



Aalto University
School of Science

Interim report

MS-E2177, Seminar on Case Studies in Operations Research
May 30, 2022

Lauri Jokinen	lauri.jokinen@aalto.fi	(project manager)
Christian Segercrantz	christian.segercrantz@aalto.fi	
Ilmari Vauhkonen	ilmari.vauhkonen@aalto.fi	

Client: Aktia Life Insurance

Contents

1	Project status	1
2	Changes in the objectives and the scope of the project	1
2.1	The input arguments	1
2.2	Linear approximations of the entropy optimization constraints	2
2.3	Portfolio optimization	2
2.4	User warnings	2
3	Changes to the initial project plan	3
4	The updated risk management plan	3
5	Preliminary results	5

1 Project status

As of April 20, the coding part of the project is mostly done. The functions for performing the views preprocessing, entropy minimization, and Markowitz portfolio optimization have been implemented and tested together. Some code for visualization has been written as well. However, creating and testing the main script for running the process, and implementing a Jupyter notebook for creating a concise results analysis still lie ahead. The code documentation and instructions will also be written in a separate document.

Currently, we are a bit behind the original schedule. All the past deadlines were finished on schedule, but we will not be able to finish the next deadline in time. However, the schedule is very loose relative to the course's pace, so we still have time to complete the tasks in time.

We have a few more requests from the client, but after that, we can start working on the code documentation and final report. The background section of the project plan has some topics, e.g., methods for gathering the views data, that we still haven't studied any further.

2 Changes in the objectives and the scope of the project

The objectives are unchanged, but with some additional technical details regarding our implementation. We have listed some of these details below.

2.1 The input arguments

Talks with our client have helped us to clarify especially the way the inputs are given, processed, and interpreted. For example, defining each individual financial instrument's (stocks, bonds, derivatives) returns in each scenario was solved by adding another input, a matrix containing each asset's sensitivities to economic factors (stock indices, bond yields, macroeconomic measures, and so on).

The 'traditional' Markowitz portfolio procedure was altered a bit by including more constraints concerning the bounds for different asset position sizes. Now the user can specify

bounds for any individual asset or combination of assets, and assets requiring no immediate capital invested (e.g. swaps) can also be included this way. The format in which the inputs will be provided is also mostly fixed.

2.2 Linear approximations of the entropy optimization constraints

Interestingly, the optimization constraints in [1] were assumed linear, but, e.g., the variance of an option cannot be formulated as a linear constraint. In [2], mean values of options are fixed, so that linear constraints can be formed, but no implications of this assumption are mentioned. We used this assumption as well, but we aim to analyze its impact on the results' accuracy.

2.3 Portfolio optimization

In the project plan, we mentioned the possibility of different risk definitions. For the time being, we have decided to go with a simple definition: standard deviation of expected returns. Different definitions for risk are left for future research.

2.4 User warnings

As the code implementation proceeded, we became aware of numerous situations that could cause errors in the process, either by making the program fault or by distorting the results so that they become non-informative. The idea to prevent the first type of error is to implement adequate error handling procedures for each function. The latter type of errors can be avoided by writing good instructions and documentation for the end-user. A few possible caveats have been listed below.

1. Wrong data type of input arguments, either those given by the user or by other functions. This includes also the shape of the data vectors and matrices (errors related to transposing are very likely).
2. Creating such a set of constraints either at the view formulation or Markowitz optimization stage, that make the optimization problems infeasible. This is an important issue since the feasibility of the problem is currently not tested by the code before the actual optimization algorithm runs.
3. Errors, such as typos in the user inputs, that create the wrong result but do not make the program crash. Many of these cannot be detected by the program, but it is the user's responsibility to ensure the accuracy of the data. Some certain points can be detected, however, e.g., that the correlation coefficient lies in $[-1, 1]$ and that all variances are non-negative.
4. Confusions with the data inputs. E.g., as bonds are usually denoted by their yields, and the yields are measured in basis points (1 percentage point equals 100 bp), the user might accidentally denote factor scenarios and asset sensitivities in the wrong way.

3 Changes to the initial project plan

The project plan, like the project scope, remains largely unchanged. However, we have granted ourselves an additional week for completing the report, because of the additional work related to the programming (for example, user input preprocessing). Another minor change is the increased priority of documentation, as we concluded that adequate and clear documentation and user instruction are crucial for the successful adoption of the program in the client organization. Otherwise, we stick to the initial schedule and targets.

4 The updated risk management plan

The risks are listed in Table 1.

Risk	Likelihood	Impact	Effect on project	How to prevent
Problems in team collaboration	Low	High	Project schedule will be delayed	Meetings at regular intervals.
Insufficient communication between the team and client	Medium	High	The client is not satisfied with the results	Clear communication between team members, project manager, and the client. Also, staying on schedule will help.
Feeding the inputs is too tedious (too much and frequent manual work is required)	Medium	Medium	Project becomes irrelevant for the client	Automatize input processing as much as possible. Write clear and concise instructions. Discuss finding ways to reduce the amount of manually inputted data.
The given model does not satisfy client requirements (even though the model is given to us by the client)	Medium	Medium	Project becomes irrelevant to the client	We evaluate the results of the model critically and regularly with the client.
Shift of focus on the code	Low	High	The technical scope changes to focus more on UX-programming than mathematical programming	Re-scope the project and the features to be implemented.

Table 1: Risks

5 Preliminary results

We have been able to do some test runs with our data, and the results of these will be presented in this part. Table 2 displays the views we have used in the example. The model was run using 36 different factors to construct the views and asset sensitivities, and then the behavior of 13 example assets (those factors that can be understood as financial instruments), were used in the Markowitz phase. Figure 1 shows the Markowitz model using the prior scenario distribution, i.e., the data where each scenario has a weight equal to $1/S$ where S is the number of scenarios. Figure 2 illustrates the results when the posterior distribution (obtained from the entropy pooling minimization) was used.

The data for the results was given by the client, Aktia, and the views were generated arbitrarily by the group only for testing purposes. The results show the functionality of the code, but further testing is needed.

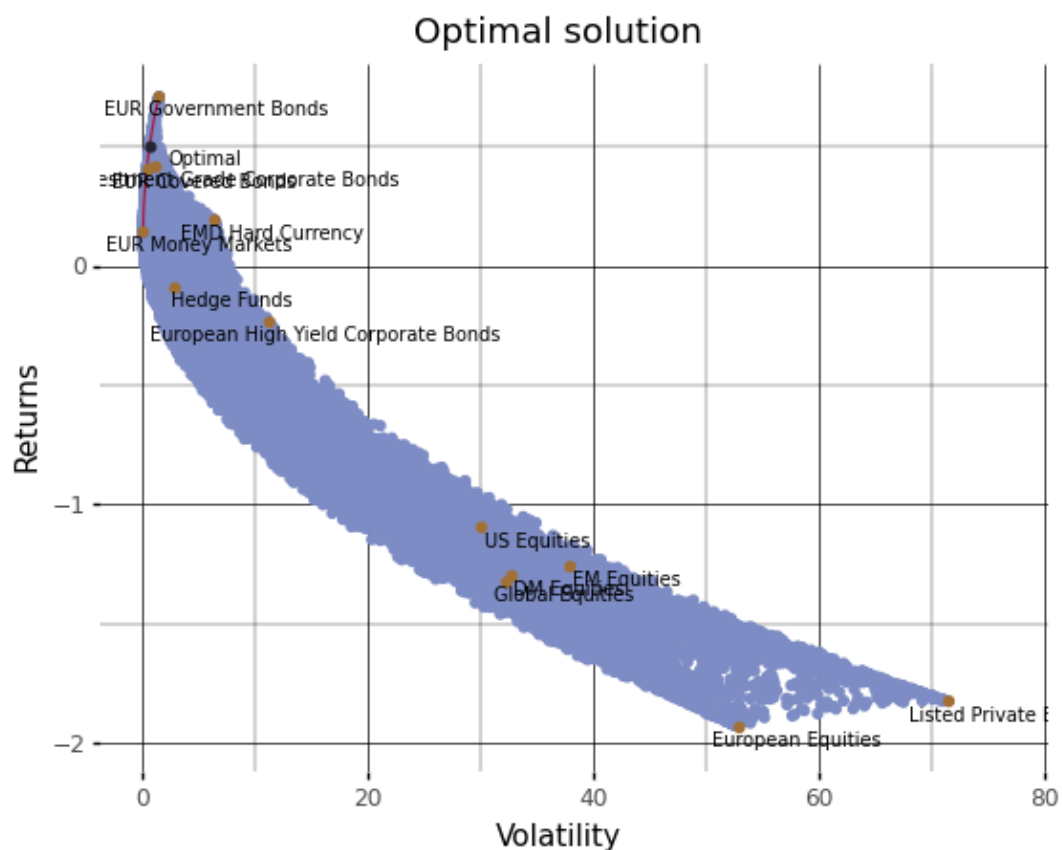


Figure 1: A visualization of the Markowitz optimization for the **posterior** data distribution. The investment options are marked with brown and the optimal frontier in red. The optimal value for a return of 50 % is marked with black. The blue cloud represents the different possible linear combinations of the investment options.

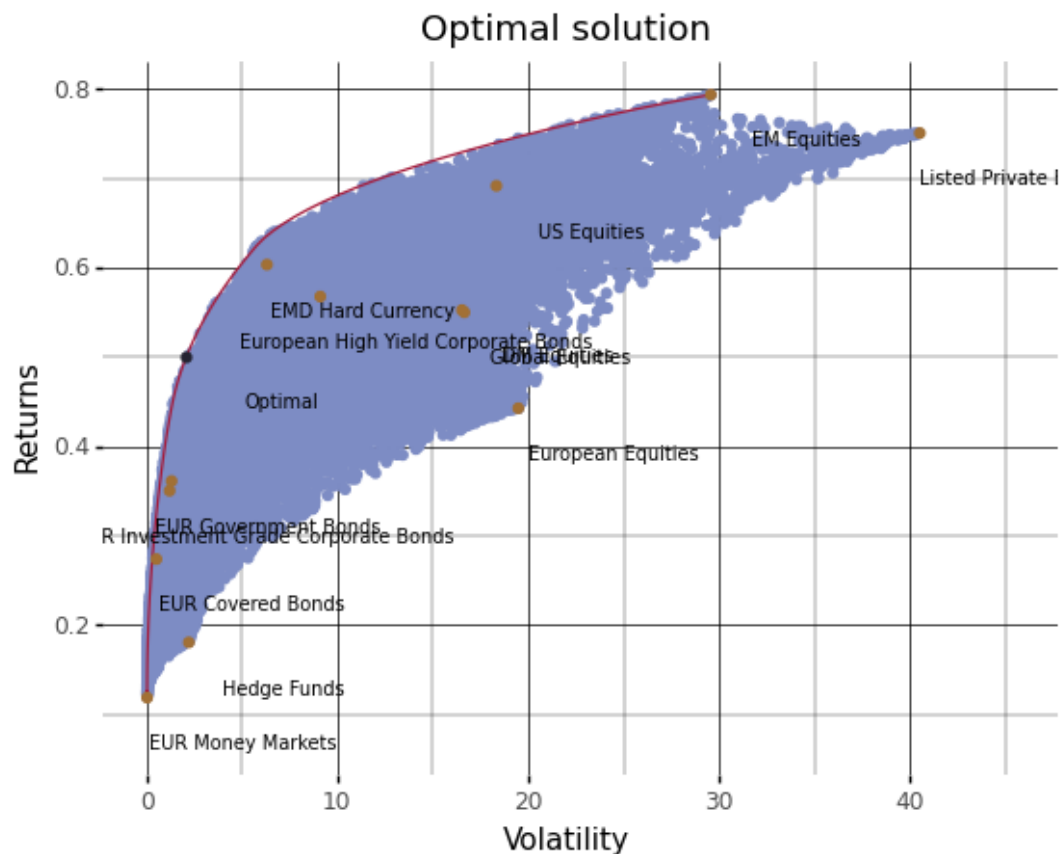


Figure 2: A visualization of the Markowitz optimization for the **prior** data distribution. The investment options are marked with brown and the optimal frontier in red. The optimal value for a return of 50 % is marked with black. The blue cloud represents the different possible linear combinations of the investment options.

Table 2: Views input format. One line in the table corresponds to one constraint. Detailed instructions for filling the table will be provided later.

* View on	* Risk factor 1	Risk factor 2 (applicable for corr)	* Operator	* Constant (alpha)	Multiplier (beta)	Risk factor 3	Risk factor 4 (applicable for corr)
Mean	European Equities	-	=	0,05	0,00	Global Equities	-
Vol	European Equities	-	<	0,10	1,00	Global Equities	-
Corr	European Equities	Global Equities	=	0,90	0,00	EM Equities	European Equities
Corr	EM Equities	European Equities	>	0,00	0,50	Global Equities	Global Equities

References

- [1] Attilio Meucci. “Fully Flexible Views: Theory and practice”. *Fully Flexible Views: Theory and Practice*, *Risk* 21.10 (2008), pp. 97–102.
- [2] Jan Alexandersson. *The Entropy Pooling Approach: Incorporating Views on Forecast Distributions*. Matematiska institutionen, Stockholms universitet, 2021.