can be accessed at https://github.com/matus-jan-lavko/ReinforcementLearning-vs-EW

most stable model. The log in Figure 1 presents such a screen for a search over SAC models for the SPX Growth 50 data set. Since it is desirable that the networks learn the Q-function – or other learned function – in a stable fashion, we would like to minimize the critic and actor loss such that the curve is smooth. Critic loss is a standard MSE loss and we want it to converge towards a particular quantity, whereas the actor loss is a negated version of the Critic loss (Lapan, 2018, ch. 12) and we would like it to smoothly decrease. Similarly we would like the entropy loss – or any other endogenous loss function specific to the model – to be stabilized and decrease in an orderly fashion. From Figure 1 we see that the best candidate is the model drawn in red, as other candidate models such as the one drawn in pink suffer from stability issues. We would therefore pick this model to be our globally optimal model within our soft search space and use the parameters associated with it in the testing. After the best model is picked, an out of sample test is conducted by rolling the weights vector forward together with the other benchmark models, applying a transaction cost penalty. The resulting sample terminal portfolios of the U.S. Growth 50 data set are illustrated in 2.

(a) Terminal weights A2C

(b) Terminal weights DDPG

(c) Terminal weights PPO



(d) Terminal weights TD3

Fig. 2: Sample SAC grid search log

III. Results

1) Single-Factor: Looking at the results from the S&P 500 single-factor data sets in Table II we see that all of the algorithms except A2C have outperformed the equi-weight portfolio on a risk-adjusted basis in the Growth 50 data set. None of them, however, were able to beat this benchmark on the Momentum and Size data sets. Comparing the performance to the mean-variance approaches, we see that these portfolios consistently beat the equal risk contribution as well as the minimum variance portfolio in the case of SAC and DDPG. Interestingly, none of the algorithms beat the factor constrained portfolio in terms of performance.

	Growth 50				Momentum 50				Size 50			
	r^{ann}	SR	Max DD	$skew(r)$	r^{ann}	SR	Max DD	$skew(r)$	r^{ann}	SR	Max DD	$skew(r)$
A2C	37.1%	1.34	-32.8%	-0.35	25.9%	0.95	-35.5%	-0.75	2.54%	0.03	-57.8%	-0.73
DDPG	36.4%	**1.30**	-34.0%	-0.48	24.4%	0.94	-32.5%	-0.80	3.8%	0.05	-59.9%	-0.59
PPO	35.0%	**1.27**	-34.6%	-0.53	26.2%	0.98	-33.5%	-0.77	1.0%	-0.01	-59.6%	-0.81
SAC	37.6%	**1.40**	-32.3%	-0.52	26.6%	1.00	-32.6%	-0.77	4.2%	0.07	-57.3%	-0.69
TD3	38.3%	**1.41**	-32.%	-0.37	26.0%	0.98	-33.8%	-0.66	2.3%	0.02	-57.9%	-0.48
EW	36.8%	1.36	-33.1%	-0.45	27.4%	1.04	-33.5%	-0.76	5.2%	0.09	-57.5%	-0.54
MV	25.2%	1.13	-22.3%	-0.06	22.4%	0.97	-28.6%	-1.04	3.1%	0.05	-41.7%	-0.86
ER	27.8%	1.19	-26.7%	-0.59	19.3%	0.84	-29.6%	-0.94	-2.3%	-0.14	-44.5%	-0.94
FC	53.1%	1.67	-28.4%	-0.45	43.8%	1.46	-31.3%	-0.61	38.9%	1.04	-41.5%	0.10
MSR					46.2%	1.70	-30.4%	-0.91				

TABLE II: S&P 500 Single Factor Datasets Result

In terms of skewness, the RL agents are good at reducing left-tail risk uniformly in the growth and momentum factor, but struggle with the Size data set. Although all of them beat the mean-variance benchmarks in the size factor, negative skewness remains larger than the equi-weight portfolio. There are no large deviations in terms of drawdowns from the equi-weight portfolio, although generally the RL models deflate the drawdowns slighly. This is true particularly for the SAC that is the only model that is able to decrease drawdown across all of the three factor data sets.

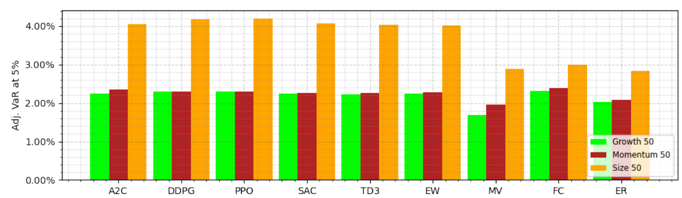


Fig. 3: S&P 500 Cornish-Fisher Adjusted Value at Risk

We can see from Figure 3 that the 5% VaR is very similar for the equi-weight benchmark and the RL models, where only the SAC and TD3 perform better in the momentum and growth factors but the problems with the Size data set remains. In terms of the mean-variance approaches, these both feature a lower VaR than RL for the size factor. Minimum variance and equal risk parity also outperform on the growth and momentum factors but we see that all of the RL models outperform the factor constrained specification in the momentum factor. For A2C, SAC an TD3, better performance is noted for the Growth 50 data set with a slightly lower VaR than the benchmark.

In the European equities, the RL algorithms seemed to be more successful at beating the equi-weight in the performance measures. All of them reported a slightly better performance on the Growth 50 dataset with SAC beating the EW Sharpe by 11%. In the momentum and growth factors TD3 was less successful, but SAC and DDPG still managed to cross the line. All of the algorithms outperformed the minimum variance, but fell short when compared to the linear factor constrained portfolio. An interesting anomaly happens with the momentum factor, where the equal parity outperforms not only the RL models but also all of the mean-variance benchmarks and the MSR which often results in a portfolio that is largely overfit and very concentrated[11].

	Growth 50				Momentum 50				Size 50			
	r^{ann}	SR	Max DD	$skew(r)$	r^{ann}	SR	Max DD	$skew(r)$	r^{ann}	SR	Max DD	$skew(r)$
A2C	17.6%	**0.90**	-26.6%	-1.95	5.3%	0.11	-51.0%	-0.31	-4.9%	**-0.21**	-46.6%	-0.49
DDPG	18.2%	**0.92**	-27.2%	-1.88	7.8%	**0.19**	-51.0%	-0.37	-5.1%	**-0.21**	-46.8%	-0.47
PPO	17.5%	**0.85**	-27.6%	-2.03	6.6%	**0.15**	-51.4%	-0.33	-5.9%	-0.24	-47.4%	-0.40
SAC	19.5%	**0.97**	-27.1%	-1.78	9.5%	**0.25**	-51.1%	-0.51	-5.2%	**-0.22**	-45.2%	-0.41
TD3	18.0%	**0.87**	-27.7%	-1.99	4.8%	0.09	-51.5%	-0.27	-6.2%	-0.25	-46.6%	-0.63
EW	17.0%	0.84	-27.4%	-1.96	5.1%	0.11	-51.3%	-0.25	-5.6%	-0.23	-46.6%	-0.58
MV	8.2%	0.44	-21.7%	-1.37	-2.4%	-0.16	-44.1%	-1.07	-8.2%	-0.47	-35.1%	-1.28
ER	6.1%	0.30	-21.8%	-1.56	0.1%	0.96	-41.2%	-0.72	13.3%	-0.49	-36.5%	-1.59
FC	42.2%	1.68	-28.9%	-0.92	24.8%	0.93	-38.5%	-0.97	15.1%	0.44	-45.7%	-0.4
MSR	42.24%	1.95	-23.5%	-0.44	25.1%	-0.06	-38.5%	-1.03	-8.6%	0.39	-45.7%	-0.39

TABLE III: Bloomberg 500 Single Factor Datasets Result

Assessing the distributional properties of the result, RL seems to increase the negative skew in the growth dataset, where all of the models outperform the benchmark, suggesting that they are taking more risky bets. In the momentum and size dataset, however, they reduce negative skewness and beat the mean-variance benchmark as well as the EW benchmark in the size factor. Although the drawdown remains to be around the same level as the EW benchmark, this shows that there could be benefits from reducing the left-tail risk. In contrast to the U.S. equities, RL algorithms performed better in terms of risk measures what is presented in Figure 4 as the difference between the VaR of RL algorithms and the FC mean-variance optimization. In the growth factor, all of the RL algorithms perform better than the factor model. In the size factor, this result is not as pronounced, although notably the SAC algorithm outperforms the FC and the EW. Momentum factor shows that the mean-variance outperforms on VaR measure, although SAC beats the EW again by a small margin.

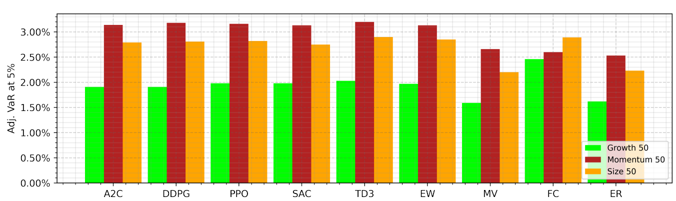


Fig. 4: Bloomberg 500 Cornish-Fisher Adjusted Value at Risk

2) Multi-Factor: In the two factor data sets, where I combine growth and momentum factors with the size factor in order to learn a hedging strategy, RL performs better on the U.S securities than the European-listed stocks. Hedging growth with size, SAC and PPO stands out with better SR_α. In terms of risk, SAC is able to reduce the drawdown slighly, albeit skewness remains to be about the same level. These two models also beat MV, but are unable to beat the linear factor as well as the equal parity portfolios. Notably the ER portfolio is able to reduce the DD of the test strategy by more than 13% from the EW benchmark.

In the momentum combined with size, TD3 seems to do a good job at reducing the left-tail risk as well as outperforming the EW benchmark. Similarly, SAC seems to perform just as good as EW and PPO and DDPG also perform considerably well with DDPG increasing the SR_α by 21%, while still reducing drawdowns slightly. With regards to the mean-variance benchmarks, DDPG outperforms MV and ER but falls short of the linear model again.

	Growth + Size 100				Momentum + Size 100			
	r^{ann}	SR	Max DD	$skew(r^{\text{ann}})$	r^{ann}	SR	Max DD	$skew(r^{\text{ann}})$
A2C	19.8%	0.55	-45.9%	-0.75	17.6%	0.51	-44.5%	-0.95
DDPG	20.1%	0.58	-43.6%	-0.86	20.6%	**0.63**	-42.1%	-0.89
PPO	22.0%	**0.63**	-44.2%	-0.83	18.8%	**0.55**	-44.5%	-0.87
SAC	21.8%	**0.66**	-42.5%	-0.82	17.4%	0.51	-43.6%	-0.88
TD3	21.1%	0.61	-45.0%	-0.80	18.6%	**0.54**	-43.3%	-0.73
EW	21.3%	0.62	-43.9%	-0.83	17.9%	0.52	-43.9%	-0.88
MV	16.1%	0.58	-32.6%	-0.56	13.1%	0.41	-39.0%	-1.00
ER	16.3%	0.68	-29.4%	-0.77	14.4%	0.60	-29.8%	-1.05
FC	23.0%	0.70	-41.2%	-0.52	21.6%	0.65	-40.4%	-0.76
MSR					20.6%	0.62	-40.4%	-0.77

TABLE IV: S&P 500 Multi-Factor Datasets Result

Looking at the results tested on the European stocks, the RL agents perform rather poorly with only TD3 and A2C outperforming the EW benchmark on growth and size and momentum and size in that order. However, it does so with problems to learn a good hedging strategy, as we see in the case of TD3 resulting in inflated negative skew. These results show that EW portfolio is superior to mean-variance approaches too as only the factor model beats the EW in both of the cases. In contrast to this, the RL approach works better for all of the models except for PPO – where after inspecting the trading pattern, transaction costs probably cut from the performance considerably. Both the MV and ER approaches perform worse with ER, failing quite hard in the momentum size dataset. In general, the RL approach underperforms but not by a wide margin, and seems to have quite stable results not deviating from the EW portfolio to a large extent.

	Growth + Size 100				Momentum + Size 100			
	r^{ann}	SR	Max DD	$skew(r^{\text{ann}})$	r^{ann}	SR	MAX DD	$skew(r^{\text{ann}})$
A2C	4.2%	0.11	-38.5%	-1.20	1.2%	**-0.01**	-47.1%	-0.50
DDPG	4.2%	0.11	-37.0%	-1.26	-0.9%	-0.08	-47.1%	-0.38
PPO	2.0%	0.02	-39.5%	-1.64	-1.0%	-0.08	-47.3%	-0.60
SAC	4.6%	0.12	-37.8%	-1.25	-0.9%	-0.08	-48.8%	-0.58
TD3	5.7%	**0.17**	-37.3%	-1.38	-0.3%	-0.06	-47.4%	-0.41
EW	5.0%	0.14	-37.4%	-1.27	0.9%	-0.02	-47.2%	-0.48
MV	1.2%	-0.01	-33.5%	-1.31	-3.9%	-0.2	-42.9%	-0.78
ER	2.6%	0.07	-23.7%	-1.64	-6.4%	-0.39	-36.0%	-1.48
FC	13.6%	0.46	-39.0%	-1.48	3.4%	0.06	-46.9%	-0.67
MSR	11.5%	0.38	-38.6%	-1.49	3.2%	0.06	-46.9%	-0.67

TABLE V: Bloomberg 500 Multi-Factor Datasets Result

In the sector ETF data set I test for the ability for the models to hedge across the sectors, as there is a different and broader set of exposures for the model to learn. The results show that the simple A2C has been the most successful to learn these paterns, with all of the other models having a relatively good performance as well. DDPG and SAC have been able to reduce the DD and negative skewness the most, suggesting that they were able to capture some of the pricing structure of these complex[12] ETF's. Minimum Variance and Equal Risk deliver lower risk-adjusted returns, but again we see that the factor constrained strategy is superior. Achieving a Sharpe close to

[11] Since this is a result of a greedy search over the efficient frontier it often results in a small number of positions being very overweight due to unstable moments estimates.

[12] Complex from the standpoint of an asset pricing equation as there are many risk premia defining an ETF.

the MSR portfolio, it is able to reduce the skew by almost a half while pushing the drawdowns down by 5%.

	Sector ETF			
	r^{ann}	SR	Max DD	$skew(r)$
A2C	16.5%	**0.58**	-36.1%	-0.78
DDPG	12.9%	0.44	-36.3%	-0.66
PPO	12.5%	0.42	-37.6%	-0.92
SAC	13.1%	0.46	-35.3%	-0.73
TD3	14.8%	**0.50**	-37.0%	-0.77
EW	13.7%	0.47	-36.9%	-0.77
MV	8.6%	0.33	-27.7%	-0.41
ER	8.1%	0.26	-34.4%	-0.70
FC	25.1%	0.87	-31.2%	-0.39
MSR	25.1%	0.88	-31.6%	-0.68

TABLE VI: Sector ETF Dataset Result

IV. Inference and Robustness

In order to look at the out of sample stability of the results, we would like to run many trials on different samples and ideally obtain some sort of a bootstrap result, or arrive at a more robust result. This is due to the the results being representative of just one instance of a sequence of random variables of prices. Because of the complexity of training and optimizing RL models, however, such a Monte Carlo experiment is hard to conduct. Hence, in addition to the monthly rolling test, a weekly, quarterly and semi-annual windows have been tested and checked for large divergences that would suggest a high turnover of the strategy rendering the model useless for practical purposes. Referring to Figure 5, we can see that most of the rankings of the algorithms in terms of performance propagate through these windows. Notably, the PPO algorithm seems to deviate more across the windows with more unstable profile due to over-trading as confirmed by the animations I produced for the rebalancing process in the tests. This has been the case with all of the data sets[13], which is interesting due to the PPO's construction and conservative updates mechanism.

Another issue, that is often times overlooked when such rolling tests are conducted is the large variance of the Sharpe Ratio estimator. It is important to realize that all of the reported parameters such as the sample statistical moments are estimators and they have the same distributional properties of a parameter. Therefore, we need a probabilistic approach to assess the relative performance of the models. Given the non-normal features of the observed distribution of returns we need a way to account for the third and fourth moment as they can lead to a large inflation of the sample Sharpe Ratio due to the fact that any observed Sharpe Ratio can be expressed as a combination of two gaussians with different parameters (Harvey et al., 2016, p. 6). This fact can be used to prove that the parameter follows a normal distribution and therefore we can form a test statistic in the form of:

$$\hat{\text{PSR}}(SR_\alpha) = z\left\{\frac{(\hat{SR} - SR_\alpha)\sqrt{T-1}}{\sqrt{1 - \hat{\gamma}_3\hat{SR} + \frac{\hat{\gamma}_4 - 1}{4}SR_\alpha}}\right\}, \quad (7)$$

[13]Even if the ranking changed slightly, a clear winner and a clear loser algorithm has emerged. This suggests using a voting ensemble of the models.

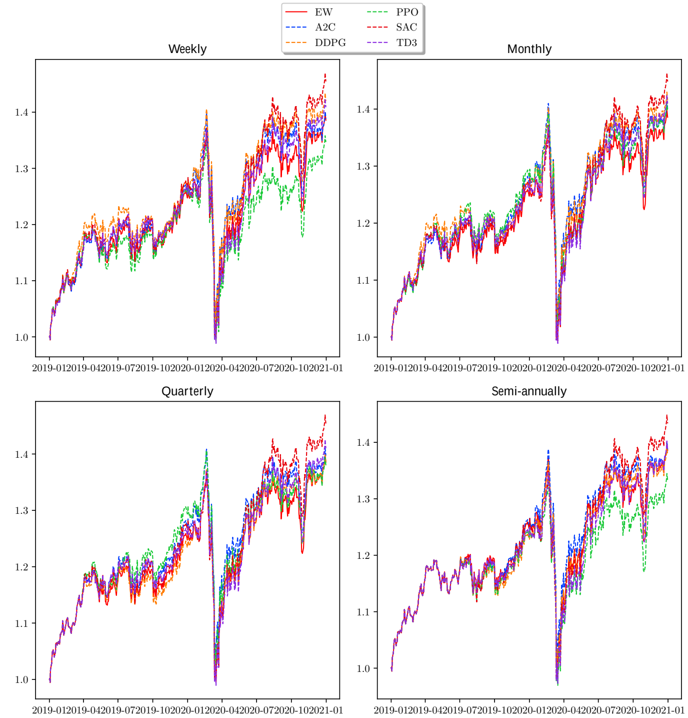


Fig. 5: BE500 Growth 50 Rolling Tests Cumulative Return

where $\hat{SR}$ is the estimate, SR_α is the reference SR, $\hat{\gamma}_3$, $\hat{\gamma}_4$ are the skewness and kurtosis estimates and $z\{\cdot\}$ is the Standard Normal cdf. We call this the Probabilistic Sharpe Ratio (Harvey et al., 2016, p. 9), and essentially we can interpret it as the probability that the observed Sharpe ratio is higher than the reference. This statistic is a function of T as well, so it requires larger returns in order to establish statistical significance.

Using Equation 7, we can construct a hypothesis test of the form:

$$H_0: \quad SR_{test} = \omega$$
$$H_1: \quad SR_{test} > \omega,$$

with a rejection threshold of $1 - P(\omega \leq \hat{SR}_{test}) > \alpha$, for a statistical significance level α. This framework can be used for deflating a reported Sharpe ratio, but I am rather interested in comparing the PSR on a relative level. For example from left Figure in 6 we can see that setting $\omega = 0$, the so-called skill-less level, all of the algorithms together with the benchmark pass the 95% significance level and with this level of confidence we can conclude that the population $SR_{test} > 0$. In addition, we can see that based on this probabilistic assessment, SAC and TD3 have a larger probability of rejecting the null, which is desirable. The Figure on the right suggests that if we run this test on an array of different possible SR, the majority of the algorithms fall short of $\alpha = 10\%$ somewhere between 0 and 0.05. Such an inverted sigmoid is present in all of the data sets, albeit with different cutoffs for statistical significance.

To compare the models, I decided to use a relative percentage marginal for a given test where $\omega = 0$. Note that we have established that the function mapping the $P(SR_{test} > \omega|H_0) \rightarrow SR^*$ does not yield a linear relationship, and therefore it is hard to compare the magnitude of the marginals across the data sets as these functions feature different domain sets. It is, however, a good proxy for seeing how the models compare against the benchmark.

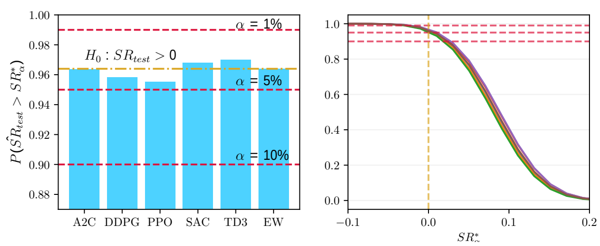


Fig. 6: PSR test on SPX Growth 50

Dataset	A2C	DDPG	PPO	SAC	TD3
SPX Growth 50	-0.02	-0.55	-0.88	**0.40**	**0.59**
SPX Momentum 50	-1.82	-2.01	-1.20	**0.02**	-1.10
SPX Size 50	-3.42	-2.28	-5.71	-1.12	-3.99
SPX Growth + Size 100	-2.66	-1.52	**0.37**	**1.43**	-0.36
SPX Momentum + Size 100	-0.45	**4.28**	**1.23**	-0.42	**0.88**
BE500 Growth 50	**1.49**	**1.97**	**0.19**	**3.18**	**0.74**
BE500 Momentum 50	**0.01**	**4.46**	**2.24**	**7.69**	-1.13
BE500 Size 50	**1.10**	**1.10**	-0.53	**0.56**	-1.09
BE500 Growth + Size 100	-1.67	-1.67	-6.78	-1.11	**1.65**
BE500 Momentum + Size 100	**0.57**	**3.43**	-3.44	-3.42	-2.29
Sector ETF	**4.54**	-1.31	-2.29	-0.43	**1.3**

TABLE VII: PSR pct. marginal over EW for $H_0 : SR_{test} > 0$

The results show in Figure 6 suggest, that the algorithms perform better on the growth and momentum factors as evident from the positive marginals. In the case of the European equities, most of the algorithms except for TD3 in the momentum data set outperformed the EW benchmark. Generally, they seemed to struggle more with the U.S stocks. The only algorithm that performed relatively well on U.S as well as European single factor data was the SAC. Notably, for the multi-factor datasets the simpler A2C and DDPG have performed better than the more complex architectures such as PPO, SAC and TD3. This suggests that different architectures have advantages in learning in different systems.

V. Conclusion

In this paper, I evaluate and benchmark state of the art Model-Free Reinforcement Learning model architectures against the Occam's Razor approach of equi-weight portfolio and the traditional mean-variance benchmarks. Evaluating the performance of these models on the European and American equities within a range of different contexts such as the single and multi-factor data sets, a complementary approach to portfolio management is proposed. This is done without the need to estimate the problematic parameters such as expected returns or the introduction of specific constraints to the SDF. Notably, it has been achieved using a sparse information set consisting only of price data and features transformed thereof.

The main conclusions of my work can be summarized in four points. Firstly, I conclude that the model-free architectures have outperformed the Mean-Variance approaches as demonstrated by the results and the probabilistic assessment of the return series. The resulting portfolios were more stable as a consequence of the models successfully capturing a part of the nonlinear structure in the SDF. This ability to capture the mispricings has not been, however, as successful as the inclusion of risk-premia in the construction of the conditional mean-variance portfolio. The second conclusion ipso facto implies that the factor model has outperformed the RL approach in agreement with the findings of L. Chen et al., 2019. Third, I note that the performance against the $\frac{1}{N}$ portfolio is largely tied and it is inconclusive to say that these algorithms outperform this benchmark in this setting. Although notably in some contexts such as the European Growth and Momentum data sets it has performed very well, in others, such as the Size data sets in both U.S. and EU equities, it fell short. The most consistent model architecture has turned out to be the SAC which hints at the inclusion of entropy within the model-free setting to be an important part of learning such nonlinear structures in a limited data setting as it introduces a lot of exploration into the framework. Fourth, I conclude that there are potential hedging benefits in using the RL approach as I have demonstrated that the models are successful at reducing left-tail risks out of the sample. This is certainly subject to the construction of a more complex decision system that would potentially make use of an ensemble model as different architectures performed well in different subsets of the data.

This paper is complementary to the current literature applying Machine Learning and Deep Learning in the Reinforcement Learning context and demonstrates the usefulness of such methods. Questions about interpretability as well as the use of coarser data frequency are subject to further research. Similarly, different asset classes and exposures should be tested to draw more general conclusions about the applicability of this approach. Since this is a rapidly developing field, an increasing number of papers considering similar applications are expected to be written in financial applications. New models from the field of computer science can be further considered for the Portfolio Allocation problem. As such, this paper provides a comprehensive review of the performance of the current state-of-the-art algorithms and provides a study similar to DeMiguel et al., 2009 in the context of Reinforcement Learning methods. Its use is of direct interest to portfolio managers as an alternative approach to construct and support investment strategies with an actionable output that consists of optimal weights similar to the Mean-Variance approach.

References

Aguilar, O., & West, M. (2000). Bayesian dynamic factor models and portfolio allocation. *J. Bus. Econ. Stat.*, *18*(3), 338–357. https://doi.org/10.1080/07350015.2000.10524875

Black, F., & Litterman, R. (1992). Global Portfolio Optimization. *Financ. Anal. J.*, *48*(5), 28–43. https://doi.org/10.2469/faj.v48.n5.28

Bloomberg. (2020). Bloomberg professional [Available at: Subscription Service (Accessed: 27th December 2020)].

Carhart, M. M. (1997). On persistence in mutual fund performance. *J. Finance*, *52*(1), 57–82. https://doi.org/10.1111/j.1540-6261.1997.tb03808.x

Charpentier, A., Élie, R., & Remlinger, C. (2020). Reinforcement Learning in Economics and Finance.

Chen, J. (2019). Relative Strength Index – RSI Definition & Calculation. Retrieved April 12, 2021, from https://www.investopedia.com/terms/r/rsi.asp

Chen, L., Pelgery, M., & Zhuz, J. (2019). Deep learning in asset pricing. https://doi.org/10.2139/ssrn.3350138

Cong, L., Tang, K., Wang, J., & Zhang, Y. (2020). AlphaPortfolio for Investment and Economically Interpretable AI. *SSRN Electron. J.* https://doi.org/10.2139/ssrn.3554486

Danthine, J.-P., & Donaldson, J. B. (2014). *Intermediate financial theory*. academic press.

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *Rev. Financ. Stud.*, *22*(5), 1915–1953. https://doi.org/10.1093/rfs/hhm075

Elton, E. J., Gruber, M. J., & Spitzer, J. (2006). Improved estimates of correlation coefficients and their impact on optimum portfolios. *Eur. Financ. Manag.*, *12*(3), 303–318. https://doi.org/10.1111/j.1354-7798.2006.00322.x

Fama, E. F., & French, K. R. (1992). The cross-section of expected stock returns. *the Journal of Finance*, *47*(2), 427–465.

Goldfarb, D., & Iyengar, G. (2003). Robust portfolio selection problems. *Math. Oper. Res.*, *28*(1), 1–38. https://doi.org/10.1287/moor.28.1.1.14260

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical Asset Pricing via Machine Learning (A. Karolyi, Ed.). *Rev. Financ. Stud.*, *33*(5), 2223–2273. https://doi.org/10.1093/rfs/hhaa009

Harvey, C. R., Liu, Y., & Zhu, H. (2016). ... and the Cross-Section of Expected Returns. *Rev. Financ. Stud.*, *29*(1), 5–68. https://doi.org/10.1093/rfs/hhv059

Hayes, A. (2016). Average True Range (ATR). *Tech. anal. chart interpret.* (pp. 243–251). https://doi.org/10.1002/9781119204800.ch19

Hayes, A. (2017). Average Directional Index - ADX Definition and Uses. Retrieved April 12, 2021, from https://www.investopedia.com/terms/a/adx.asp%20http://stockcharts.com/school/doku.php?id=chart%7B%5C_%7Dschool:technical%7B%5C_%7Dindicators:average%7B%5C_%7Ddirectional%7B%5C_%7Dindex%7B%5C_%7Dadx

Hayes, A. (2019). Bollinger Band® Definition. Retrieved April 12, 2021, from https://www.investopedia.com/terms/b/bollingerbands.asp

Hill, A., Raffin, A., Ernestus, M., Gleave, A., Kanervisto, A., Traore, R., Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., & Wu, Y. (2018). Stable baselines.

iShares. (n.d.-a). iShares U.S. Real Estate ETF — IYR — US Class. Retrieved April 2, 2021, from https://www.ishares.com/us/products/239520/ishares-us-real-estate-etf

iShares. (n.d.-b). iShares U.S. Telecommunications ETF — IYZ. Retrieved April 2, 2021, from https://www.ishares.com/us/products/239523/ishares-us-telecommunications-etf

Kempthorne, P. (2013). *Factor Models MIT 18.S096 Lecture 15* (tech. rep.). MIT.

Kozak, S., Nagel, S., & Santosh, S. (2020). Shrinking the cross-section. *J. financ. econ.*, *135*(2), 271–292. https://doi.org/10.1016/j.jfineco.2019.06.008

Lapan, M. (2018). *Deep reinforcement learning hands-on: Apply modern rl methods, with deep q-networks, value iteration, policy gradients, trpo, alphago zero and more*. Packt Publishing Ltd.

Liu, X.-Y., Yang, H., Chen, Q., Zhang, R., Yang, L., Xiao, B., & Wang, C. D. (2020). FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance. *arXiv*. http://arxiv.org/abs/2011.09607

Markowitz, H. (1952). Portfolio Selection. *J. Finance*, *7*(1), 77–91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

Merton, R. C. (1980). On estimating the expected return on the market: An exploratory investigation. *Top. Catal.*, *8*(4), 323–361. https://doi.org/10.1016/0304-405X(80)90007-0

Meucci, A. (2005). *Risk and Asset Allocation*. Springer. https://doi.org/10.1007/3-540-27904-0

Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). *Performance Functions and Reinforcement Learning For Trading Systems and Portfolios* (tech. rep.).

Rice, J. A. (2006). *Mathematical statistics and data analysis*. Nelson Education.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. https://arxiv.org/abs/1707.06347v2

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489. https://doi.org/10.1038/nature16961

SPDR. (n.d.). Fund Finder - State Street Global Advisors. Retrieved April 2, 2021, from https://www.ssga.com/nl/en%7B%5C_%7Dgb/institutional/etfs/fund-finder?g=assetclass%7B%5C%%7D3Aequity!noLabel*Sectors%7B%5C%%7D7Cmarket-cap*Large%7B%5C%%7D5EMid

Tesau, C., & Tesau, G. (1995). Temporal Difference Learning and TD-Gammon. *Commun. ACM*, *38*(3), 58–68. https://doi.org/10.1145/203330.203343

Appendix

Features

Simple Moving Average

Simple Moving Average is defined as a moving window of unweighted average prices over some interval d such that:

$$\text{SMA}_d(p_t) = \frac{1}{d}\sum_{i=n-d+1}^{n} p_i.$$

We can calculate SMA by moving this kernel along the time-series to obtain a smoothed version of the prices. In the training procedure, windows of 30 and 60 days were used.

Moving Average Convergence Divergence

MACD is also a trend feature that is constructed using two Exponentially Weighted Moving Averages:

$$\text{MACD}_{d^s,d^l} = \text{EMA}_{short} - \text{EMA}_{long}$$

This EMA smoothing of the prices is similar in logic to SMA, but the kernel applied here on the time series window is not uniform but exponential. This means that more recent observations are given more weight in the averaging, creating an auto regressive filter of the prices. We can define EMA recursively as:

$$\text{EMA}_d t = \left\{ \begin{array}{ll} p_t, & t = 1 \\ \alpha p_t + (1-\alpha)\cdot p_{t-1}, & t > 1 \end{array} \right\},$$

where $\alpha \in [0,1]$, $\alpha \approx \frac{2}{d+1}$ as derived from the power series expansion of the EMA definition. For the long EMA, the window of 26 days is used whereas for the short window I use 13.

Bollinger Bands

Bollinger Bands are a combination of a SMA and an upper Bollinger band and a lower Bollinger band. A Bollinger band is a transformed SMA by adding an estimate of a short standard deviation as a way to capture signals outside of 'normal' behavior (Hayes, 2019). We express it as:

$$\text{BB}_d = \text{SMA}_d \pm K\hat{\sigma_d}$$

where we estimate σ_d as $\hat{\sigma} = \frac{1}{d-1}\sqrt{\sum_{i=1}^{d}(p_i - \bar{p})^2}$. We use $d = 20$ for the estimation window and $K = 2$ as price moves outside of this band are considered unusual and can pose a positive or a negative signal. We directly input the upper and lower band as two features for the training of the reinforcement learning model

Relative Strength Index

RSI is a momentum feature that tries to capture oscillations within the price that could suggest a directional shift. It's bounded from 0 to 100 where a low value suggests a bullish signal and a high value a bearish signal within the selected timeframe (J. Chen, 2019). We can calculate it as follows:

$$\text{Relative Strength}_d = \frac{\bar{\text{U}}}{\bar{\text{D}}},$$

where commonly $\bar{\text{U}}, \bar{\text{D}}$ are estimated using an EMA_d of U or D, and

$$\text{p}_d = \left\{ \begin{array}{ll} u_t, u_t \in \text{U} & \forall p_t^{close} > p_{t-1}^{close} \\ d_t, d_t \in \text{D} & \forall p_t^{close} < p_{t-1}^{close} \end{array} \right.$$

Then we just index the Relative Strength from 0 to a 100 as:

$$\text{RSI}_d = \frac{100 - 100}{1 + RS}.$$

Commonly a value $d = 14$ is used to estimate the RSI.

Commodity Channel Index

CCI is also a momentum feature typically used to establish longer term trends. We use it with a configuration of 22 days, or a full trading month to complement RSI and capture longer term trends. We define it as follows:

$$\text{CCI}_d = \frac{\hat{p}_t - \text{SMA}_d(\hat{p}_t)}{0.015\frac{1}{n}\sum_{i=1}^{n}|p_t - \text{SMA}_d(p_t)|},$$

where $\hat{p}_t = \frac{p_t^{high} + p_t^{low} + p_t^{close}}{3}$.

Directional Movement Index

Directional Movement Index, often used in an averaged form over a period of time is a momentum feature that tries to gauge the strength of a formed trend. We can define it using these quantities as in (Hayes, 2017) first defining a directional movement as:

$$\begin{array}{l} \text{DM}_t^+ = p_t^{high} - {}_t^{high} \\ \text{DM}_t^- = p_t^{low} - {}_t^{low}. \end{array}$$

We also need the Average True Range, a non parametric definition of volatility of a security (Hayes, 2016) defined as:

$$\text{ATR}_t = \frac{1}{d}\sum_{t=1}^{d}\max[(p_t^{high} - p_t^{low}), \text{abs}(p_t^{high} - P_t^{close}), \text{abs}(p_t^{low} - p_t^{low})].$$

Then consider a smoothing function of the arbitrary form:

$$\gamma_d(x_t) = \sum_{t=1}^{d} x_t - SMA_d(x_t) - x_t.$$

Hence we can define the directional indicators and directional movement index as:

$$DI_t^{\pm} = \frac{\gamma_d(DM_t^{\pm})}{ATR_t}$$
$$DMIX_t = \frac{|DM_t^+ - DM_t^-|}{|DM_t^+ + DM_t^-|} \times 100.$$

In this paper, $d = 30$ is used for capturing medium-term dynamics within the trend for the components of $DMIX_t$.

Covariance Matrix

For each state s_t I also calculate a sample covariance matrix $\hat{\mathbf{\Sigma}}$ that is used as a feature to train the associated neural networks. A lookback of one year or 252 trading days is used, and the covariance matrix is estimated as:

$$\hat{\mathbf{\Sigma}} = \frac{1}{N-1} \sum_{t=1}^{N} (\mathbf{p}_{\cdot t} - \hat{\mathbf{x}})(\mathbf{p}_{\cdot t} - \hat{\mathbf{x}})^T$$
$$\hat{\sigma}_{ij} = \frac{1}{N-1} (\mathbf{x}_{i,t} - \hat{\mathbf{x}}_{\mathbf{i}})(\mathbf{x}_{j,t} - \hat{\mathbf{x}}_{\mathbf{j}}).$$

Turbulence Index

I also indirectly condition on the overall level of volatility in the enviroment $\mathcal{E}$ used for. This is done using a turbulence index as defined in (Liu et al., 2020):

$$TB_t = (\mathbf{r}_t - \boldsymbol{\mu}) \hat{\mathbf{\Sigma^{-1}}} (\mathbf{r}_t - \boldsymbol{\mu})^T,$$

where $\mathbf{r}_t$ is a vector of stock returns in $\mathcal{E}$, $\boldsymbol{\mu}$ is a vector of expected returns estimated as a simple average historic returns over a period of 252 trading days, and $\hat{\mathbf{\Sigma}}^{\mathbf{-1}}$ is an inverted sample covariance matrix as described in previous section.

Selection of securities for collateralized debt obligations contracts : an optimization approach applied to European SME loan portfolios

August 2021

Abstract

Collateralized Debt Obligations are a subclass of financial derivatives that allow banks to transfer their credit risk to external investors. To maximize the transferring power of those products, CDOs must be as attractive as possible for investors. To this aim, we formulate in this thesis a loan selection optimization problem. We consider a problem where the originator of the CDO aims at maximizing the attractiveness of the senior tranche while ensuring that the other tranches still have a coherent risk profile by choosing the right loans to include in the collateral portfolio. This optimization problem is a large-scale non-linear integer optimization problem. Solving the problem is challenging from a numerical point of view. Hence, several approximations and simplification of the true problem are proposed in this thesis and benchmarked. We observe that the approach developed to solve the problem outperforms benchmark techniques while finding solutions in a few minutes. We also observe that results yielded by the optimization algorithm appear sensitive to some user-defined parameters and that the simplifications perform significantly worse for small portfolios (less than 100 loans).

1 Introduction

The core business of traditional commercial banks is to lend their available funds to their clients and to earn revenue thanks to interest payments. A significant proportion of the banks' assets are therefore composed of the receivables of their various loans. Banks are consequently widely exposed to credit risk.

To reduce this exposure, banks can transfer their various claims to external investors. But, in it initial setting, a claim on an obligor is an unattractive investment. The risk is indeed concentrated in a single name and there is no secondary market for the claim. To make investment in credit assets more attractive, banks have therefore developed a wide range of credit derivatives, including asset backed securities (ABS) and collaterallized debt obligations (CDOs).

Those financial products are much more attractive for an external investor than the underlying assets, which therefore allow financial institutions to transfer a part of their credit risk to an external investor, freeing up capital reserves that can be reinvested in new loans.

To maximize the efficiency of those products, they must be designed to be as attractive as possible. Appointed portfolio managers have two main levers to reach that goal. Firstly, CDOs have a tranche structure (see appendix8.1), and portfolio managers can therefore change the different attachment points to change the riskiness of the various tranches. Secondly, depending on the assets chosen in the collateral pool, the notes will be more or less attractive.

This work focuses on the second lever, the exact assets chosen. Indeed, often, portfolio managers receive a list of loans that are eligible for securitization, and have to select, from this list, the final portfolio that will serve as collateral. To reach their goal, they often rely on a time-consuming trial and error approach.

The objective of this work is to automate the process, wtih the help of optimization algorithms. But, the problem of selecting the optimal loans is a large scale non-linear integer problem, who, in its initial setting, might require several days in order to be solved numerically. On the contrary, the objective of this paper is to introduce several simplifications, so that solutions can be found in a few minutes. Indeed, the long running time would prevent the use of the algorithms by portfolio managers. The work has been realized in collaboration with the European investment fund, whose portfolio managers often face the choice of the exact assets to include in created financial products.

2 Mathematical formulation

This paper focuses on choosing a pool that make the most senior tranche as attractive as possible for investors while considering constraints on the other tranches. The main reason to focus on the most senior tranche is that, in most securitization deals, the senior tranche is by far the largest. Additionally, it is assumed in this work that the tranche attachment and detachment points are given exogenously.

To quantify the attractiveness, an EIF specific rating is used. The lower the rating, the more a tranche is considered attractive[1], all others being equal. To account for both the preferences of investors regarding the potential losses and timing of those losses, the rating is a function of the Expected Loss EL_w and Weighted Average Life WAL_w of the tranche w. The rating is increasing with regards to the EL_w (the more losses, the less attractive),

[1]The rating is a measure of the credit quality of the different tranches. Hence, the higher the rating, the higher the return investors will ask in exchange for their investment.

and decreasing with regards to the WAL_w (investors prefer having losses spread across a long period rather than within a short time). To calculate the rating, a table is used that, for any level of EL_w and WAL_w, returns a specific rating. A contour plot of the rating can be found in appendix 8.2.

$$Rating = R(EL_w; WAL_w) \tag{2.1}$$

Formally, there are N different loans that can potentially be chosen. For any chosen portfolio P, ω_i is the proportion of loan i that has been selected. Because loans can only be selected fully, ω_i is constrained to equal either 0% or 100%. Analytically, the chosen portfolio can be can be represented by the vector $\Omega = (\omega_1, ..., \omega_N)$. The total initial collateral of the final portfolio P consisting of loan i equals to $\sum \omega_i O_i$ where 0_i is the principal outstanding of loan i.

The time when a loan of the portfolio defaults, is denoted by the random vector $\mathcal{T} = (\tau_1, \tau_2, .., \tau_N)$. If τ_i is smaller than the maturity of loan i, the loan defaults and a loss is incurred. Obviously, the EL and WAL of every tranche w depend on the vector τ, chosen weights Ω and their respective attachment and detachment points. The mathematical tools for deriving the distribution of vector $\mathcal{T}$ will be presented in section 3, but the EL, WAL and hence rating of the tranches are dependent on the weights Ω of each loan.

For now, let's define the set of functions $(f_1(\Omega), f_2(\Omega), f_w(\Omega))^2$, that link the rating of the different tranches w to the chosen assets Ω.

We are now in position to formulate a loan selection problem where the portfolio manager seeks to maximise the attractiveness of the most senior tranche (and hence minimize its rating), but faces additional constraints. Firstly (a), the other tranches still need to have a coherent risk - profile and therefore their rating needs to remain in defined boundaries (between $R_{l,w}$ and $R_{h,w}$). Secondly, usually, there will be additional constraints that stipulates that the collateral exposures to a particular country C or industry I cannot be higher than a certain value V_I and V_C (b) and (c). Thirdly (d), when securing assets, originators will target a specific size for the collateral (for instance, the collateral has to be between 250M€ and 500M€), that has to be between S_L and S_H. The loan selection problem can therefore be written:

$$\begin{aligned} &\min_{\Omega} f_a ts(\Omega) \\ &\text{Subject to} \\ \forall w : R_{l,w} \leq\ & f_w(\Omega) \leq R_{h,w} \ (a) \\ \forall I : & \sum \omega_{iI} O_{iI} \leq E_I \ (b) \\ \forall I : & \sum \omega_{iC} O_{iC} \leq E_C \ (c) \\ S_L \leq & \sum \omega_i O_i \leq S_H \ (d) \end{aligned} \tag{2.2}$$

In addition to the constraints in problem (2.2), additional constraints can also be considered. For instance, constraints on the WAL or maturity of the underlying portfolio can be added.

2.1 Challenges of loan selection problems

This problem shares, at first glance, some similarities with traditional modern portfolio theory, which is widely studied since the framework introduced by (Markowitz, 1952). indeed, just like portfolio optimization, loan selection aims at investing in some loans so as to minimize a constrained cost function, related to a measure of risk. However, there are several differences that make them quiet different in practice. Firstly, the risk metrics that are optimized are different in both approaches. Indeed, a portfolio manager of structured credit instrument must create the most attractive financial product for an external investor. The portfolio of external investors will not solely consist of the structured product and hence, there is little point in minimizing the variance. Secondly, there is an integer constraint on each loan. Those differences make solving loan selection problems much more challenging than its equity-based counterpart, for at least 4 reasons.

1. As the list of eligible loans can reach several thouasands, the size of the problem is very large
2. The problem is mixed integer problem, because of the binary constraints on each loan. Those problems are non-polynomial time hard, which means that the number of computations needed to solve the problem grows exponentially with the number of loans available.
3. There is no closed form expression of the expected loss and WAL of each tranche. Evaluating those quantities and hence objective and constraint functions relies mostly on computer-intensive semi-analytical or Monte Carlo techniques.
4. The particular shape of the problem does not allow to use gradient-based optimization solvers. Less efficient derivative-free solvers must be used.

To overcome those challenges, several approximations of the true optimization problem that we have developed in

[2] to determine $f(\Omega)$, the multivariate distribution of vector $\mathcal{T}$ has to be known. The methods used to determine this multivariate distribution will be presented in section **??**

order to find close-to-optimal solutions in short amount of time will be presented in section 4.

3 Credit risk model

Two risk metrics must be calculated to solve the loan selection problem (2.2), the WAL of the tranche and the expected loss. To do so, full distribution of the portfolio loss rate L_{pf} must be calculated and a dependence across the default events of the obligors in the underlying pool must be introduced. Since (Li, 2000) seminal paper *On default correlation: A copula function approach*, the primary tool to price CDOs and introduce a dependence structure for the vector of time to default variables τ, the primary mathematical tool used by finance professionals are copulas. In this thesis, the rather simplistic one-factor Gaussian copula framework, who was the market standard before the 2008 financial crisis, is used to compute those quantities.

In one-factor Gaussian copula framework, with equicorrelated loans with correlation equaling ρ, defaults are made dependent through a latent-variable model. A loan i defaults before a specific date t if a latent variable Y_i falls below a specific threshold k_t. Its latent variable Y_i is the sum of a random variable M, common across all obligors, and an idiosyncratic random variable ϵ_i, specific to obligor i. This can be written:

$$Y_i = \sqrt{\rho}M + \sqrt{1-\rho}\epsilon_i \tag{3.1}$$

In the Gaussian copula framework, M and ϵ_i and hence Y_i are considered to be distributed according to the standard-normal distribution, and conditional on M, defaults are independent. The tresholds, $k_{i,t}$, can be inferred from the marginal default probability curve by noting that :

$$\mathbb{P}\left(Y_i < k_{i,t}\right) = F_i\left(t\right) \tag{3.2}$$

where $F_i(t)$, the marginal default curve, gives the marginal probability that loan i default before t. The tresholds therefore equal to: $\Phi^{-1}\left(F_(t)\right)$.

Conditional on M, the defaults are independent. Several semi-analytical techniques can be used to calculate the distribution of the portfolio loss rate L_{pf}, from the conditional distributions $L_{pf|M}$, where defaults are considered independent.

4 Simplifications

Having introduced the dependence structure across default events in previous section, we introduce in this section 3 simplifications that allow to overcome the numerical challenges presented in section 2.1.

4.1 Approximation of losses with one-period model

The objective was to develop a method that can be used on portfolios consiting of amortizing assets. The exposure and hence loss-severity o those assets vary over time. As a result, to compute the tranche's expected loss, a Monte Carlo technique simulting all cash-flows paid must be used. To avoid the numerical burden and approximate the tranche losses, we approximate the distribution of the losses of the different tranches with a one-period model. In our model, we have following quantities for each loan:

- $F_i(t)$ is the marginal default probability curve of loan i. It is a function that, for each t, returns the cumulative probability of defaulting before period t.
- RR_i, is the recovery rate in the case that loan i defaults in proportion of the outstanding capital. The recovery rate is assumed constant in our model.

Furthermore, with the single-period approximation, the tranche losses depend only of the total number of losses L_{pf} that the portfolio has undergone over the course of the existence period of the structured credit instrument. Hence, for a given scenario of losses, tranche loss rate of a tranche w TL_w equals to:

$$TL_w = \min\left(\max\left(\frac{L_{pf} - A_w}{D_w - A_w}; 0\right); 1\right) \tag{4.1}$$

To approximate with a single period, we construct for each loan follwing quantities:

- p_i: a WAL-adjusted marginal probability that the loan defaults before the maturity of the CDO notes, inferred from the marginal default curve. The WAL adjustment is performed to account for the difference in riskiness of assets with different amortizing profiles. Indeed, all other things being equal, a Bullet asset that repays all its principal at maturity is riskier than an amortizing loan[3].
- RR_i: a recovery rate in case the loan defaults, in the proportion of the initial notional of loan i. This quantity is assumed fixed and not random.
- O_i: the principal outstanding of loan i at the origination of the CDO.
- ω_i: a decision variable that equals 1 if the loan has been selected in the final portfolio and 0 if not.

[3] The principal at risk for a bullet loan is the same after day one as one day before its maturity. Hence, the potential loss severity of a bullet loan is higher than for an amortizing loan since the exposure to an amortizing loan is reduced each time a principal repayment is made.

4.2 Asymptotic distribution of losses

To further simplify the problem, we use the widely studied fact that, in the Gaussian copula factor framework, when the exposure to a single obligor tends to 0,or in other words when the portfolio tends towards infinite granularity, the loss distribution will tend to the distribution of its expectation conditional on the factor M. This result lies at the basis of the Basel 2 regulatory framework. As exposed in appendix 8.3, this approximation is very accurate for portfolios that display a low concentration of exposures in a few obligors.

This can be used for approximating the expected loss of a CDO tranche. Indeed, the conditional expectation of the portfolio loss rate is:

$$\mathbb{E}[L_{pf}|M] = \frac{\sum_i \omega_i O_i \Phi\left(\frac{\Phi^{-1}(p_i)-\sqrt{\rho}M}{\sqrt{1-\rho}}\right)(1-RR_i)}{\sum_i \omega_i O_i} \quad (4.2)$$

Then, similarly as in Burtschell et al. (2009), an approximation of the tranche expected loss is given by substituting equation (4.2) in following formula:

$$\mathbb{E}[TL_w] \approx \int_{-\infty}^{\infty} min\left(max\left(\frac{\mathbb{E}\left[L_{pf}|M\right]-A_w}{D_w-A_w}\right);0;1\right)\phi(M)dM \quad (4.3)$$

Calculating the expected tranche loss EL_w with formula (4.3) takes a fraction of a second on a usual computer. The approximations of this section hence circumvent the challenge of the long computing time of objective and constraint functions.

4.3 Ammortization profile

The optimization problem (8.16) does not only depend on the expected loss of the different tranches but also on their WAL who is linked to the amortizing and default profile of the underlying portfolio and the considered waterfall, which is not captured in the simplified one-period model.

We have made several assumptions to be able to derive a closed-form expression of the WAL of the different tranches. They are explained in detail in appendix 8.4. But can be summarised as follows:

- The collateral pool is subdivided in two sub-portfolios, a defaulted portfolio comprising all loans that default before their maturity and a performing with all loans that do not default.
- The contribution of both portfolios to the tranche principal repayments are weighted by their size.
- The ammortization profile of both portfolios is assumed to be determined by the ratio between the weighted average of the WAL of the loans in the portolio

4.4 Clustering

Both the approximation of the EL of tranches and assumptions enabling to find their WAL significantly speed up the evaluation of constraint and objective functions of the loan selection problem. A last numerical challenge still remains. The size of the problem, the number of loans can potentially be very large and furthermore, there is a binary constraint on the weights ω_i that can be invested in each loan.

To reduce the size of loan selection problems, Sirignano et al Sirignano et al. (2016) argue that if two loans have very common loan features X_i and X_j, or in other words, $||X_i - X_j||$ is small, selecting loan i instead of loan j will have a small impact on the objective function and constraints.

Therefore, loans can be clustered together in relatively homogeneous groups. The binary decision variables of selecting a loan or not is then replaced by the continuous decision variable of the proportion to invest in each cluster. This has two sources of computational advantages. Firstly, the integer constraints are dropped. Secondly, the number of clusters can remain constant when the number of loans increases. Therefore, the computational cost of the solver does not increase anymore with the size of the initial portfolio. Not all features are, according to our loss model and assumptions, useful for the calculation of the constraints and objective functions. Only the loan features impacting those functions must be used. Appendix 8.7.1 provides more information about the exact features considered.

5 Overview of algorithm for solving the loan selection problem

The previous section introduced several simplifications that enables to solve most of the numerical issues related to the estimations of the objective and constraint functions of problem 2.2 and to reduce its size. The next step is to find algorithm that performs following steps:

1. Starts from the initial portfolio with N loans
2. Cluster them in R different clusters
3. Compute the vector of optimal weights Ω_r^* invested in each cluster, that minimizes the rating given various constraints, with the help of an optimization solver.
4. Select the loans of the final portfolio in order to replicate the vector of optimal weights.

It has been decided, for the first step, the clustering , similarly as in Sirignano et al. (2016), to use a k-mean clustering algorithm. The last step, is done in order to minimize the difference between the final weights Ω_r^f and the optimal weights found by the solver Ω_r^*. The conversion is explained in detail in appendix 8.8.

5.1 Finding the optimal weights

To compute the vector $Omega_r^*$, a last numerical challenge must be overcome: no gradient of the objective function can be calculated and hence, gradient-based optimization solvers must be used. Those solvers suffer from several drawbacks. They can converge to local minima and their convergence time is usually longer than their gradient-based counterparts(Larson et al., 2019). In this section, we show that the specificities of the rating function, namely the fact that it is always decreasing with regards to the WAL and increasing with regards to the expected loss, can be exploited to find a reformulation that solves aforementioned issues.

To do so, the algorithm explores a small sub-region of the feasible weights: the optimal region.

5.1.1 Finding the optimal region

To understand where this sub-region lies, consider a feasible portfolio P with rating of the most senior tranche R_p. Only portfolios that yield a smaller EL_s or higher WAL_s of the senior tranche can potentially improve the rating of the portfolio. Furthermore, if we restrict the region to a specific level of weighted average life of the most senior tranche, then it is guaranteed that the best portfolio in this sub-region is the portfolio that minimizes the EL of the senior tranche.

Hence, the global optimal portfolio lies in the sub-set of portfolios P_{minEL} that for every feasible level of WAL of the senior tranche minimizes the EL. The search-region can therefore been restricted to P_{minEL}, that we refer to as the optimal region.

To find the optimal region, the problem of minimizing the expected loss must be solved. The expected loss of a tranche, calculated with equation (4.3), is an integral of the form $\int f(M)\phi(M)dM$. Those integrals can be approximated as a weighted sum of function evaluations of $f(M)$ at specific points, denominated nodes, with the help of a Gaussian quadrature technique(Choi et al., 2021). Rewriting the integral as a weighted sum does not only speeds up the evaluation of the integral, it also allows to reformulate the problem of minimizing the expected loss as mixed-integer linear problem. The reformulation can be found in appendix 8.7.1.

MILP have several advantages over problems that have to be solved with derivative-free techniques.

1. MILP do not have local minimums. Hence, the algorithms can only converge towards the global optimal result;

2. The running time of mixed-integer linear solvers is typically shorter than the running time of derivative-free algorithms for problems of the same size;

3. Constraints that are written as linear quantities from the chosen weights Ω_r are easily implementable. The various constraints regarding exposures to countries are industries, the weighted average maturity of underlying pool can therefore be implemented in a straightforward manner.

Two artificial constraints can be added to the mixed integer linear problem of minimizing the expected loss, to restrict the optimal weights found to a specifc level of WAL and weighted average maturity of the underlying pool. Thanks to the assumptions in section 4.3, this restricts the weights to a specific WAL of the senior tranche. Those constraints can be written:

$$\begin{aligned} \frac{\sum \omega_r WAM_r}{\sum \omega_r} &= WAM \\ \frac{\sum \omega_r WAL_r}{\sum \omega_r WAL_r} &= \gamma \end{aligned} \tag{5.1}$$

A portfolio lying on the optimal region can therefore be found by for a specific level of WAL_s of the senior tranche can be found by fixing an arbitrary value for those constraints and solving the MILP. The optimal weights, for their part are at specific value of those two parameters, that has to be found.

5.2 Exploring the optimal region

To find the optimal value of those two paramters, a computer function that executes the following code can be considered:

Algorithm 2 Cost-function used to find optimal portfolios

procedure CONSTRAINED OPTIMAL PORTFOLIO(WAL, WAM)
  Construct problem 8.11
  Calculate WAL_s Senior ▷ (with formula **??**) and WAL and WAM provided to the function.
  Add WAL and WAM constraints (constraints 5.1)

  CALL MILP Solver ▷ The MILP solver finds the weights that minimize EL given WAL and WAM constraints
  ▷ The objective value reached by the solver is EL_s
  CALL RatingFunction(EL_{sen}, WAL_{sen})
  ▷ Formula 2.1
  return rating

When entering a feasible WAM and WAL of the underlying portfolio, the function returns the rating associated to this precise set of parameters, and hence WAL of the senior tranche that minimizes the expected loss. As a consequence, all solutions found by the MILP solver in the function lie in the optimal region. The optimal parameters WAM^* and WAL^* that minimize the function yield the optimal weights Ω_r^* when introduced as additional WAL and WAM constraints in optimization problem 8.11. Those parameters can be found, instead of by calculating the whole optimal region, by using an optimization algorithm that uses this function as an objective function.

This function does not have an analytical expression, nor a gradient. Therefore, to minimize the function, a gradient-free optimization method has to be used. To minimize this function, the Nelder-Mead algorithm can be used. The Nelder-Mead method is a heuristic method that computes a local optimum of a function.(Dennis Jr and Woods, 1985). The method is based on the evaluation of the function objective on a non-linear simplex.

The optimization problem in its simplified setting could also be solved by a derivative-free optimization method. However, the reformulation introduced in the previous two sections enhances the efficiency of the derivative-free algorithm. In its simplified setting, the size of the problem for the derivative-free algorithm was the number of clusters R. When optimizing against this function , the size of the problem is reduced to two, namely the value WAL and WAM. Hence, the number of iterations needed to converge will be way smaller.

Secondly, the explored region by the algorithm optimizing a function that returns only results that minimize the expected loss is way smaller and concentrated on the region where the optimal value lies with certitude.

6 Numerical results

6.1 Methodology

Having showed how the numerical challenges can be circumvented to find close to optimal solutions in less than a few minutes, we have to asses wether the results yielded by the optimization algorithm developed outperform other selection techniques and if the results yielded are reliable. To do so, we tested the performance numerically on a large database of 1200 different randomly generated initial portfolios with different underlying loan-features varying according different dimesnions. More details about the data can be found in appendix 8.7, and details about the practicalities of the numerical integration of the algortihm can be found in appendix 8.8.

Subsequently, we compared the rating of portfolios found by the optimization for a benchmark problem described in appendix 8.9 with the rating of portfolios found with other benchmark techniques. The main benchmark techniques consisted of drawing randomly a large number of portfolios and selecting in priority loans with the smallest expected loss.

6.2 Results

6.2.1 Comparison with random results

All of the analysis is based on the results that can be found in appendix 8.10.

- Even though our method relies on several simplified assumptions, it enables to find in less than 30 seconds in most cases portfolios that outperform our benchmark, random selection.
- On average, the rating of the senior tranche of portfolios found with the optimization outperforms the best random solution by 3.2, and an average random result by 3.93
- When analyzing the results more in-depth, we can note that the primary lever that the optimization routine chooses is to improve the credit quality of the underlying pool of loans. Portfolios chosen by the optimization have a rating average of 10,59, which is 2.90 times better than the average ratings of randomly selected portfolios.
- The optimization on small (less than 100 initial loans) and concentrated portfolios performs significantly worse.

6.3 Comparison with heuristics

In this section, the result of the optimization is compared with benchmark portfolios that have been constructed by selecting the 60% loans that have the smallest EL. The analysis is based on the results displayed in appendix 8.11.

As can be seen in table 10, when calculated with the LHP approximation, the approximate optimal solutions are very close to those found by this heuristic.For 5 out of 10 portfolios of this sample, the simplified optimal solution beats the solution found with the heuristic of taking in priority loans with the smallest expected loss. Nevertheless, the various constraints encountered in realistic loan selection problems do not allow such a simple heuristic.

6.4 Performance of the MILP reformulation

In this section, the performance of the reformulation is compared with more traditional formulations, that leverage derivative-free algorithms. The objective value being optimized is the rating of the senior tranche, where the EL is calculated with the infinitely granular approximation (equation (4.3)). The decision variables are the weights invested in each cluster. The value reached with the reformulation, who leverages both a MILP solver and a derivative-free one(to find the optimal value for the WAL and WAM parameter) is compared to a more traditional derivative-free algorithm that minimizes a cost - function based on the weights invested in each cluster. More details about the cost-function being optimized can be found in appendix 8.13.

Comparing the results yields (see appendix 8.5) :

- The optimal weights found by the reformulation with a Gaussian quadrature outperforms all weights found with a traditional technique.
- The running time is significantly faster with the Gaussian quadrature: 104 seconds compared to up to 1 hour for the derivative-free optimization algorithm.
- The number of iterations is indeed much lower for the Gaussian quadrature technique. Only 56 iterations are needed, even though the time of an iteration with the Gaussian quadrature technique is much longer. Indeed, a MILP must be solved at each iteration, which is more computationally intensive than computing the rating.

6.5 Sensitivity to user-defined parameters

The effect of user-defined parameters must be investigated aswell. It is of primary importance to check that the choices made on several parameters do not lead to significantly different solutions. Four different parameters have been investigated. The number of clusters, the standardisation method, the loan selection criteria to match the weights, and finally, the number of node points chosen for the quadrature. A detailed analysis about the effect of those parameters can be found in appendix 8.12. The main takeways from this analysis is that the results, in some specific cases, can be sensitive to the number of clusters chosen and the number of quadrature nodes. The sensitivity to the other parameters appeared to be lower. Hence, restarting the algorithm several times with different values for those parameters in order to compare the results might appear adequate.

6.6 Sensitivity to data-estimation errors

Lastly, two metrics are mostly subject to estimation errors: the recovery rate and the one-year probability of default. The sensitivity of the results to those estimations must therefore be computed. Naturally, we expect the sensitivity of the model to estimation errors to be small. Even slightly modified, the loans will probably be assigned to the same cluster. Furthermore, if there is no bias in the estimation errors of loan features, the errors will probably wash out within the cluster and hence, the cluster features used by the optimization algorithm would remain close to those of the reference portfolio. The exact procedure and results yielded for estimating the sensitivity can be found appendix 8.14. The main results are:

- The sensitivity to the probability of default is small. Approximately 90% of loans that were selected initially are still selected after having introduced estimation errors. The impact on the metric of the senior tranche remains moderate, 6% in the worst case.

- The sensitivity to relative errors on the recovery rate is larger. This is not surprising because recovery rate estimates are typically higher than probabilities of default. Hence, the impact of relative errors might be more important.

7 Conclusion

In this thesis, the problem of selecting the right loans that will serve as collateral for an asset-backed security is considered from an optimization perspective. Besides formulating and calculating the problem, this thesis also focused on introducing several simplifications to be able to find close-to-optimal solutions within an efficient time frame (less than few minutes).

The objectives pursued by portfolio managers in charge of the structuring of ABS differ widely from investors. Moreover, no standard formulation of the loan selection problem has been widely adopted. This thesis therefore started by formalizing the objective of a portfolio manager selecting loans as a problem of maximizing the attractiveness of the ABS for investors. Mathematically, this has been defined as minimizing a specific risk rating of the senior tranche that depends on its EL and WAL.

Solving the loan selection problem from a numerical point proves to be way more challenging than its equity-based counterpart. Our first main contribution towards simplifying the problem is that the large size of the problem and its binary formulation can be significantly reduced by clustering the loans together in homogeneous clusters. The binary decision variable of selecting a loan or not is replaced with a continuous decision variable of how much to select in each cluster.

Secondly, we have used the well-known result that the loss distribution of a loan portfolio converges towards its conditional expectation when the granularity of the portfolio increases.

Lastly, we show that Gaussian quadrature techniques to approximate the various integrals that have to be calculated allow to rewrite the problem of minimizing the Expected Loss of a tranche as a mixed-integer linear problem. This avoids problems linked to usual non-linear solvers, resulting in increased speed and reliance on those solutions.

Various numerical tests have been performed to assess the reliability and performance of our simplifications. Our main result is that, in our simplified model, the credit quality of the underlying pool of loans is strongly correlated with the credit quality of the most senior tranche. Our portfolio outperformed almost systematically randomly selected portfolios. However, when evaluated with a large homogeneous pool approximation, the rating of the most senior tranche did not outperform portfolios constructed by using the heuristic of selecting the loans with the smallest EL.

Nevertheless, numerous parameters must be (user) defined before using this simplified solver. The main parameters are the number of clusters, the selection criteria of loans and the number of quadrature points. Assessing the impact of those parameters on the solutions proved that results yielded by the optimization algorithm can be improved by changing the parameters used and unfortunately, no general trend has been identified. The optimal set of parameters could therefore be case-specific. On the other hand, our model showed a small sensitivity towards data estimation errors.

8 Appendix

8.1 Tranche structure

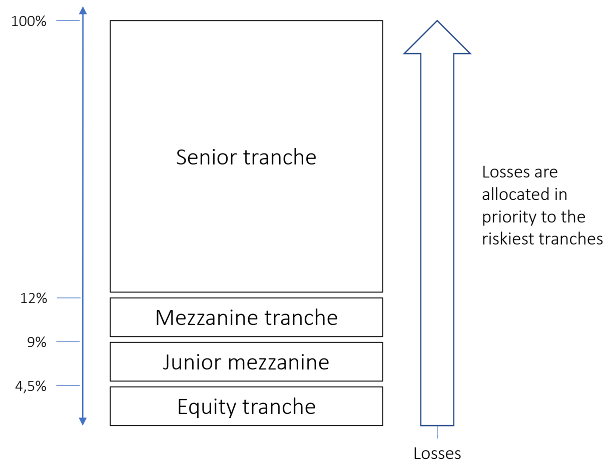


Figure 1: Example of a CDO structure with 4 tranches, with attachment points 0%, 4.5%, 9% and 12%. The two riskiest tranches are commonly referred to as junior tranches.

8.2 Contour plot rating

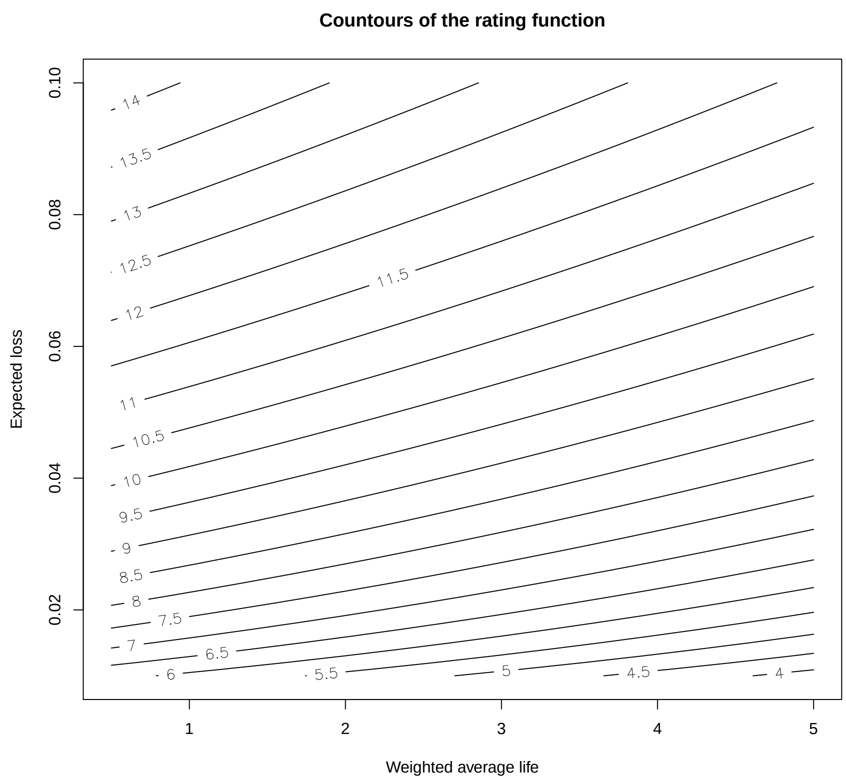


Figure 2: Contours of the rating function (2.1) for tranches with expected losses varying between 0.001 and 0.1 and WAL between 0.5 and 5 years. The rating function is an increasing function of the expected loss and decreasing a decreasing function of the WAL.

8.3 Asymptotic distribution of losses

To quantify concentration, one metric that can be used is the Herfindahl-Hirschman Index(Slime et al., 2016). This index can be calculated as follows. Let's define $s_i = \frac{A_i}{\sum A_i}$ as the share of exposure concentrated in credit instrument i. Then, the Herfindal-Hirschman index is calculated as follows:

$$HHI = \sum_i s_i^2 \tag{8.1}$$

Obviously, the closer the index is to one, the more concentrated the portfolio and consequently, the bigger the distance in distribution between $\mathbb{E}[L_{pf}|M]$ and L_{pf}. Fig 3 shows the difference between the quality of the approximation for a concentrated ($HHI = 0.1$) and for a granular portfolio ($HHI = 0.01$).

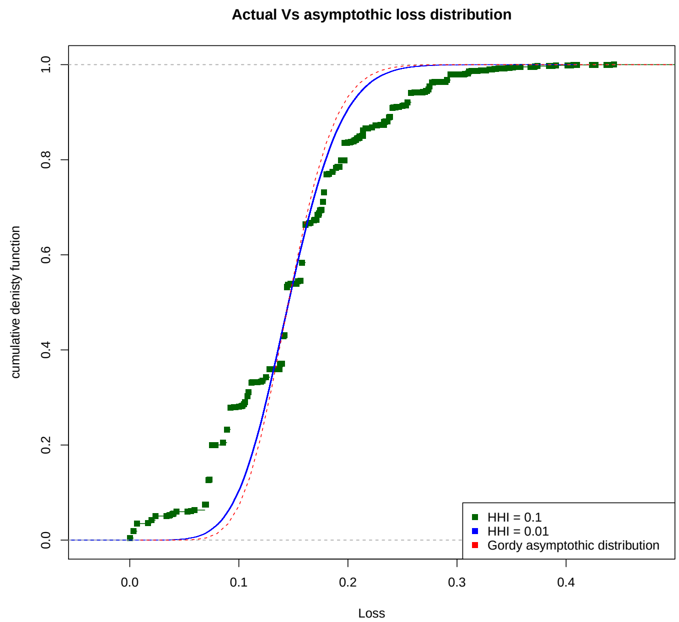


Figure 3: The difference between the asymptotic distribution and the simulated loss distribution is way larger for concentrated portfolios. The green (HHI = 0.1) and the red (HHI = 0.01) curve have been computed via a Monte Carlo simulation involving 100 000 simulations. The blue curve is computed with the cumulative distribution function of the conditional expected losses $\mathbb{E}[L_{pf}|M]$. The correlation parameter ρ equals to 10%.[4]

8.4 WAL calculation

The WAL_w of a tranche of a CDO is the weighted average times of the payments and losses allocated to the tranche. All payments to the CDO tranches are made with cash-flows from the underlying pool of loans. There are three types of cash-flows a loan pays: principal repayments, interests, and if the loan defaults a recovery amount. The first assumption that we do to simplify the calculations of the WAL_w is to divide the underlying pool into two-sub -portfolios. One performing sub-portfolio consisting of all loans that do not default before the maturity of the CDO tranches and one defaulted portfolio, comprising all defaulted loans. The performing sub-portfolio is the source of principal and interest repayments and the defaulted sub-portfolio is the source of losses.

The performing sub-portfolio is characterized by an amortization function g(t) that at each moment t gives the proportion of collateral that is still outstanding. For notational simplicity, it is considered that at the origination of the CDO, $t = 0$ and that $t = T$ when all loans of the CDO have matured. At $t = 0, g(0) = 100\%$ and at $t = T, g(T) = 0$.

To make assumptions about this function g(t) we start by considering two very simple cases. First, we consider a portfolio consisting only of linear amortizing loans having the same maturity T. The principal outstanding of the portfolio g(t) is therefore reduced by exactly the same amount in each period. g(t) is therefore decreasing linearly over time. Because the amortizing profile is linear, this corresponds to the case where the ratio between the weighted average life (WAL) of the portfolio and its weighted average maturity (WAM) equals 0.5.

Secondly, we consider a case where the collateral pool consists only of bullet assets having the same maturity and where no defaults occur. For these types of pools, the function g(t) takes also a very simple form. Namely, g(t)

[4]For comparison purposes, we have assumed that both portfolios consist of assets distributed in 5 different homogeneous ratings R with p_R, and RR_R, homogeneous within the rating, but heterogeneous across ratings. The concentrated portfolio (HHI = 0.1) has been simulated by considering 10 loans spread across the 5 ratings, while the granular portfolio consists of 100 loans spread across the 5 ratings.

equals 100% until the assets' maturity date. At the date of maturity, all assets repay their outstanding collateral, and g(T) equals 0. In this case, the WAM and the WAL equal to T. Fig 5 show schematically the amortization profile of both portfolios.

Next, we consider the same two portfolios, but where all loans default (this is the defaulted loan portfolio). We consider as well another function, k(t) the default profile that for each date t gives the proportion of the portfolio that has defaulted. Further, it is assumed that bullet loans only default when they have to service their debt, or in other words, at their maturity. Hence, for bullet loans, the size of the defaulted collateral k(t) of the defaulted portfolio equals 0% until maturity T, where it equals 100%. For linear amortizing loans, we assume that the size of the defaulted portfolio also increases linearly. The default profile k(t) is therefore a function that increases linearly with time and that equals 100% when $t = T$.

Now, we have assumptions for the amortizing and default profile of portfolios consisting of bullets and linear amortizing loans. Those cases correspond to a ratio between the WAL and the WAM of 0.5 and 1 respectively. Further assumptions are made for intermediary portfolios. The portfolio amortizing profile g(t) is assumed to be split into two linear sections with a different slope. The higher the proportion of bullet collateral (or the closer the ratio between the WAL and the WAM of the underlying portfolio to 1), the longer and less steep the first section and the shorter and steeper the second section(see fig 6). The reason for this choice is that, with a great proportion of bullet collateral, before maturity, only a small proportion of the collateral is repaid. When the ratio between WAL and WAM reaches 1, the first section size is equal to the WAM and its slope is equal to 0 (the size of the outstanding collateral remains stable until maturity).

The exact point where the amortizing profile is split in two must be chosen in such a way that the weighted average life of the portfolio deducted from function $g(t)$ equals the weighted average life of the considered portfolio. This point is located on the following diagonal: $f(t) := \frac{1}{WAM}t$ and the exact point where the amortizing profile is split in two equals to $t = WAL$ and $g(t) = \frac{WAL}{WAM}$. The reader can easily check that for a linear amortizing profile the function $g(t)$ is split at the point where 50% of the collateral has been amortized and consequently, both segments have exactly the same slope and size.

The same logic is kept for the default profile of portfolios but reversed. The size of the defaulted portfolio is increasing over time and hence the two linear portions have a positive slope. The point where the default profile is split in two is located on the diagonal $f(t) = -\frac{1}{WAL}t + 100\%$ and the exact point where this happens is located at $t = WAL$ and $g(t) = 100\% - \frac{WAL}{WAM}t$.

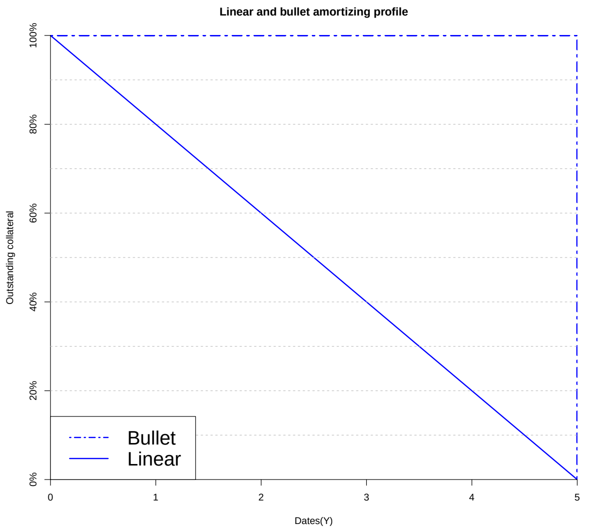


Figure 4: Amortization profile g(t) of two portfolios. Portfolio 1 consists only of bullet loans and portfolio 2 solely consists of linear amortizing loans. All maturities equal to 5.

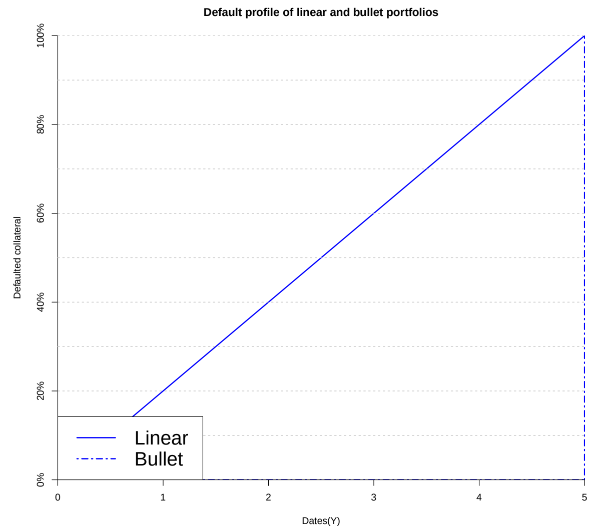


Figure 5: Default profile k(t) of two portfolios. Portfolio 1 consists only of bullet loans and portfolio 2 of linear loans with maturity equaling to 5 years.

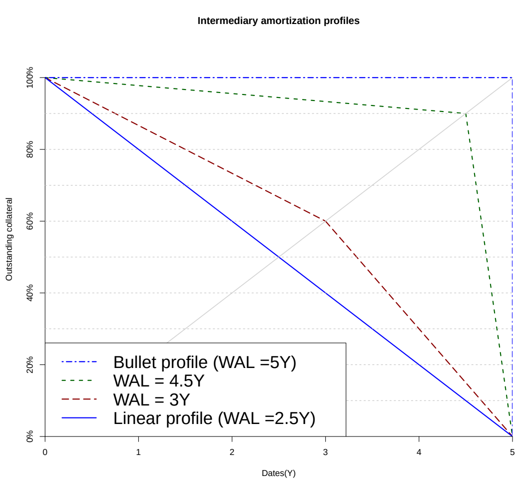


Figure 6: Amortization profile g(t) of 2 intermediary portfolios with same maturity equaling to 5. The portfolio with WAL equaling to 3 is close to a linear amortization profile whereas the portfolio with WAL equaling to 4.5 is close to a bullet amortization profile.

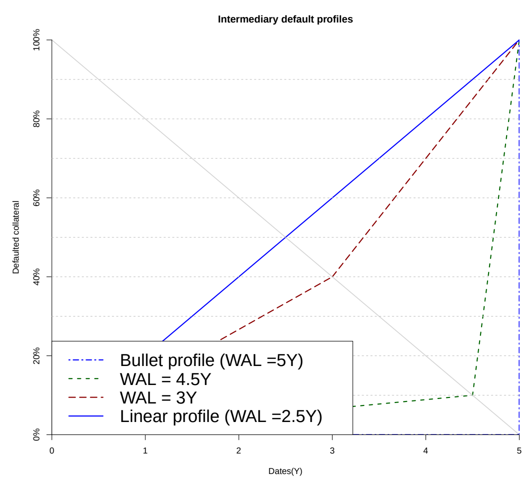


Figure 7: Default profile k(t) of 2 intermediary portfolios. Similar to the amortization profile, a portfolio with WAL equaling to 3 is close to a linear default profile while a portfolio with WAL equaling to 4.5 is close to bullet.

8.4.1 Performing sub-portfolio

The analytical expressions that give the amortization profile g(t) of the performing sub-portfolio according to the assumptions about the amortization profile can be easily found by solving the set of equations:

$$g(t) \equiv \begin{cases} g(WAM) & = 0 \\ g(WAL) & = \frac{WAL}{WAM} \\ g(0) & = 100\% \end{cases} \tag{8.2}$$

which yields:

$$y = g(t) \equiv \begin{cases} \alpha t + 1 & 0 < t < WAL \\ -\frac{WAM}{\beta} t + \beta & WAL \leq t \leq WAM \end{cases} \tag{8.3}$$

where $\beta = \frac{WAL}{WAM-WAL}$ and $\alpha = \frac{1}{WAM} - \frac{1}{WAL}$. This function can be reversed to find the date at which a specific proportion y of the notional is still outstanding. This yields:

$$g^{-1}(y) \equiv \begin{cases} \frac{-WAL}{\theta}(y-1) & \gamma \leq y \leq 1 \\ \frac{WAL-WAM}{\gamma} y + WAM & 0 \leq y \leq \gamma \end{cases} \tag{8.4}$$

Where $\gamma = \frac{WAL}{WAM}$ and $\theta = 1 - \gamma$, and $g^{-1}(y)$ gives the date where $y\%$ of the collateral is still outstanding. $g^{-1}(y)$ enables to find the weighted average time of principal repayments that the performing sub-portfolio pays to a tranche w. The detailed calculations are in appendix **??**, but this weighted average time WAL_{pfw} equals to:

$$WAL_{pfw} \equiv \begin{cases} \frac{g^{-1}(A_w)+g^{-1}(D_w)}{2} & A_w \leq D_w \leq \gamma \\ \frac{g^{-1}(A_w)+g^{-1}(D_w)}{2} & \gamma \leq A_w \leq D_w \\ \frac{(1-\psi)}{2}(WAL + g^{-1}(A)) + \frac{\psi}{2}(WAL + g^{-1}(D_w)) & A_w \leq \gamma \leq D_w \end{cases} \tag{8.5}$$

where $\psi = \frac{D_w - \gamma}{D_w - A_w}$. The main logic for finding expression (8.5) is that the first payment to a tranche w happens when more senior tranches have been repaid and therefore equals to $g^{-1}(D_w)$. Furthermore, when $g^{-1}(D_w)$ and $g^{-1}(A_w)$ lie on the same linear segment, the weighted average time of payments equal to the average of the time of the first payment $g^{-1}(D_w)$ and of the last payment $g^{-1}(A_w)$, which equals to $\frac{g^{-1}(A_w)+g^{-1}(D_w)}{2}$.

8.4.2 Defaulted sub-portfolio

In an analogous manner as the performing sub-portfolio, the analytical expression of the default profile k(t) of the defaulted sub-portfolio can be found by solving the system of equations $k(0) = 0$, $k(WAL) = 1 - \frac{WAL}{WAM}$ and $k(WAM) = 1$. This yields the following expression for the default profile:

$$y = k(t) \equiv \begin{cases} \frac{\theta}{WAL} t & 0 < t < WAL \\ \frac{\gamma}{WAM-WAL} t + 1 - \frac{1}{\theta} & WAL \leq t \leq WAM \end{cases} \tag{8.6}$$

where $\theta = \frac{WAM-WAL}{WAM}$ and y is the total proportion of collateral that defaulted before t. Similarly, the function can be reversed to find the dates t where a proportion y has defaulted. This yields:

$$k^{-1}(y) \equiv \begin{cases} \frac{WAL}{\theta}(y) & 0 \leq y \leq \theta \\ \frac{WAM-WAL}{\gamma} y + WAM(2 - \frac{1}{\gamma}) & \theta \leq y \leq 1 \end{cases} \tag{8.7}$$

A similar reasoning as for the performing sub-portfolio can be followed to find the contribution WAL_{pfw} to the WAL_w of the defaulted sub-portfolio.

$$WAL_{dfw} \equiv \begin{cases} \frac{k^{-1}(A_w)+k^{-1}(D_w)}{2} & A_w \leq D_w \leq \theta \\ \frac{k^{-1}(A_w)+k^{-1}(D_w)}{2} & \theta \leq A_w \leq D_w \\ \frac{\delta}{2}(WAL_{pf} + k^{-1}(A_w)) + \frac{(1-\delta)}{2}(WAL_{pf} + k^{-1}(D_w)) & A_w \leq \theta \leq D_w \end{cases} \tag{8.8}$$

Where $\delta = \frac{\theta - A_w}{D_w - A_w}$.

8.4.3 Overall

To find the WAL of tranches, the weighted average times of the payments of both portfolios to the tranches are weighted by their respective proportion. The proportion of the defaulted portfolio equals DR, the default rate of the portfolio. The proportion of the performing sub-portfolio equals the survival rate $1 - DR$. The default rate is a random variable that can vary between 0% (no loans have defaulted) and 100% (all loans default).

$$WAL_w = (1 - DR)WAL_{pfw} + (DR)WAL_{dfw} \tag{8.9}$$

The distribution of DR, the portfolio default rate can be found with the help of the credit risk models presented in section ??. Nevertheless, knowing the expectation of DR is enough to derive the expected WAL of the tranches. The expected WAL can indeed be written:

$$\mathbb{E}\left[WAL_w\right] = \int_0^1 [(1 - DR)WAL_{wpf} + (DR)WAL_{wdf}]h(DR)dDR \tag{8.10}$$

Where $h(DR)$ is the density function of DR. This simplifies to:

$$\boxed{\mathbb{E}\left[WAL_w\right] = (1 - \mathbb{E}[DR])WAL_{pfw} + \mathbb{E}[DR]WAL_{dfw}}$$

The main advantage of the assumptions of this section is that the WAL_w of the different tranches of the portfolio depends only on few parameters of the underlying portfolio and have a closed-form expression. Nevertheless, it has to be stressed that the amortization and default profile have been assumed and that there is no mathematical guarantee that all considered portfolios can be correctly modeled with the help of the equations introduced in this section. However, most of the cases encountered in practice are close enough to the aforementioned assumptions.

8.5 Performance reformulation

	Running time (seconds)	Number of iterations	Obj value (rating senior tranche)
Derivative-free	2230 s	300 000	8,243
Derivative-free	802 s	100 000	8,25
Derivative-free	10s	1000	9,9
Gaussian quadrature	**104 s**	**56**	**7,93**

Table 1: Performance comparison of the Gaussian quadrature technique($K = 256$) and a traditional derivative-free algorithm. The reformulation with a Gaussian quadrature outperforms derivative-free optimization techniques, both on running time and objective-value reached. The problem is to find the optimal weights Ω_R^* of the benchmark problem ??.Final objective values of solutions are calculated with formula (??)

8.6 Reformulation

To rewrite the problem of minimizing the expected losses, we start by considering the following set:

- w in (1,..,W), the number of tranches.
- k in (1,..,K), the number of quadrature nodes.
- r in (1,..,R), the number of clusters.

We have also following parameters:

- λ_{rk}, the conditional expected loss of 1 unit of collateral of cluster r when the common factor M equals to the node M_k. It is calculated as follows: $p_{r|M_k}(1 - RR_r)$ when $M = M_k$
- S: the total notional that has to be selected.
- A_{aw}: the absolute attachment point of tranche w. For notational simplicity, we denote by A_{as}, the attachment point of the most senior tranche. $A_{aw} = SA_w$, where A_w lies in the unit interval.

- D_{aw}: the absolute detachment point of tranche w.
- O_r: the total collateral of initial portfolio allocated to the cluster r.
- m_k: the weight k of the Gaussian quadrature.

Furthermore, we have the following variables:

- ω_r, total collateral of cluster r selected.
- L_k, the conditional EL of the portfolio evaluated at node k.
- TL_{wk}, the conditional EL of tranche w when the conditional factor equals to M_k.
- $\mathbb{E}[TL_w]$, the EL of tranche w. For notational simplicity, we note $E[TL_s]$ the EL of the senior tranche.

The problem of minimizing the EL of the senior tranche is therefore[5]:

$$
\begin{aligned}
\min \frac{\mathbb{E}[TL_s]}{(D_{as} - A_{as})} \\
\text{Subject to} \\
\forall k : \sum_r \omega_r \lambda_{rk} = L_k \ (a) \\
\forall k, w : \min(\max((L_k - A_{aw}, 0), D_{aw} - A_{aw}) = TL_{wk} \ (b) \\
\forall w : \sum m_k TL_{wk} = \mathbb{E}[TL_w] \ (c) \\
\sum \omega_r = S \ (d) \\
\forall r : \omega_r < O_r \ (e)
\end{aligned}
\tag{8.11}
$$

Constraints a, b, and c are not constraints that a portfolio manager faces, but are the approximations of the portfolio conditional losses (a), the conditional tranche losses (b) and expected losses (c). Constraint d imposes that the collateral S is selected. Constraint (e) prevents from selecting more collateral than assigned to cluster R. The only non-linear constraint is constraint (b), but constraint b can be transformed into a series of linear constraints with the introduction of slack variables.

To do so, we start by noting the following equivalence:

$$
\min(\max(L_k - A_w; 0); D_w - A_w) \equiv \begin{cases} 0 & L_k \leq A_w \\ L_k - A_w & A_w \leq L_k \leq D_w \\ D_w & D_w \leq L_k \end{cases}
\tag{8.12}
$$

For each tranche w and for each evaluation k, we introduce three binary variables that takes 1 only if:

- $Z_{1kw} = 1$ if $L_k \leq A_w$
- $Z_{2wk} = 1$ if $A_w \leq L_k \leq D_w$
- $Z_{3wk} = 1$ if $D_w \leq L_k$

and equal to 0 otherwise. To model the three previously mentioned conditions on the values of Z_{1kw}, Z_{2kw}, Z_{3kw} with linear constraints, we use :

- $\forall (k, w), Z_{1kw} + Z_{2kw} + Z_{3kw} = 1$

[5]For notational simplicity, we have excluded the industry and country constraints, which are written as linear functions of the weights Ω_R

- $\forall(k, w), L_k \leq A_w + (Z_{2w} + Z_{3w})M$
- $\forall(k, w), L_k \geq D_w - (Z_{1w} + Z_{2w})M$
- $\forall(k, w), L_k \geq A_w - Z_{1w}M$
- $\forall(k, w), L_k \leq D_w + Z_{3w}M$

Where M is a sufficiently high number, 8.12 can therefore be rewritten:

$$\min(\max(L_k - A_w; 0); D_w - A_w) = Z_{2kw}(L_k - A_w) + Z_{3kw}(D_w - A_w) \tag{8.13}$$

Which is only bi-linear in following term: $Z_{2kw}(L_k)$. To transform the bi-linear term, we use a Gloover linearization scheme: Glover (1975). We therefore introduce a new variable, T_{wk} that satisfies the following constraints:

$$T_{wk} \equiv \begin{cases} L_k & if\ Z_{2kw} = 1 \\ 0 & if\ Z_{2kw} = 0 \end{cases} \tag{8.14}$$

To model this with linear integer constraints : we use the the big M method. :

- $\forall T_{wk} \leq Z_{2kw}M$ (A)
- $\forall T_{wk} \leq L_k$ (B)
- $\forall T_{wk} \geq L_k - M(1 - Z_{2wk})$ (C)

Where M is a sufficiently high number. The reader can check that the aforementioned constraints imply that, when $Z_{2w} = 0$, $T_{kw} = 0$ and that when $Z_{2kw} = 1$, $T_{kw} = L_{kw}$.

8.7 Appendix data

The generated portfolios vary across the following 4 dimensions. All combinations of the different cases have been used.

1. Total initial outstanding amount from which to choose: We have considered 4 cases varying from 250M€ to 5B€;
2. Distribution of 1 year Probability of Default: We have considered 5 different cases varying in homogeneity and values;
3. Distribution of Recovery Rates: 5 different cases with varying degrees of homogeneity have been considered;
4. Distribution of the exposures to single obligor: 3 different cases have been considered;
5. Distribution of maturities: 4 different cases have been considered;

Portfolio denominations Not all tests have been performed on the whole database. To identify the distribution of the loan features of each test, a structured nomenclature is given to each portfolio. The nomenclature works as follows:

- A letter has been given to each loan feature (see table 2).
- Each column of the tables displaying one of the distributions of loan-features tested is numbered.
- A specific portfolio is specified by a code, where the letter of each loan feature is followed by the column number of the distribution of the loan feature.

For instance, O0P1R2E1M2 is the portfolio with initial outstanding equaling to column number 0, distribution of one year probabilities of default distributed according to column 1 of default probability table, recovery rates distributed as column 2 of recovery rate table, exposures distributed as column 1 of exposure table and maturities distributed as column 2 of maturities tables.

Nomenclature	
Loan feature	**Code**
Initial Outstanding	O
One year Probabilities of Default	P
Recovery Rates	R
Exposures	E
Maturities	M

Table 2: Letters for denominating each portfolio of the database

Additionally, because the portfolios found by the optimization algorithm do not have the guarantee to be the global optimal solutions, they are referred to as simplified optimal portfolios.

8.7.1 Distributions of data

In this section, the distribution of the various loan features we used to construct the portfolios of our database is provided. Each column has a column number that can be used to identify the distribution of the loan-feature of a specific portfolio. All tables can be read according to the same logic. Here is an example on how to read the table of one-year probabilities of default of a portfolio with a name of the form : O...E...P3R..M...

- 10% of loans have a PD_{1y} of 0.5%;
- 10% of loans have a PD_{1y} of 1%;
- 10% of loans have a PD_{1y} of 1.50%;
- 10% of loans have a PD_{1y} of 2%;
- 10% of loans have a PD_{1y} of 2.5%;
- 10% of loans have a PD_{1y} of 7%;
- 10% of loans have a PD_{1y} of 7.5%;
- 10% of loans have a PD_{1y} of 8%;
- 10% of loans have a PD_{1y} of 8.5%;
- 10% of loans have a PD_{1y} of 9%

	Distribution of PD1Y				
Col number	*0*	*1*	*2*	*3*	*4*
PD1Y					
0.50%	0.00%	20%	0%	10%	0%
1.00%	0.00%	20%	0%	10%	0%
1.50%	0.00%	20%	0%	10%	0%
2.00%	10.00%	20%	0%	10%	0%
2.50%	10.00%	20%	0%	10%	0%
3.00%	10.00%	0%	0%	0%	0%
3.50%	10.00%	0%	0%	0%	5%
4.00%	10.00%	0%	0%	0%	5%
4.50%	10.00%	0%	0%	0%	5%
5.00%	10.00%	0%	0%	0%	5%
5.50%	10.00%	0%	0%	0%	15%
6.00%	10.00%	0%	0%	0%	20%
6.50%	10.00%	0%	0%	0%	15%
7.00%	0.00%	0%	20%	10%	15%
7.50%	0.00%	0%	20%	10%	10%
8.00%	0.00%	0%	20%	10%	5%
8.50%	0.00%	0%	20%	10%	0%
9.00%	0.00%	0%	20%	10%	0%
9.50%	0.00%	0%	0%	0%	0%
10.00%	0.00%	0%	0%	0%	0%

Table 3: Table of different cases of one year probabilities of default considered

	Recovery rates				
Col number	*0*	*1*	*2*	*3*	*4*
Recovery rate					
0%	0%	0%	0%	0%	0%
5%	0%	25%	0%	13%	0%
10%	0%	25%	0%	13%	5%
15%	0%	25%	0%	13%	10%
20%	10%	25%	0%	13%	15%
25%	10%	0%	0%	0%	15%
30%	10%	0%	0%	0%	20%
35%	10%	0%	0%	0%	15%
40%	10%	0%	0%	0%	5%
45%	10%	0%	0%	0%	5%
50%	10%	0%	0%	0%	5%
55%	10%	0%	0%	0%	5%
60%	10%	0%	25%	13%	0%
65%	10%	0%	25%	13%	0%
70%	0%	0%	25%	13%	0%
75%	0%	0%	25%	13%	0%
80%	0%	0%	0%	0%	0%
85%	0%	0%	0%	0%	0%
90%	0%	0%	0%	0%	0%
95%	0%	0%	0%	0%	0%

Table 4: Table of different recovery rates considered

	Maturity			
Col number	*0*	*1*	*2*	*3*
Maturity				
36	0%	0%	30%	5%
48	0%	0%	20%	5%
60	0%	0%	0%	5%
72	0%	0%	20%	5%
84	0%	0%	30%	5%
96	0%	0%	0%	5%
36	0%	0%	0%	5%
48	25%	0%	0%	5%
60	25%	0%	0%	5%
72	25%	0%	0%	5%
84	25%	0%	0%	5%
96	0%	0%	0%	5%
24	0%	0%	0%	5%
36	0%	25%	0%	5%
48	0%	25%	0%	5%
60	0%	25%	0%	5%
72	0%	25%	0%	5%
84	0%	0%	0%	5%
24	0%	0%	0%	5%
24	0%	0%	0%	5%

Table 5: Table of different cases of maturities considered

	Exposures to single obligor		
Col number	*0*	*1*	*2*
Exposure (K€)			
50	0%	0%	0%
100	0%	0%	0%
150	0%	0%	0%
200	0%	0%	0%
250	0%	10%	20%
350	10%	15%	15%
450	30%	15%	15%
500	30%	15%	10%
600	15%	10%	10%
750	15%	15%	0%
900	0%	5%	0%
1100	0%	5%	0%
1300	0%	10%	0%
1500	0%	0%	0%
2500	0%	0%	0%
4000	0%	0%	30%
200	0%	0%	0%
200	0%	0%	0%
200	0%	0%	0%
200	0%	0%	0%

Table 6: Table with the different cases of exposures considered

Total outstanding collateral (O)				
Col number	0	1	2	3
Total outstanding collateral (€)	250,000	50,000	500,000	5,000,000
Number of loans (approximately)	500	100	1000	1,0000

Table 7: Table with the different sizes of portfolios considered

8.8 Numerical integration

In this section, further information on how the algorithm has been numerically implemented is provided. The algorithm was implemented in the statistical programming language R. All mixed-integer linear programs are solved with the academic version of Gurobi's solver[6]. The Nelder-Mead optimization solver used is the *neldermead* function of the open-source NLOPTR R package. All tests have been performed on a personal computer with processor Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz, and 12Gb of RAM.

8.8.1 Clustering

The first step of the algorithm is to cluster the loans in several distinct clusters with homogeneous loan features. To do so, the data is first standardized. Two standardization techniques commonly used for K-mean clustering algorithms (Alpaydin, 2020) have been tried:

- Z-standardisation, the variables are re-scaled by subtracting their mean and divide the result by their standard deviation.
- Range standardization, this technique subtracts the variable with its minimum, and divide the result with difference between the maximum and the minimum.

[6]The code can be easily adapted to a variety of open-source solvers such as RGLPK, even though those solvers are slower.

For small portfolios (typically less than 100), reducing the problem to a limited number of clusters is not necessary. For those portfolios, each loan is assumed to be the center of a cluster, comprising only this single loan. Therefore, for those portfolios, the binary constraint is relaxed, and transformed into a box constraint, namely it is possible to select between 0 and the notional of each loan.

Once the different loan features that are used for the clustering have been standardized, clusters are found by using the *kmeans* function of R *stats* package. The average loan features of each cluster are calculated and the loans are assigned to a cluster-specific data frame.

8.8.2 Optimization

The previous step yields a matrix with relevant loan features of each cluster. The first step is to generate the conditional loss matrix Λ_{rk}, that for each evaluation M_k of the quadrature, gives the conditional loss λ_r of cluster R. To find the points M_k and weights m_k of the quadrature, the function *gaus.quad.prog* of R package *statmod* is used. Subsequently, λ_{rk} of each cluster is calculated with formula (4.2) [7].

All mixed-integer linear programs are solved with *Gurobi* mixed-integer linear solver. All constraints of problem (8.11) and the objective function are provided to the solver in Matrix format. Except for the artificial WAL and WAM constraints (5.1) used to derive the optimal region, the constraint matrix is built before entering in the optimization routine.

Function 1 creates an additional constraint matrix with the tested WAL and WAM, and merges the additional matrix with the previously built constraint matrix. Subsequently, *Gurobi* solver is called and the rating of the portfolio with found weights is calculated and returned.

Function 1 is provided as the cost function to *neldermead* function of R package *NLOPTR*. The neldermead function finds the optimal WAL^* and WAM^*. Those quantities are used to solve the mixed integer linear problem a last time, which yields the vector of optimal weights Ω^*_R.

8.8.3 Rounding of the weights

The found weights Ω_R of each cluster must be rounded to accommodate the integers constraints. For clusters where the constraint has been violated, the two available bounds between which the ω^*_r lies must be found. This can be done efficiently by solving two small-scale binary linear programs.

The first one is to maximize the amount selected of the cluster, while not exceeding the value found by the solver. The second one is to minimize the amount selected by the cluster, under the constraint of selecting at least the value found by the optimization program. Those two programs are solved with *Gurobi* as well. Even though those two programs are binary constrained, their small size enables to find the two bounds quickly (typically less than 0.2 second). For each cluster where the integer constraint is violated, two bounds ω_{rl} (the lowest) and ω_{rh} (the highest) are found.

Once those two bounds have been found, the next step is to choose to which bound the clusters have to be rounded. To do so, following procedure is applied :

1. Each cluster value where an integer constraint has been violated is initialized to the lowest bound ω_{rl}
2. The cluster with largest ω_{rl} is set at his highest bound ω_{rh}.Subsequently, the other clusters' weights ω_r are set to their highest bound, in order of size, until the point where the sum of rounded weights reaches the constraint of minimum size.

[7] Because we are interested in the conditional loss of 1 unit of cluster R, O_i in formula 4.2 equals to 1

8.8.4 Matching the weights

The last part is to match the rounded weights Ω_R^{*r} with a binary vector $\Omega = (\omega_i, ..\omega_n)$ where ω_i equals to 1 if loan i is selected. To choose in priority in each cluster the loans according to specific criteria, a last binary linear problem is solved for each cluster. The objective is to minimize the weighted average of the criteria under the constraint to reach ω_r selected in the cluster. Again, this is done with *Gurobi* solver. The small size of problems considered allows for each cluster to find a solution in less than a second.

8.9 Benchmark problem

The portfolio manager must select 60% of the initial collateral while minimizing the rating of the senior tranche of a CDO consisting of 3 different tranches. Attachment points are typically chosen depending on the riskiness of considered portfolios. Hence, for each portfolio in our database, we vary the attachment points according to the initial pool EL ($\mathbb{E}\left[L_{pf,in}\right]$)*. Following attachment points are considered :* $A = \left(0, \frac{\mathbb{E}[L_{pf,in}]}{2}, \mathbb{E}[L_{pf,in}]\right)$*. The senior tranche attachment and detachment points are consequently* $A_s = \mathbb{E}[L_{pf,in}]$ *and* $D_s = 1$*. To make a comparison with random solutions possible, no additional constraints are considered.*

8.10 Random solutions results

		Rating of senior tranche and portfolio			Difference between benchmark and optimization	
		Optimization	**Mean Random**	**Best result random**	**Average improvement**	**improvement compared to best random result**
Senior tranche	Average	4,625	8,588	7,85	3,963	3,225
	Portfolio smallest efficiency solver (O1E2P0R4M1)	6.94	8.35	6.83	1.41	-0,11
Final portfolio	Average	10,59	13,49	12,96189	2,90	2,37189
	Portfolio smallest efficiency solver (O1E2P0R4M1)	12.34	13.43	12.25	1,09	-0,09

Table 8: Comparison between simplified optimal solutions and random solutions. On average, the best random solution is outperformed by 3.225 by the simplified optimal solution. Rating is calculated as in appendix **??**. Portfolios of simplified optimal solutions have a much higher credit quality than benchmark solutions (32 nodes were chosen for the quadrature).

		Rating of senior tranche and portfolio			Difference between benchmark and optimization	
		Optimization	**Mean Random**	**Best result random**	**Average improvement**	**Improvement compared to best random result**
Senior tranche	Average of portfolios (O1E2...)	4,6456	7,574	6,106	2,928	1,4599
	Worst portfolio (O1E2P0R4M1)	6,94	8,35	6,83	1,41	-0,11

Table 9: Rating of senior tranche of simplified optimal solutions compared to random benchmarks for small initial portfolios with exposures highly concentrated in small amount of loans. The problem considered is problem **??**. The simplified optimal solutions do not yield significant better results.

8.11 Heuristics

Portfolio	Rating senior tranche Simplified optimal portfolios	Heuristic
O2E0P0R3M2	5,586	5,956
O2E0P1R1M0	5,874	5,876
O2E0P1R3M2	4,751	4,650
O2E0P2R1M0	11,776	11,760
O2E0P2R3M2	5,329	5,566
O2E0P3R1M0	6,347	6,233
O2E0P3R3M2	5,836	5,095
O2E0P4R1M0	10,412	10,394
O2E0P4R3M2	5,668	6,001
O2E1P0R1M0	8,779	8,759
O2E1P0R3M2	6,101	6,315

Table 10: Rating of senior tranche of simplified optimal and benchmark portfolios (heuristic) for problem **??**. Both ratings were calculated with a LHP approximation (see appendix **??**. The benchmark portfolios were constructed by taking the 60% with smallest expected loss.

8.12 Effect parameters

8.12.1 Standardization method

We tested 150 clusters on 10 different portfolios of 10 000 loans. For each portfolio, the algorithm is launched two times. One time with the range-standardization, and one time with the Z standardization. Subsequently the reached objective value is computed for both cases with the homogeneous large approximation method. Problem solved is **??** with correlation parameter $\rho = 10\%$. Our initial hypothesis is that both techniques would yield similar results and none would be preferred. Testing the effect of standardization technique yields :

- For most portfolios, both techniques indeed lead towards similar results and no technique systematically outperforms.
- For one portfolio (O3E0P0R1M0), the sensitivity to this parameter seems more pronounced and changing from Z-standardisation to range standardization improves the rating of the senior tranche by more than 0.6.
- Hence, our hypothesis seems only partly valid. For most portfolios the effect is limited, but in some specific cases changing standardization techniques might improve results.

Portfolio	Z-standardisation	Range-standardisation
O2E2P4R4M3	5.02	5.06
O3E0P0R0M0	3.87	3.86
O3E0P0R0M1	4.68	4.74
O3E0P0R0M2	3.95	3.90
O3E0P0R0M3	6.25	6.25
O3E0P0R1M0	5.37	4.80
O3E0P0R1M1	5.55	5.55
O3E0P0R1M2	4.92	4.90
O3E0P0R1M3	3.60	3.58
O3E0P0R2M0	2.60	2.95

Table 11: Rating of the most senior tranche of simplified optimal solutions for two different standardisation techniques. The problem solved is benchmark problem **??** with correlation parameter ρ equaling to 10%

8.12.2 Number of nodes

In this section, the impact of the number of nodes K chosen for the Gaussian quadrature is studied.

The number of evaluations K of a Gaussian quadrature typically increases the precision of the approximation, if specific conditions are met. However, introducing additional quadrature points increases the number of slack variables that have to be introduced to solve the mixed-integer linear problem 8.11 and therefore the running time. In this section, we experiment how much K nodes are needed for finding reliable results.

To do so, for 10 different portfolios the optimization routine is started. For each portfolio, the problem is solved 9 times with quadrature points ranging from 4, for the first iteration, to 356 for the last iteration. The number of clusters chosen is equal to 150. Then, we calculate the correspondence rate of each solution with the solution found with the maximal number of nodes. The correspondence rate of an iteration with K nodes is calculated as follows: the number of loans that were chosen in the considered iteration that are also chosen in the iteration with the most number of nodes is divided with the total number of loans selected in considered iteration. By construction, it equals to 100% for $K = 356$. Furthermore, the corresponding metric of the most senior tranche is also computed for each iteration according to the homogeneous large pool approximation.

We expect the solution found to change for the first evaluations, with relatively few nodes, as the approximation would not be precise enough. Once enough precision is reached, we expect the found solution not to change anymore and therefore having a convergence rate equaling to 100% when enough precision is reached, making additional nodes unnecessary. Doing this yields:

		Number of nodes								
Portfolio		4	8	16	32	64	128	256	346	356
O2E0P0R2M1	Correspondence rate	92%	97%	99%	92%	92%	92%	100%	100%	100%
	Senior tranche metric	3,89	3,85	3,82	3,90	3,91	3,90	3,82	3,82	3,82
O2E0P0R4M3	Correspondence rate	95%	98%	100%	99%	100%	100%	100%	97%	100%
	Senior tranche metric	5,76	5,76	5,72	5,69	5,72	5,72	5,72	5,66	5,73
O2E0P1R2M1	Correspondence rate	100%	100%	80%	100%	100%	100%	100%	100%	100%
	Senior tranche metric	4,34	4,34	3,98	4,34	4,34	4,34	4,34	4,34	4,34
O2E0P1R4M3	Correspondence rate	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Senior tranche metric	7,17	7,17	7,17	7,17	7,17	7,17	7,17	7,17	7,17
O2E0P2R2M1	Correspondence rate	92%	100%	100%	100%	100%	100%	92%	92%	100%
	Senior tranche metric	2,05	2,06	2,06	2,06	2,06	2,06	2,05	2,05	2,06
O2E0P2R4M3	Correspondence rate	68%	100%	100%	72%	71%	100%	100%	100%	100%
	Senior tranche metric	3,67	4,62	4,62	3,55	3,49	4,62	4,62	4,62	4,62
O2E0P3R2M1	Correspondence rate	66%	66%	66%	66%	66%	66%	100%	100%	100%
	Senior tranche metric	5,41	5,41	5,41	5,41	5,41	5,41	5,16	5,16	5,16
O2E0P3R4M3	Correspondence rate	100%	97%	97%	97%	97%	97%	97%	100%	100%
	Senior tranche metric	7,37	7,23	7,23	7,23	7,23	7,23	7,23	7,37	7,37
O2E0P4R2M1	Correspondence rate	92%	100%	100%	92%	100%	92%	100%	100%	100%
	Senior tranche metric	2,08	2,56	2,56	2,05	2,56	2,05	2,56	2,56	2,56
O2E0P4R4M3	Correspondence rate	97%	99%	99%	99%	99%	99%	99%	100%	100%
	Senior tranche metric	3,74	3,43	3,44	3,43	3,43	3,43	3,43	3,4	3,4

Table 12: Correspondence rate of each solution with the solution when $K = 356$ and corresponding metric of the most senior tranche. Contrarily to our hypothesis, the results do not seem to converge when additional nodes are added. Results were calculated with a Z-standardisation and correlation parameter ρ equaling to 10% .The problem used is benchmark problem **??**

.

- Except for certain specific portfolios, results do not seem to converge towards a global solution. Furthermore, when calculating the metric with the LHP approximation, adding more than 8 nodes do not seem to improve the rating of the most senior tranche systematically. On the contrary, better results are sometimes achieved with a very small number of nodes than with large values of this parameter.

- Counter-intuitively, the results are already very good with 4 nodes, despite the fact that with only 4 nodes, the estimation error of the quadrature is way larger than with additional nodes. This is probably explained by the fact that a significant bettering of the metric of the most senior tranche is achieved by choosing loans that have a small EL. Even though the objective value isn't calculated precisely with 4 nodes, it is already positively influenced by non-risky loans when only 4 nodes are chosen.
- While for most portfolios the rating does not seem overly influenced by the choice of a number of nodes (typically changing the number of nodes impacts the rating by less than +/-0.3), some portfolios appear way more sensitive to this parameter. For instance, portfolio O2E0P2R4M3 reaches a rating of 3.67 with only 4 nodes and reaches a much worse value of 4.62 with 256 nodes.
- Because no general rule can be derived regarding the optimal choice of a number of nodes, it might appear adequate to relaunch the optimization routine a certain number of times with different values for this parameter.

8.12.3 Number of clusters

Choosing the right number of clusters is delicate. Indeed, the more clusters are added the more similar the loans within a cluster will be. A high degree of similarity within the cluster leads to cluster features Y_r that are very close to the loan features of all loans within the cluster. Hence, the error of evaluating the different constraint and objective functions with the loan features of the cluster rather than with the loan features of the specific loans is small.

But on the other hand, the more clusters are added, the smaller the number of loans within the cluster. Hence, when too many clusters are chosen the difference between the optimal weights Ω_r^* and the rounded optimal weights might become important, leading to potential errors.

To assess the impact of this choice, we run the optimization routine on 10 large portfolios of 10 000 loans with cluster numbers varying between 100 and 160, and compute the metric of the senior tranche.

	Rating of the senior tranche achieved						
Number of clusters	100	110	120	130	140	150	160
Portfolio							
O2E2P4R4M3	5.62	5.65	5.59	5.68	5.58	5.60	5.64
O3E0P0R0M0	5.10	5.08	5.05	5.05	5.05	5.06	5.04
O3E0P0R0M1	4.06	4.46	4.51	4.44	3.75	3.78	3.75
O3E0P0R0M2	4.60	4.66	4.68	4.70	4.72	4.69	4.61
O3E0P0R0M3	3.91	4.01	3.98	3.92	3.97	3.92	3.96
O3E0P0R1M0	6.27	6.26	6.25	6.25	6.25	6.25	6.25
O3E0P0R1M1	5.37	4.83	4.81	5.37	4.80	5.38	4.80
O3E0P0R1M2	5.55	5.57	5.55	5.57	5.55	5.54	5.55
O3E0P0R1M3	5.03	4.94	4.96	4.91	4.92	4.97	4.91
O3E0P0R2M0	3.62	3.61	3.60	3.60	3.61	3.60	3.60

Table 13: rating of the most senior tranche achieved for a different choice of number of clusters

- 8 out of the 10 tested portfolios seem globally insensitive to the number of clusters chosen. Changing the number of clusters only impacts the rating of the most senior tranche by a small amount (typically less than 0.05 notch). However, the impact is significantly more important for 2 portfolios.
- A closer look at the loan features of those two portfolios does not allow us to identify a characteristic that would be uniquely shared by these two portfolios, explaining their greater sensitivity to the number of clusters.

8.13 Cost-function optimization

We rely on penalty functions to add constraints to the traditional derivative-free algorithm. Penalty functions are a convenient way to add constraints to non-linear problems.The idea of penalty functions is to convert a constrained problem into an unconstrained problem by adding an artificial penalty term to the objective function if the constraints are violated. The penalty term should be chosen in such a way that solutions violating the constraints are largely sub-optimal compared to feasible solutions. The benchmark problem **??** is relatively unconstrained. The only constraint that we need to add is the minimum collateral to select, which should equal to at least 60% of the outstanding amount of the initial portfolio. We introduce the penalty term C_i as follows:

$$C_i = \max\left(C \times (0.60 - \sum \Omega_r); 0\right) \tag{8.15}$$

Where C is a penalty coefficient. We choose the arbitrary value 15 for C. The values of Ω_r have been re-scaled to have $\sum_r \omega_r = 1$ if the whole initial portfolio is selected. It can be checked that C_i equals 0 if the sum of Ω_r equals at least 0.6, and hence the solution is feasible. Furthermore, a box constraint on each cluster is added. This constraints solutions to values where the weight invested in cluster R does not exceed the outstanding collateral of loans assigned to cluster R. The exact cost function of the derivative free algorithm is therefore :

$$\begin{aligned} \min_{\Omega_R} R(EL_s, WAL_s) + \max\left(C \times (0.60 - \sum \Omega_r); 0\right) \\ \text{Subject to} \\ \forall R : 0 \leq \ \omega_R \leq U_R \ (a) \end{aligned} \tag{8.16}$$

Where constraint a is the box constraint on each cluster R constraining ω_r to be below U_r, the available collateral of cluster R and EL_s and WAL_s are calculated with formula **??**. We use the Nelder- Mead algorithm to find the solution of the derivative-free formulation.

8.14 Sensitivity to estimation-errors

In this section, the sensitivity to estimation errors of the different inputs is assessed. To do so, the optimization algorithm is first launched on an initial benchmark portfolio O2E0P0R0M3 (50 clusters are chosen) on our benchmark problem **??**. The simplified optimal solution of this portfolio is used as a reference solution to evaluate the sensitivity.

Then, the value of the loan feature of the benchmark portfolio to which the sensitivity of the algorithm is tested is modified randomly according to a uniform distribution with a mean equaling to the initial value of the reference portfolio. Different bounds for the uniform distribution are tested, ranging from ±2% of the initial value to ±20%.

Subsequently, the correspondence rate is calculated. This is done by calculating the proportion of loans that have been chosen on basis of the modified value and that were selected in the reference portfolio. The higher this proportion, the less sensitive the algorithm to estimation errors. The value of the metric of the senior tranche, is also compared (and calculated on basis of the initial value). The closer the two metrics, the smaller the sensitivity to the input. To have a robust estimation of the sensitivity, the process is repeated 10 times and the average rating and correspondence rate for each error bound is computed.

In this section, the procedure applied to compute the sensitivity to estimation errors of the different inputs is explained.

To applied procedure, the optimization algorithm is first launched on an initial benchmark portfolio O2E0P0R0M3 (50 clusters are chosen) on our benchmark problem **??**. The simplified optimal solution of this portfolio is used as a reference solution to evaluate the sensitivity.

Then, the value of the loan feature of the benchmark portfolio to which the sensitivity of the algorithm is tested is modified randomly according to a uniform distribution with a mean equaling to the initial value of the reference portfolio. Different bounds for the uniform distribution are tested, ranging from ±2% of the initial value to ±20%.

Subsequently, the correspondence rate is calculated. This is done by calculating the proportion of loans that have been chosen on basis of the modified value and that were selected in the reference portfolio. The higher this proportion, the less sensitive the algorithm to estimation errors. The value of the metric of the senior tranche, is also compared (and calculated on basis of the initial value). The closer the two metrics, the smaller the sensitivity to the input. To have a robust estimation of the sensitivity, the process is repeated 10 times and the average rating and correspondence rate for each error bound is computed.

Applying the aforementioned procedure yields:

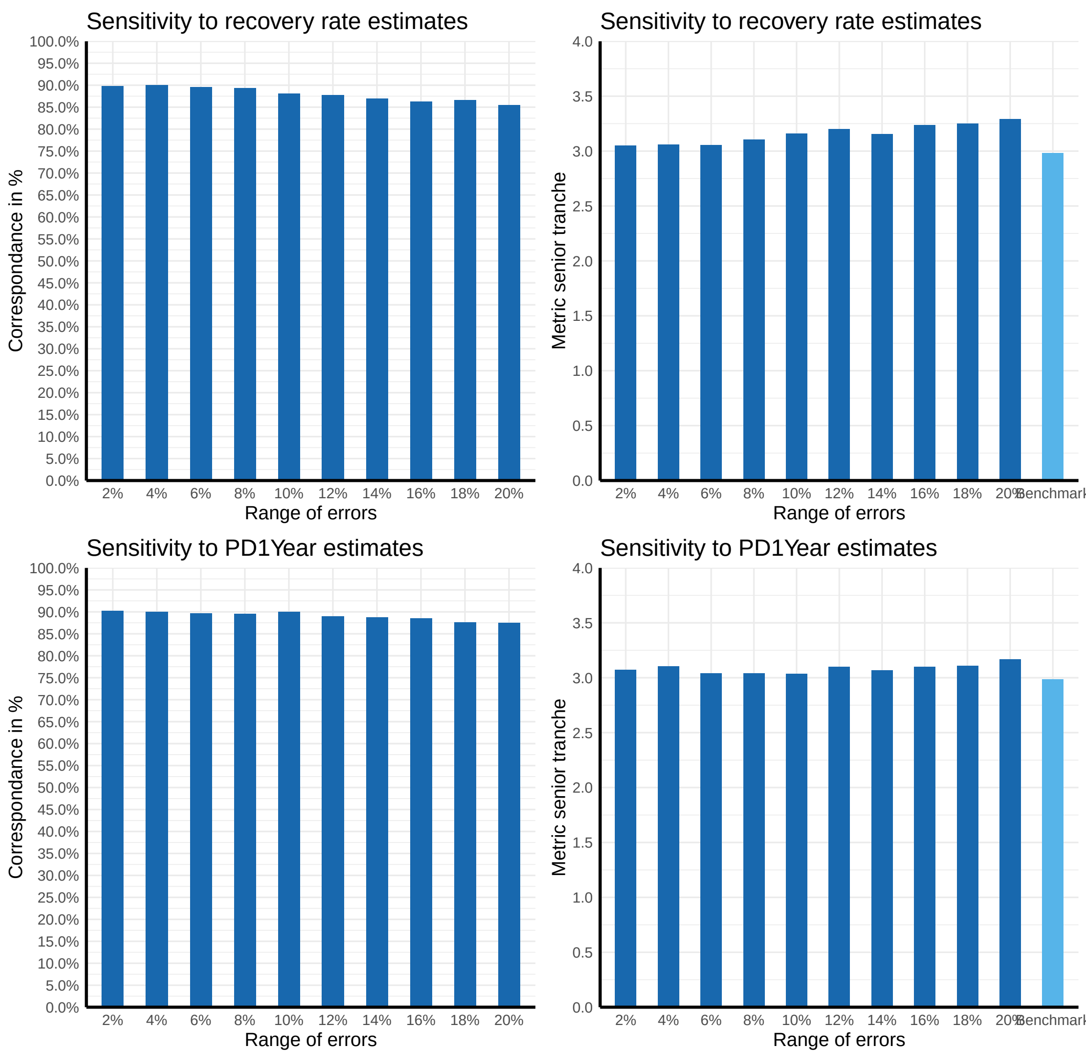


Figure 8: Correspondence rate and rating of the most senior tranche for different error bounds on recovery rates and probabilities of default. Sensitivity to recovery rate estimates is higher. The rating of the most senior tranche increases by 10% when an error bound of 20% is set on the recovery rate estimate.

References

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Burtschell, X., Gregory, J., and Laurent, J.-P. (2009). A comparative analysis of cdo pricing models under the factor copula framework. *The Journal of Derivatives*, 16(4):9–37.

Choi, J., Du, Y., and Song, Q. (2021). Inverse gaussian quadrature and finite normal-mixture approximation of the generalized hyperbolic distribution. *Journal of Computational and Applied Mathematics*, 388:113302.

Dennis Jr, J. E. and Woods, D. J. (1985). Optimization on microcomputers. the nelder-mead simplex algorithm. Technical report.

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460.

Larson, J., Menickelly, M., and Wild, S. M. (2019). Derivative-free optimization methods. *Acta Numerica*, 28:287–404.

Li, D. X. (2000). On default correlation: A copula function approach. *The Journal of Fixed Income*, 9(4):43–54.

Markowitz, H. (1952). Portfolio selection*. *The Journal of Finance*, 7(1):77–91.

Sirignano, J. A., Tsoukalas, G., and Giesecke, K. (2016). Large-scale loan portfolio selection. *Operations Research*, 64(6):1239–1255.

Slime, B. et al. (2016). Credit name concentration risk: Granularity adjustment approximation. *Journal of Financial Risk Management*, 5(04):246.

Who Are the Socially Responsible Investors?: A Machine Learning Approach

Abstract

This paper analyzes which of the investor characteristics are most important in describing the preferences for socially responsible investment (SRI). I take advantage of machine learning techniques that have a several advantages over the logit model which is a standard in empirical analysis of the discrete choice model. Based on several analysis, I find that household financial wealth, age, income, and education level are most important predictors for the preferences in SRI. Also, the results suggest that social investors have bounded willingness to forgo returns to invest prosocially.

1 Introduction

Socially Responsible Investment (SRI) has been arising as one of the most important investing factors in financial markets around the world. Investors are becoming more conscious about the decisions they make and derive utility not only from financial gains, but also from incorporating one's personal values by investing in SRI (Bollen, 2007; Nilsson, 2008; Bauer and Smeets, 2015).

In light of this, using the Dutch sample, Rossi et al. (2019) suggest that highly educated, older, female, and those who consider themselves to be not financially knowledgeable show more potential interest in SRI. Also, they find that although SR investors are willing to forgo some return to invest in SRI, when the return penalty gets bigger, they refrain from SRI.

The primary aim of this paper is to extend Rossi et al. (2019) and address what characterizes individuals who are currently investing in SR products and who are potentially interested, using a survey administered on a representative sample of the Dutch population. Specifically, I take advantage of a machine learning model, Random Forest to better explain which variables have more power in predicting the household preferences for sustainable investing.

Also, as shown in Levantesi and Piscopo (2020), I employ the interpretation tools to overcome the drawbacks of *black box* Random Forest: variable importance and partial dependence plots, to identify which variables have the most explicative power in predicting the investor preferences on SRI.

2 Literature Review

Extant body of literature on SRI has been dedicated mainly to the financial performance of SR products, and the motives and the characteristics of the SR investors. SR investors may make such a decision out of financial motives i.e. they believe SRI consistently outperforms traditional investments or want to diversify their portfolios. On the other hand, their decision may be purely out of prosocial motives in that they want to contribute to social good.

Empirical literature has been investigating the financial performance of SRI compared to traditional funds but inconclusive of the true relationship. Some empirical studies show that SR products may perform better than or on par with conventional funds. Bauer et al. (2005) reveal returns for ethical mutual funds provide are not statistically different from conventional counterparts. Derwall et al. (2005) and Kempf and Osthoff (2007) show that SRI portfolios provide higher returns than the comparable benchmark. On the other hand, others present the traditional investments have higher returns than SRI. Fabozzi et al. (2008) conclude that sin portfolio [1] outperforms comparable benchmark and suggest that SRI that screens out the sin stocks will come with economic consequences. The higher returns of sin stocks may be due to the low investment from the norm-constrained investors such as pension funds and the litigation risks (Hong and Kacperczyk, 2009).

Bollen (2007) and Nilsson (2008) find that SR investors enjoy utility from being socially responsible and consistent with their personal values in investments on top of financial gains. This is different from traditional investors who make decisions based on profit maximization. Similarly, Riedl and Smeets (2017) demonstrate that the intrinsic social preferences play a significant role in the investment decisions on SRI while socioeconomic characteristics have a minor impact. Also, they suggest that diversification is one of the motives to invest in SRI. On the contrary, Bauer and Smeets (2015) find socioeconomic profile of investors play an important role as investing in SR products is driven by one's social identity, which tends to be stronger among individ-

[1] Investing in stocks classified in the six industries of alcohol, tobacco, defense, biotech, gaming, and adult services.

uals with higher education, low financial wealth and at young age. Like wise, Rossi et al. (2019) emphasize the profile of the SR investors that they tend to be female, highly educated, and consider themselves less financially literate. Also, they reveal that SR investors are purely driven by pro-social motives, even willing to forgo financial returns to a certain degree without needing compensation such as in-kind gifts.

3 Methodology

3.1 Random Forest

Random forest trains a number of decision trees on random subset of the data called the *"bootstrapped* samples" and aggregate the results [2]. At each split, a random subset of m predictors out of the full set of p predictors are considered. This way the trees from each bootstrapped sample are less correlated, while averaging many uncorrelated trees produces more stable results. When we have B bootstrapped samples and the bth sample follows a function $\hat{f}^{*b}(\mathbf{x})$, we obtain the aggregate results of all predictions as follows:

$$\hat{f}_{bag}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(\mathbf{x}) \quad (1)$$

Random forest algorithm is known to be one of the best performing among the machine learning classification models (Zhao et al., 2020). Hence, following the structure of Levantesi and Piscopo (2020), I use Random Forest to predict the investor preferences on SRI and subsequently, to identify the profile of these investors by looking at which variables have the most predictive power.

3.2 Variable Importance

Since Random Forest aggregates the results of multiple trees, it is not possible to understand how $\hat{f}_{bag}(\mathbf{x})$ uses the data to produce the final output. However, Breiman et al. (1984) suggested *variable importance* to overcome the problem of the black-box machine learning models. Variable importance is determined by how many times a variable was selected for splitting in the tree building process and how much it improved the squared error[3].

3.3 Partial Dependence Plot

In order to understand the relationship between each predictor and the target variable in Random Forest, I use the *partial dependence plot* proposed by Friedman (2001). The partial dependence plot shows the marginal effect of particular feature(s) on the target variable.

For the 'variable of interest x_s and some function $F(\mathbf{x})$ to fit a model that obtains $\hat{y}$, the partial dependence of $F(\mathbf{x})$ on x_s is estimated from the data with Monte Carlo integration by:

$$\hat{F}_s(x_s) = \frac{1}{N}\sum_{i=1}^{N}F(x_s, \mathbf{x}_{i\backslash s}) \quad (2)$$

where the mean is over the marginal distribution of the variables in the subset $\mathbf{x}_{\backslash s}$ which is the complement of x_s, hence the entire variable set is $\mathbf{x} = (x_s, \mathbf{x}_{\backslash s})$. $\{\mathbf{x}_{i\backslash s}\}_1^N$ are $\mathbf{x}_{\backslash s}$ in the data. The function $F_s(x_s)$ reveals the influence of x_s after *averaging out* the influence of all other variables in $\mathbf{x}_{i\backslash s}$.

3.4 Random Oversampling Technique

In many cases of the empirical analysis, the class categories in the target variable are not equally represented. This class imbalance is particularly problematic when it comes to tree-based algorithms as they will automatically achieve a high predictive accuracy just by assigning all the outcomes to the majority class. There are several ways to deal with this

[2] More detailed explanation on decision trees can be found in Appendix D

[3] Exact equation for variable importance can be found in Appendix E

problem in machine learning including the use of appropriate evaluation metrics such as *precision and recall*, or *F1 score*, and employing resampling methods (Pazzani et al., 1994; Lewis and Catlett, 1994; Kubat and Matwin, 1997; Japkowicz, 2000). The data in this paper also shows imbalance in classes as the majority of the investors do not have a prior experience in SRI and do not show stated preferences for SRI. Here, I use a simple resampling method, *random oversampling* to tackle the class imbalance problem. Random oversampling randomly selects observations from the minority class with replacement and duplicate them to increse the sample size.

4 Data

The empirical analysis is based on the survey data that was conducted on a representative sample of Dutch population collected by CentERdata, Tilburg University, originally for Rossi et al. (2019). The participants are the members of CentERpanel aged above 17 and consist of more than 2000 households that provide different information to DNB Household Survey (DHS). This data contains information about general socio-demographic characteristics, employment, financial and housing wealth, health, and income. The description for the variables are shown in Table 1. The abbreviated names as in Table 1 are used in all the plots in the later sections for the sake of simplicity.

Table 1: Variable Description

Variable	Description
GEN	Gender (1: ‘male’; 2: ‘female’)
AGE	Age
EDUC	Education level: 1 (primary) to 6 (university)
FLITSUB	Subjective (self-assessed) financial literacy level: 1 (not knowledgeable) to 4 (very knowledgeable)
FLITOBJ	Objective financial literacy level: 1 (not knowledgeable) to 4 (very knowledgeable)
EMP	Employment status (0: ’unemployed’; 1: ’employed’)
PAR	Partnership status (0: ‘single’; 1: ‘not single’)
NKID	Number of children in the household
NMEM	Number of household members
LWLTH	Log (houshold financial wealth)
LINC	Log (individual income)
URB	Residence urbanity rank based on address density: 1 (very urban) to 5 (not urban)

The survey was conducted in May 2016 with the questionnaire about the respondents’ revealed and stated preferences for SRI. For revealed preference questions, individuals state whether they are currently investing in SRI (Q2), and why they invest in it if so, or why they do not invest if not (Q3). For the stated preferences questions, they are given a hypothetical inheritance, with which they need to make investment decisions in a fixed time horizon of one year. SRI options have salient return penalty compared to the traditional investment options. Also, some of them offer in-kind gifts such as a book or a voucher for cultural activities as a compensation, allowing us to identify if the SR investors need a little nudge (Rossi et al., 2019). To investigate the degree of the willingness to invest in SRI depending on the situation, the hypothetical inheritance and the return penalty are randomized (Rossi et al., 2019). In the first SP question (Q5), they choose to invest the hypothetical inheritance in a savings account at (i) a traditional bank, (ii) a bank that only invests in SR companies, which yields a lower return[4], or (iii) an SR bank similar to (ii) but with an in-kind gift of a deluxe edition book at the cost of even lower return compared to (ii)[5]. The second SP question (Q6) asks individuals whether they would want to invest in a savings account at (i) a traditional bank, (ii) an SR bank with a lower return where the proceeds of the return penalty will be used for the vaccinations for children in Africa or microcredits for women’s business in developing countries, or (iii) an SR bank offering an even lower return but with an in-kind gift of a voucher for a cultural activity of the respondent’s choice. In the third question (Q7), individuals

[4]The return penalty is either 0.2% or 0.4% compared to the traditional bank that yields 1%.

[5]The return penalty is either 0.25% or 0.5% .

indicate how much percentage they are willing to allocate in a savings account at a traditional bank and an SR bank with lower returns where the forgone returns will be used for the same cause as in Q6. In the last question (Q8), individuals choose to invest either in (i) a mutual fund linked to the AEX (Amsterdam Stock Exchange) index, (ii) a mutual fund that only consists of socially responsible companies with lower return[6], (iii) an SR mutual fund similar to (ii) but with even lower return and an in-kind gift of a deluxe edition book.[7]

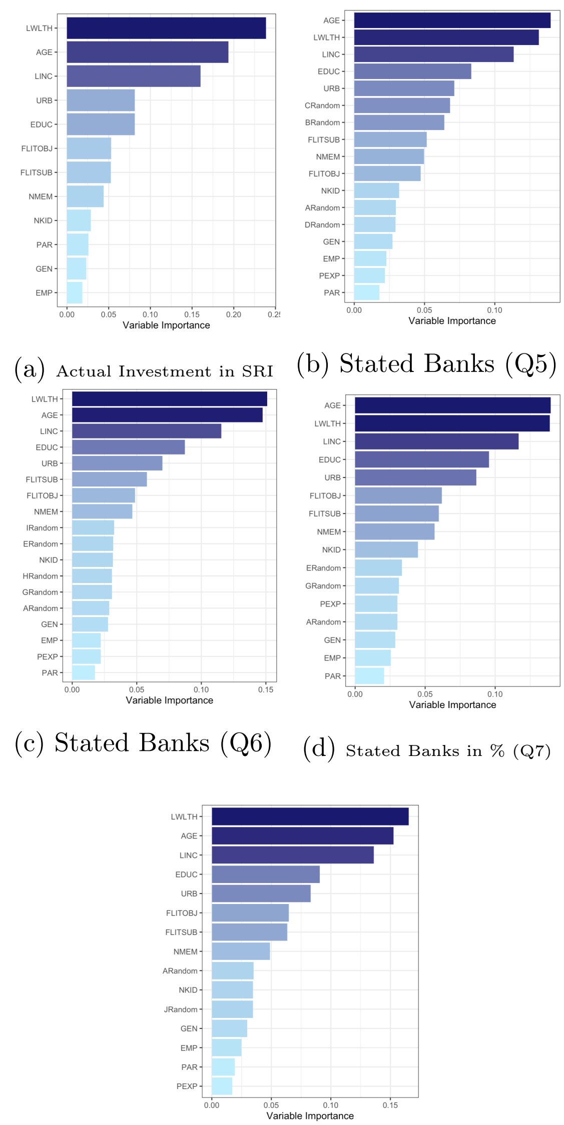
5 Results

5.1 Random Forest Analysis

Random Forest is a tree-based model and therefore retain the characteristics of decision trees. Decision trees are susceptible to overfitting when the tree gets deep as it tries to split the data even based on a noise. Therefore, I limit the depth of the tree, *max depth* to be always less than 40. Also, since it is an *ensemble* technique that aggregates a number of trees to reduce the variance of the model, I limit the minimum number of trees, *ntrees* to be at least 50. However, as the number of trees reaches a certain point, adding more trees does not progressively improve the estimation of the model and only increases the training time. Therefore, maximum number of trees are set to be less than 500. As mentioned in Section 3.1, Random Forest considers a random subset of m predictors out of the full set of p predictors. Here, I set the possible value of m to be any number between 1 and p. Based on the range of these hyperparameters, I implement a grid search that performs a random search of all the combinations of the them.

Since Random Forest is an *ensemble* method, it is considered *black-box* and difficult to understand how a particular function of the model, $\hat{f}_{bag}$, uses $\mathbf{x}$ to generate the results. Several methods have been suggested to overcome this lack of interpretability of black-box models. In the following sections, variable importance and partial dependence plots are used to better understand the relationship between the features and the target variable.

5.1.1 Variable Importance



(a) Actual Investment in SRI (b) Stated Banks (Q5)

(c) Stated Banks (Q6) (d) Stated Banks in % (Q7)

(e) Stated Stocks (Q8)

Figure 1: Variable Importance

[6] The SR mutual fund have return penalty of 0.5% or 1% compared to the indexed mutual fund.

[7] The return penalty is either 0.6% or 1.2%.

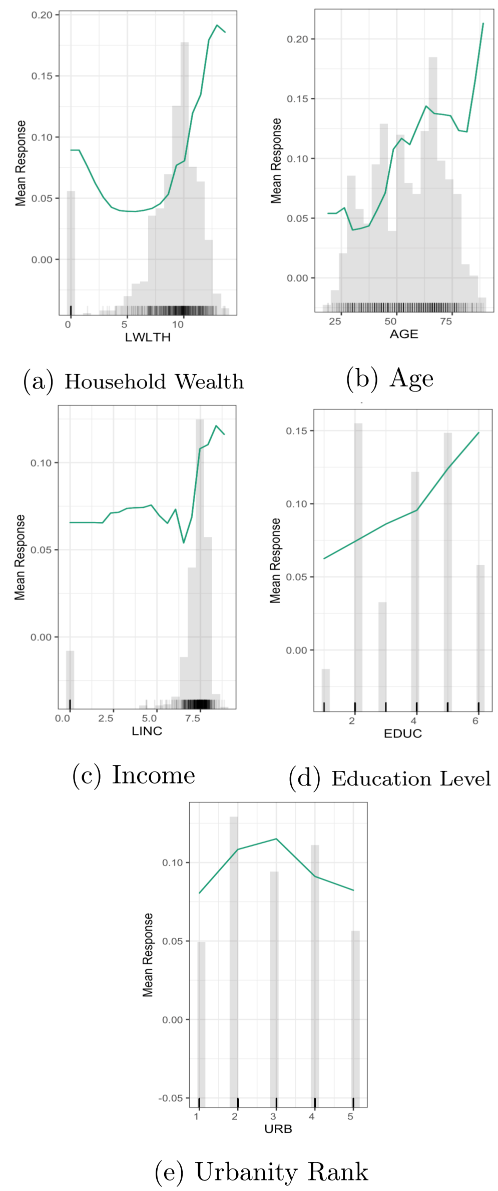
The variable importance in Figure 1 measures how frequent a predictor is selected as a splitting criterion and how much *pure* the tree becomes as a result. It shows that either age and household financial wealth are always the most important variables for the prediction of the revealed and the stated preference for SRI. Income, education level and urbanity rank are most commonly occurring predictors on the top of the list.

Notably, prior experience in SRI is usually one of the least important features. This could be due to the fact that the complex interactions between the predictors made in the tree-building process may already account for the effect of the prior experience in SRI in predicting the stated preferences for SRI.

5.1.2 Partial Dependence Plot

In this section, I use partial dependence plot by Friedman (2001) as described in Section 3.3 to investigate the marginal effects of the variables on the revealed and the stated preferences for SRI. Partial dependence plot is calculated based on the expectation of the predictions assuming all the data points to take a particular value of the predictor of interest. Therefore, the distribution of the predictor of interest is important. At the value where the predictor does not have many data, the average effect can be largely misleading (Molnar, 2020). Therefore, I do not rely my interpretation on the predictor values where the distribution is sparse.

Figure 2 presents the single variable partial dependence plots for most important predictors according to Figure 1a. SR investors have higher mean values for education level, age, income, and household financial wealth. For the household financial wealth, the partial dependence plot shows that the probability of actual investment in SRI has a convex curve that decreases initially and then increases with a steep slope after $LWLTH > 6$. Income shows a similar pattern to the household financial wealth. However, both features do not have many data points available in the lower quartile after 0. Therefore, the estimates of the partial dependence may be less reliable. In general, the older the investor, the higher the predicted likelihood that individual has an actual investment experience in SRI. Also, education level is strictly positively associated with the actual investment in SRI.

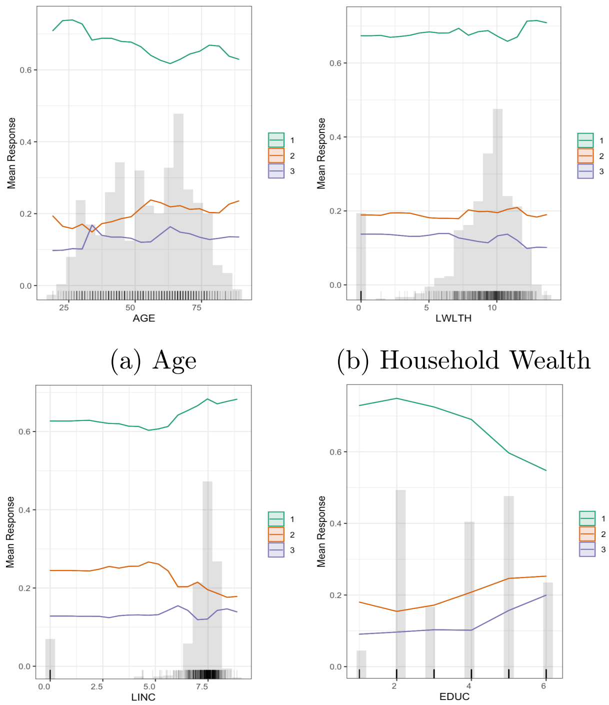


(a) Household Wealth (b) Age

(c) Income (d) Education Level

(e) Urbanity Rank

Figure 2: PDPs for Actual Investment in SRI (Q2)



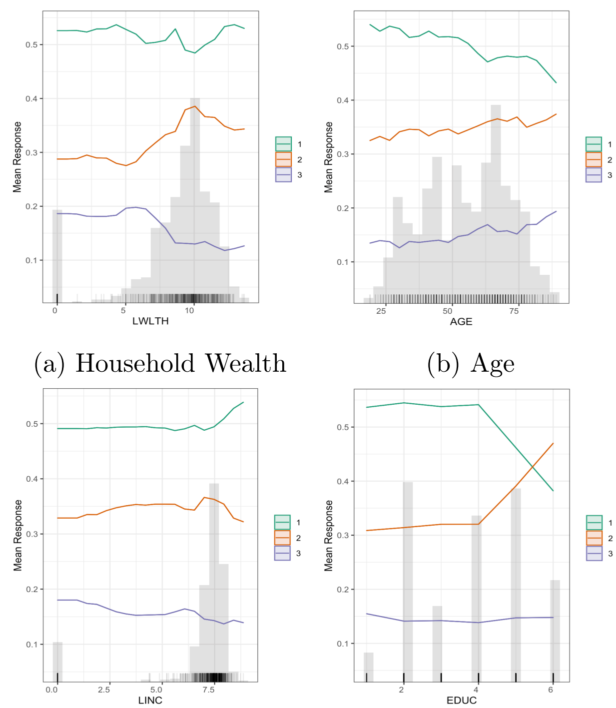
(a) Age (b) Household Wealth

(c) Income (d) Education Level

Figure 3: PDPs for Stated Banks (Q5)

In Figure 3, partial dependence plots for the top 4 predictors on the variable importance in Figure 1b are presented. The stated preferences for either of the two SR bank options slighly increase as investors get older. Household financial wealth does not show significantly steep curves although the preferences for SRI decreases slightly at the highest wealth level. Individual income has a negative association with the stated preferences for SRI. Education has a significant effect on the stated preferences for SRI. Investors with higher education level are more willing to invest in both SRI options. The partial dependence plots for the SP question with the information on where the forgone returns will be used (Q6) show similar patterns as Figure 3.

The parital dependence plots for the stated preferences in mutual funds are displayed in Figure 4. In Figure 4, the second option (pure SR) shows a positive vertical shift compared to Figure 3. Investors generally show higher willingness to invest in the pure SR option when the choices are made on mutual funds instead of banks. This may be attributed to the difference in the relative size of the return penalties on SRI. The pure SR bank suggests a 0.2-0.4% lower return than the return of 1% of the traditional bank. Investors might feel the relative size of this return penalty is too much compared to the SR fund offering a 0.5-1% lower return than the index fund which becomes trivial if they assume the index to have a return of about 5-6% at least.[8] Although they are willing to give up some returns in order to invest responsibly, they become sensitive to the return penalty as they feel the size of it is too big.



(a) Household Wealth (b) Age

(c) Income (d) Education Level

Figure 4: Partial Dependence Plots for Q8

For individuals with household financial wealth below the upper quartile, their stated preferences in pure SR fund is positively associated. On the other hand, if they are from a high wealth range, they have lower stated preferences for SRI as their wealth becomes larger. Investors tend to be less incentivized

[8]The 5-year average yearly returns between 2012 and 2016 when the survey took place is 9.2%.

by an in-kind gift to give up even more returns as their household financial wealth becomes larger. Older investors are generally more willing to invest in SRI. Similar to wealth, investors show concave curve for the income in prediction of the stated preferences for pure SRI. Lastly, investors with higher education level are always more likely to choose the pure SR fund while they are not willing to forgo more returns in exchange for an in-kind gift.

6 Conclusion

This paper investigates the household preferences for socially responsible investment. In particular, Random Forest model, an advanced machine learning algorithm is applied to the survey data of a representative sample on the Dutch population. Several techniques to overcome the drawbacks of the black-box machine learning models have been employed.

The results present the complex interactions between the socio-demographic characteristics of the investors. Overall, higher educated and older individuals are always more interested in SRI which is in line with the findings by Rossi et al. (2019) using linear models. Looking at their current investment behavior, individuals with high household financial wealth and income are more likely to have ever invested in SRI. Investors' stated preferences for SRI have more complex relationship than the revealed preferences. Marginal effects of the variables estimated by Random Forest algorithm suggest investors with highest income and financial wealth prefer traditional investment over SRI. Moreover, investors are willing to give up some returns to invest prosocially but tend to be sensitive to additional return penalty even when compensated with in-kind gifts.

This paper makes use of advanced machine learning model, Random forest, which is suitable for such an analysis in this paper where the relationship between the socio-demographic characteristics of investors and the decisions they make are likely to be intricate with complex interactions. Also, the limitation of using black-box machine learning models in exchange for achieving results with better generalizability is overcome with different techniques such as variable importance and partial dependence plots. With this approach, this paper provides more general results on the profile of the SR investors that is not limited to the sample in this data, while also allowing to understand the complex relationships between the characteristics of them.

Despite the several contributions and the advantages of this paper in understanding the SR investors, there are a few limitations. Since investors participate in the financial market with the intention of reaping investment returns, majority of them always choose the investment options with higher returns. Therefore, the data such as presented in this paper is naturally imbalanced, which is problematic for tree-based algorithms that require balanced distribution of the data. To handle this problem, I use random oversampling. However, resampling techniques are merely more sophisticated variations of replicating the observations in the data. This does not add any new information that helps with the estimation of the models. Also, resampling techniques may be subject to bias in the model. Thus, more data is needed for more robust and thorough investigation of the investor profiles for SRI in future research.

References

[1] M. Rossi., D. Sansone., A. van Soest., C. Torricelli. (2019), "Household preferences for socially responsible investments. " *Journal of Banking and Finance.*, vol. **105**, pp. 107-120.

[2] US SIF. (2020), "Report on US sustainable, responsible and impact investing trends. " *The Forum for Sustainable and Responsible Investment.*

[3] GSIA. (2018), "2018 Global sustainable investment review. " Global Sustainable Investment Alliance.

[4] C. Molnar (2020), "Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. "

[5] A. Riedl., P. Smeets. (2007), "Why do investors hold socially responsible funds. " *Journal of Finance.*, vol. **72**(6), pp. 2505-2550.

[6] X. Zhao., X. Yan., A. Yu., P. van Hentenrycka. (2020), "Prediction and behavioral analysis of travel mode choice: A comparison of machine learning and logit models " *Travel Behaviour and Society*, vol.**20**, pp. 22-35

[7] S. Levantesi., G. Piscopo. (2020), "The Importance of Economic Variables on London Real Estate Market: A Random Forest Approach. " *Risks.*, vol. **8(4)**, pp. 1-17

[8] J. Derwall., N. Guenster., R. Bauer., K. Koedijk. (2005), "The eco-efficiency premium puzzle. " *Financial Analysts Journal.*, vol. **61**, pp. 51-63.

[9] A. Kempf., P. Osthoff. (2007), "The effect of socially responsible investing on portfolio performance. " *European Financial Management.*, vol. **13**, pp. 908-922.

[10] F. J. Fabozzi., K. C. Ma, B. J. Oliphant. (2008), "Sin stock returns . " *Journal of Portfolio Management.*, vol. **35**, pp. 82-94

[11] H. Hong., M. Kacperczyk. (2009), "The price of sin: The effects of social norms on markets. " *Journal of Financial Economics.*, vol. **93**, pp. 15-36

[12] J. Nilsson. (2008), "Investment with a conscience: Examining the impact of pro-social attitudes and perceived financial performance on socially responsible investment behavior. " *Journal of Business Ethics* **83** (2), pp. 307-325.

[13] G. James., D. Witten., T. Hastie., R. Tibshirani. (2017), "An Introduction to Statistical Learning: With Applications in R. " Springer Texts in Statistics. Cham: Springer Publishing Company, Incorporated. ISBN 10: 1461471370.

[14] N.P.B. Bollen. (2007), "Mutual fund attributes and investor behavior. " *Journal of Financial and Quantitative Analysis.*, vol. **42**, pp. 683-708.

[15] R. Blagus., L. Lusa. (2013), "SMOTE for high-dimensional class-imbalanced data. " *BMC Bioinformatics.*, vol. **14**.

[16] M. Hossin., M.N. Sulaiman. (2015), "A Review on Evaluation Metrics for Data Classification Evaluations. " *International Journal of Data Mining Knowledge Management Process.*, vol. **5**, pp. 01-11.

[17] G.W. Snedecor., W.G. Cochran. (1989), "Statistical Methods. " *Iowa State University Press.*, vol. **8**.

[18] B.L. Welch. (1947), "The generalization of "Student's" problem when several different population variances are involved. " *Biometrika.*, vol. **34**, pp. 28â35.

[19] J.H. Friedman., B.E. Popescu. (2008), "Predictive learning via rule ensemble. " *Annals of Applied Statistics.*, vol. **2**, pp. 916-954.

[20] H. Han., W.Y. Wang., B.H. Mao. (2005), "Borderline-SMOTE: A New Over-Sampling

Method in Imbalanced Data Sets Learning. " *Advances in Intelligent Computing.*, vol. **3644**, pp. 878-887.

[21] M.A. Friedl., C.E. Brodley. (1997), "Decision Tree Classification of Land Cover from Remotely Sensed Data. " *Remote Sensing of Environment.*, vol. **61**, pp. 399-409.

[22] N. V. Chawla., K. W. Bowyer., L. O. Hall., W. P. Kegelmeyer. (2002), "SMOTE: Synthetic Minority Over-sampling Technique. " *Journal Of Artificial Intelligence Research.*, vol. **16**, pp. 321-357.

[23] G. Douzas., F. Bacao., F. Last (2018), "Improving Imbalanced Learning Through a Heuristic Oversampling Method Based on K-Means and SMOTE. " *Information Sciences.*, vol. **465**.

[24] D. Ariely.. A. Bracha., S. Meier. (2009), "Doing Good or Doing Well? Image Motivation and Monetary Incentives in Behaving Prosocially. " *American Economic Review.*, vol. **99**, pp. 544-555.

[25] L. Breiman., J.H. Friedman., R.A. Olshen., C.J. Stone. (1984), "Classification and regression trees. " *Boca Raton.*

[26] L. Breiman. (1996), "Bagging Predictors . " *Machine Learning.*, vol. **24**, pp. 123.140

[27] K.M. Ting., J.R. Wells., S.C. Tan., S.W. Teng., G.I. Webb. (2011), "Feature-subspace aggregating: ensembles for stable and unstable learners. " *Machine Learning.*, vol. **82**, pp. 375-397.

[28] M. Pazzani., C. Merz., P. Murphy., K. Ali., T. Hume., C. Brunk. (1994), "Reducing Misclassification Costs. " *Machine Learning Proceedings* ., pp. 217-225.

[29] D. Lewis., J. Catlett. (1994), "Heterogeneous Uncertainty Sampling for Supervised Learning. " *ICML.*

[30] M. Kubat., S. Matwin. (1997), "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. " *ICML.*, pp. 179-186.

[31] N. Japkowicz. (2000), "Learning from Imbalanced Data Sets: A Comparison of Various Strategies. "

[32] C. Ling., C. Li. (1998), "Data Mining for Direct Marketing: Problems and Solutions. " *KDD.*

[33] J. Huang., C.X. Ling. (2005), "Using AUC and accuracy in evaluating learning algorithms. " *IEEE Transactions on Knowledge and Data Engineering.*, vol. **17**, pp. 299-310.

A Descriptive Analysis

Table 2: Descriptive Statistics

The table shows the summary statistics for the entire sample (column 1), sub-samples for the non-SR owners (column 2) and the SR owners (column 3), and the two-tailed p-value for mean differences t-test. Partner is a dummy for respondents who are either married or living together with a partner. Income is a logarithm of the gross income of each individual and household financial wealth is in logarithm as well.

	(1) Total		(2) Traditional Investors		(3) SR Investors		(4) Differences: (2)-(3)
	Mean	SD	Mean	SD	Mean	SD	p-value
Gender (Female)	1.473	0.499	1.479	0.500	1.408	0.493	0.086
Age	54.758	16.286	54.251	16.402	60.783	13.487	0.000
Education Level	3.819	1.484	3.768	1.478	4.433	1.420	0.000
Subjective Financial Literacy Level	2.206	0.724	2.190	0.723	2.389	0.713	0.001
Objective Financial Literacy Level	2.348	0.816	2.316	0.828	2.708	0.544	0.000
Employment Status	0.495	0.500	0.499	0.500	0.446	0.499	0.204
Partnership Status	0.760	0.427	0.763	0.425	0.726	0.447	0.299
Number of Children in Household	0.621	1.012	0.633	1.017	0.484	0.945	0.077
Number of Household Members	2.405	1.194	2.431	1.206	2.210	1.138	0.027
Individual Income	6.947	2.353	6.887	2.406	7.657	1.426	0.000
Household Financial Wealth	8.466	3.244	8.335	3.215	10.021	3.194	0.000
Urbanity Rank	3.081	1.301	3.028	1.319	2.871	1.257	0.153
N	2022		1865		157		2022

Table 2 shows the summary statistics for the predictors on the entire sample, sub-samples for the traditional investors and the investors who have ever invested in SRI. The p-value for the mean difference is calculated based on the Welch's t-test with the following formula (Welch, 1947):

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{3}$$

where $\bar{x}_1$ and $\bar{x}_2$ are sample means for the tradtional and SR investors, s_1^2 and s_2^2 are the sample variances, and n_1 and n_2 are the sample sizes. Based on the t-test, we see that SR investors tend to be older, with higher education level and financial literacy, have less number of household members, and have noticeably higher income and household financial wealth.

B Variable Correlation

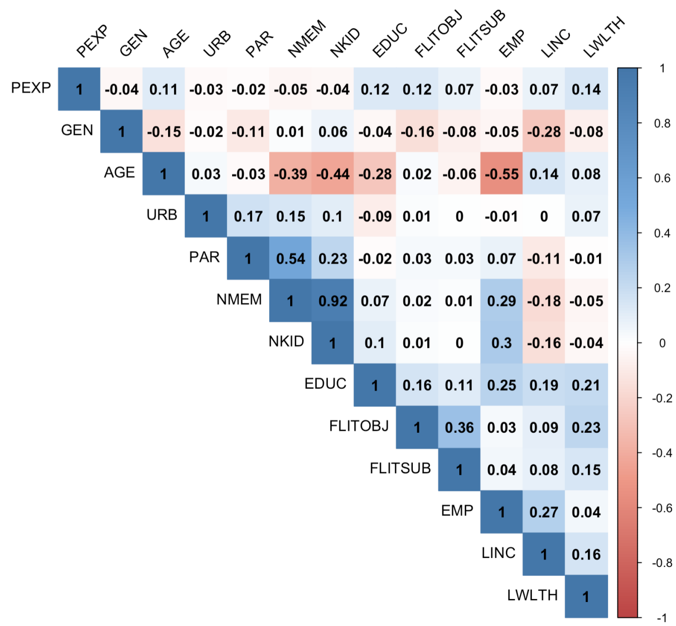


Figure 5: Correlation Heatmap

Figure 5 shows the correlation between the socio-demographic characteristics of the investors. The number of household members is strongly correlated with the number of children in household as well as the relationship status. The sample is administered on the adult population of the Netherlands and the median and the mean is mid-50. Therefore, age variable is negatively correlated with the number of household members and children, education level and employment status.

Women are associated with slightly lower income. As anticipated, education level shows slight positive correlation with the employment status, income and household financial wealth. Interestingly, objective financial literacy level is not strongly correlated with subjective level. This is due to the fact that investors with lower objective financial literacy level overestimate themselves while investors with higher objective literacy underestimate themselves in this data. Individual income and household financial wealth are not much correlated.

C SR Investor Motives

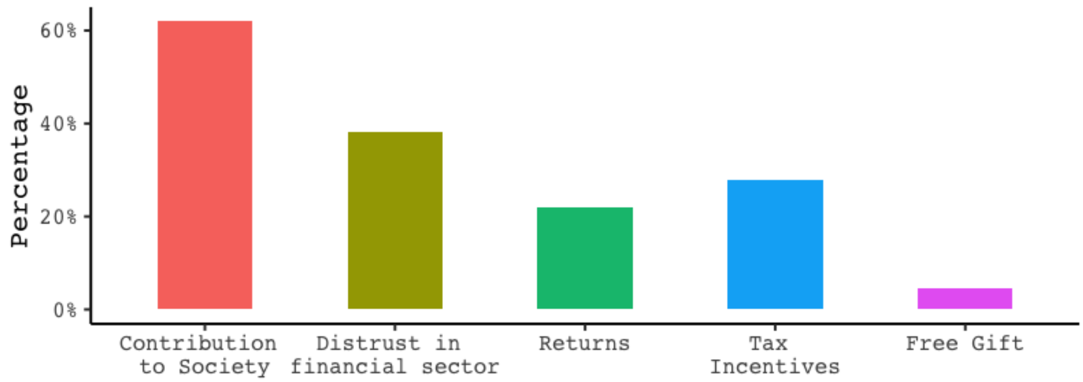


Figure 6: Reasons for Investing in SRI

In our sample, 7.82% of the investors state they have ever invested in SRI. Among them, most participate with a prosocial motive. 61.9% of the investors state the reason for investing in SRI is to contribute to improving society. 38% report they have more confidence in the banks in the SR sector than the rest of the financial sector. The third reason is because they have fiscal advantages investing in these products. As Rossi et al. (2019) states, Dutch government provides incentives on certain SR products through tax exemption and therefore, the advantage in after-tax return could be about 0.7-1.9 percentage points. Less common reasons are SR investment returns and receiving a free gift from making an SR account.

D Decision Tree

In this appendix, a detailed description of decision tree, the base model for Random forest, is provided. Decision tree algorithm is a base method for many machine learning models. It is widely used for classification or regression problem ever since Breiman et al. (1984) introduced CART (Classification and Regression Tree) model. The basic process of building a classification tree follows two steps (James et al., 2017):

1. The predictor space - that is, the set of possible values for $x_1, x_2, \ldots, x_p$ - is divided into J distinct and non-overlapping regions, $R_1, R_2, \ldots, R_J$.
2. For every observation that falls into the region R_j, the algorithm makes the same prediction, which is the most commonly occurring class of training observations in R_j to which it belongs.

The regions R_j are selected to minimize the *Gini index*, given by the following equation:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{4}$$

where $\hat{p}_{mk}$ is the proportion of the training data in the mth region of the kth class. G gets smaller as a node gets more *pure*, meaning it predominantly contains observations from a single class.

However, due to computational limitations, it is not feasible to consider all possible partition of the data space into J regions (Breiman et al., 1984). Therefore, CART model takes *binary recursive splitting*. The data space is recursively splitted with a splitting rule that divides the space in the form of $\{\mathbf{x}|x_j < s\}$ and $\{\mathbf{x}|x_j \geq s\}$ where it considers all predictors $\mathbf{x}$ and all possible values of the cutpoint s at each node and select the predictor and s that minimizes the *Gini index*. The tree repeats the same process within each of the resulting region until reaching the final node. At the final node, the overall prediction is made based on the *majority vote*, that is, the most commonly occurring class among all predictions.

Since decision trees are built from the root node to leaves (terminal nodes) by successively partitioning the remaining data space, they naturally capture the interactions and the complex relationships between the variables. In addition, they show which variable is the most important in the estimation of the model as the variable selected at the root node attains the smallest *Gini index* and the next variables down the tree have the smallest *Gini index* within that subspace and so forth. However, despite the advantages, decision trees suffer from high variance: a slight modification in the data can lead to somewhat different results and splits. Decision trees are therefore regarded as unstable and sensitive to the noise in the data (Ting et al., 2011). To overcome this problem of unstable machine learning models, Breiman (1996) developed *"bootstrap aggregating"*, also known as *bagging*. In this paper, I focus on random forest method which is an improvement over bagged decision trees.

D.1 Random Forest Split Criterion

Within each tree, random forest makes split decisions based on the squared error proposed by Friedman (2001) instead of *Gini index*.[9] The potential splits from the current region into two sub-regions (R_l, R_r) are

[9] The machine learning interface **h2o** uses this impurity measurement.

determined with the least-squares improvement criterion calculated as the following:

$$i^2(R_l, R_r) = \frac{w_l w_r}{w_l + w_r}(\overline{y}_l - \overline{y}_r)^2 \tag{5}$$

where $\overline{y}_l$ and $\overline{y}_r$ are the mean response of the left and right daughter nodes and w_l and w_r are corresponding weights[10]. i^2 is the biggest when the split can best divide the regions to make the daughter nodes more *pure*. For example, in case of binary classification, i^2 is bigger and the resulting nodes are considered more pure when $\overline{y}_l = 0.8$ and $\overline{y}_r = 0.1$ compared to the case with $\overline{y}_l = 0.5$ and $\overline{y}_r = 0.5$, which is when the positive and the negative classes have equal number of observations in both right and left daughter nodes.

[10]Exact calculation of the weights can be found in Friedman (2001).

E Variable Importance

Variable importance I_j^2 for the predictor x_j:

$$\hat{I}_j^2(T) = \sum_{t=1}^{J-1} \hat{i}_t^2 1(v_t = j) \tag{6}$$

where t is a non-terminal node in the tree T with J terminal nodes, v_t is the variable considered at node t, and $\hat{i}_t^2$ is the least-squares improvement from the Equation 5. Since random forest uses bagging, the aggregated variable importance over M trees $\{T_m\}_1^M$ is calculated as Friedman (2001) proposes:

$$\hat{I}_j^2 = \frac{1}{M} \sum_{m=1}^{M} \hat{I}_j^2(T_m) \tag{7}$$

This captures which variables contribute the most to the model and which are comparatively trivial.

F Resampling Method

In this paper, I used random oversampling which is one of the simplest resampling techniques for imbalanced data. However, the same analysis as in Section 5 has been also carried out using Synthetic Minority Oversampling Technique (SMOTE) that has been widely used in the literature (Chawla et al., 2002; Han et al., 2005). SMOTE generates "synthetic" observations for the minority class using an over-sampling strategy. For each observation $i_{min,p}$ with the minority class, SMOTE randomly selects one of its k-nearest neighbors $i_{min,r}$ that have the smallest Euclidean distance. With $i_{min,p}$ and $i_{min,r}$, it generates a new observation i_n where the values for variables $\mathbf{x}$ are an interpolation of the values from $i_{min,p}$ and $i_{min,r}$ according to the following formula (Blagus Lusa, 2013):

$$\mathbf{x}_{i_n} = \mathbf{x}_{i_{min,p}} + u_{i_n}(\mathbf{x}_{i_{min,r}} - \mathbf{x}_{i_{min,p}}) \tag{8}$$

where u_{i_n} is randomly chosen from U(0, 1) for each new observation i_n.

The main intuition and interpretation of the results using SMOTE do not deviate much from those of *random oversampling*. However, despite its popularity, SMOTE does not perform well under certain situations and is not robust to the noise (Blagus Lusa, 2013; Douzas et al., 2018). The results using SMOTE suggest it underperforms simple *random oversampling* and therefore they are not presented in this paper. This could be because the data in this paper has several randomization factors for the investment options and the hypothetical inheritance which are completely exogenous to the model. These factors may have a direct causal relationship to the decisions by the investors unlike other predictors with indirect relationship. SMOTE creates synthetic values for all predictors with interpolation disregarding the nature of each variable. Therefore, newly generated observations do not have the appropriate randomization features that may explain the target variable.

F.1 Performance Comparison

Figure 7 compares the performance between different models for each question. The evaluation metric that compares the performance of the models is Log loss calculated by the following equation:

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j} ln(p_{i,j}) \tag{9}$$

where N is the number of observation, C is the number of class labels, $y_{i,j}$ is the dummy variable for jth class and $p_{i,j}$ is the prediction probability for jth class. Log loss for random guessing, $-log(1/C)$ where C is the number of classes, is provided for each model as a benchmark. Logit model is trained and tested on the same dataset as for the Random Forest model and includes the same variables plus a squared value of AGE for the non-linear effect. The results suggest Random Forest slightly outperforms the logit model for all questions except for Q8. The performance of a model shows how much generalization ability it has when tested with the unseen data (Hossin and Sulaiman, 2015). Since Random Forest does not have strict assumptions about the distribution of the data and allow the features to interact with each other on various levels, it may be able to capture the real relationship between the socio-demographic characteristics of the investors and their stated preferences for SRI.

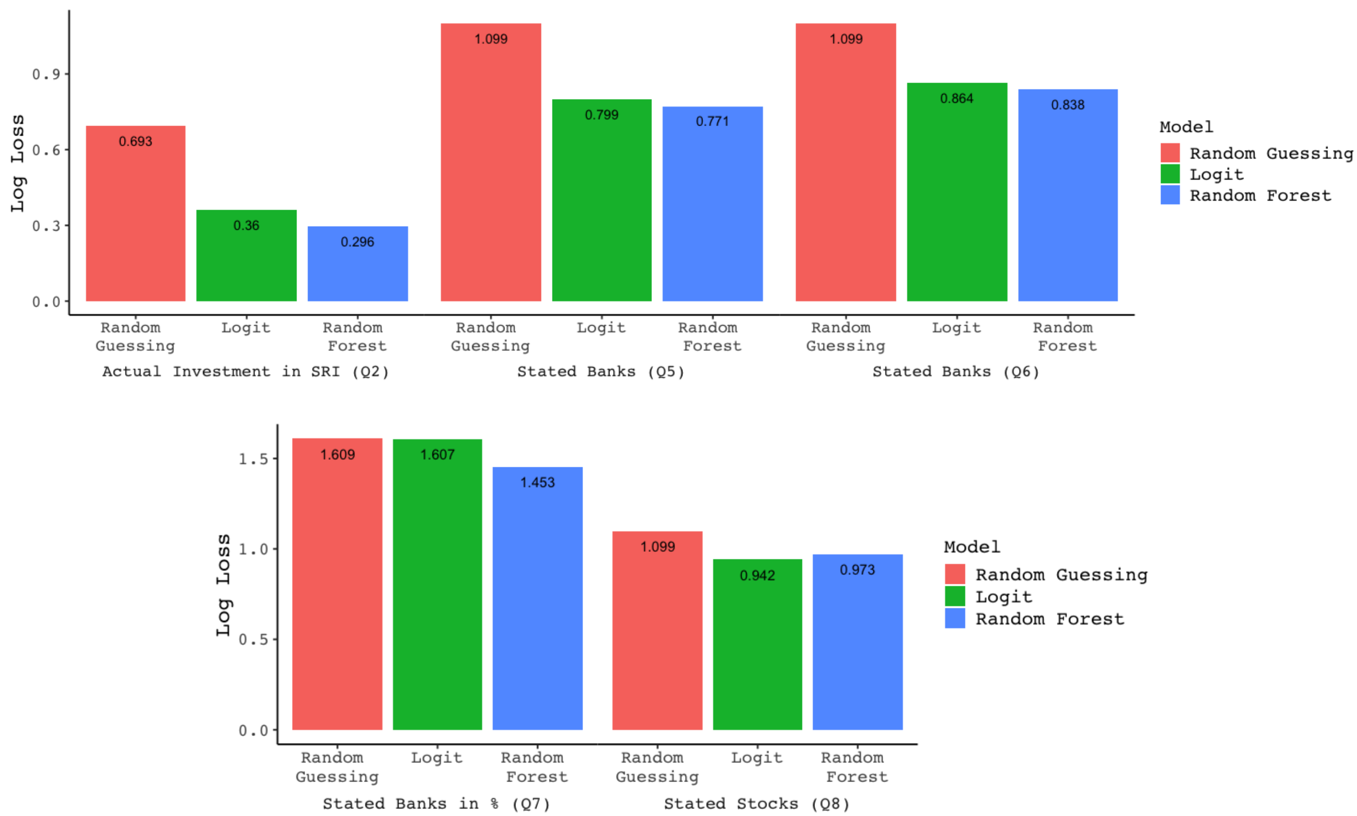


Figure 7: Comparison of the Model Results