Aalto-yliopisto
Aalto-universitetet
Aalto University

ELEC-E7852
Computational Interaction and Design

Assignment A2b

Bayesian optimization

Aitor Urruticoechea Puig
aitor.urruticoecheapuig@aalto.fi
Student N°101444219
November 2024

Notice

Acknowledgement

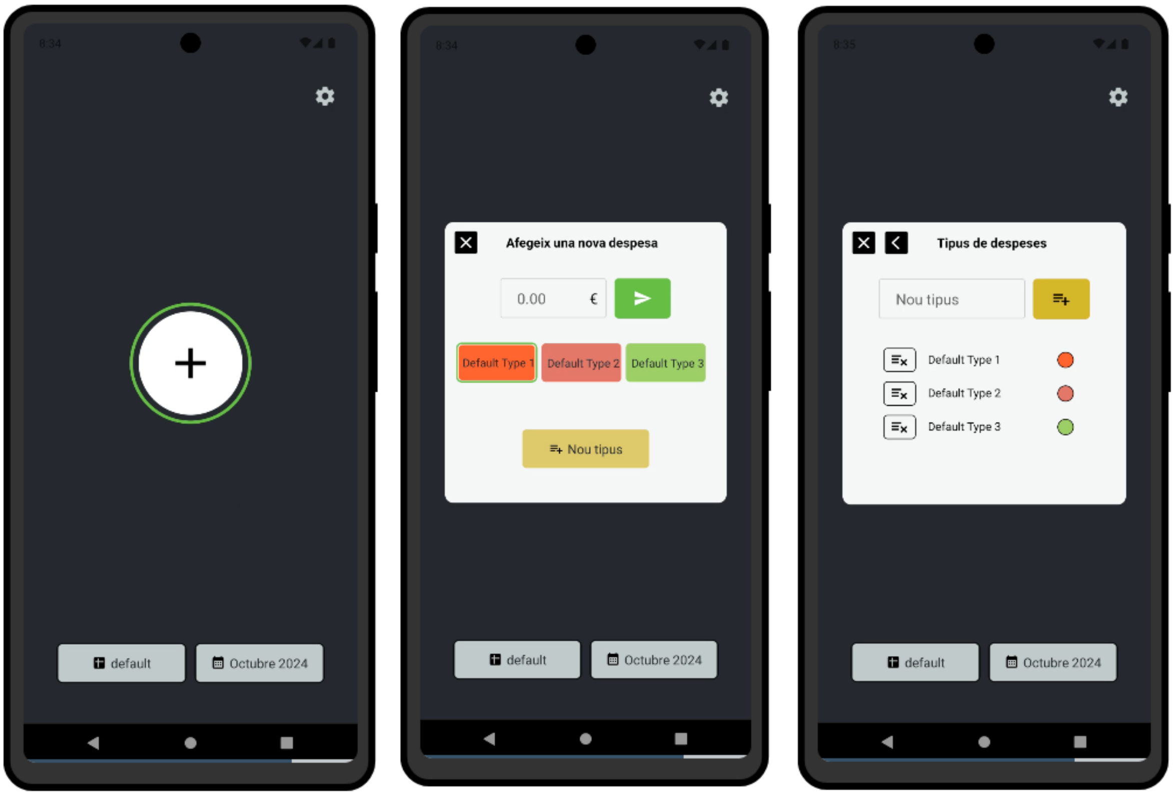
Parts of the code implemented in this work were generated with the assistance of GitHub Copilot. None of its proposals were, however, employed without refinement and necessary tweaks to adapt it to the nuances of the tasks at hand; making it impossible to identify and subsequently mark which lines of code were human or machine-made, for many were the fruit of the combination of both.

Contents

List of Figures

1 Introduction and case description

From "Urruticoechea", the assigned topic from the assignment A2a was "Optimizing a design with a designer in the loop (human-in-the-loop design)". Just as in the other assignments, a side project that I have been slowly developing is to be used as a playground. This is a spending tracking app which should help synchronize all your spendings and update your custom Google Sheets file with ease. Its UI is very much still in development, but can be seen in Figure 1 (unfortunately, the UI is only available in Catalan at the moment). The idea is to simplify the process as much as possible. The user needs only to choose a spending type and add the amount, the system should do the rest (Figure 1b). Spending types can be added, removed, and edited in a separate screen (Figure 1c).



(a) Home screen (b) "Add new spending" window. (c) "Spending types" window.

Figure 1: "DespesApp" spending tracking app overview.

In A2a, the optimization was focused on the colour picker for the spending types, which was iterated upon designer ratings. Now, for multi-objective optimization, a new variable to be iterated upon needs to be added to the mix. Staying within the topic of designer-in-the-loop approaches, let us consider another colour variable to add to the mix: the colour attributed to spending types (in Figure 1b, the centre gold button, and in Figure 1c the top right confirm button). This is an interesting dual proposition because the colour options for spending types and the type-confirm button are the basic prominent colour features in these two main windows for the app. Thus, iterating upon both of them continuously is relevant, as the colour harmony between them is critical.

2 Bayesian Optimization Implementation

The main idea for optimization is to implement two parallel optimization loops for both the now known RGB colour bar and for the colour that the "confirm" button should have (top right corner, see Figure 2). Thus, at each iteration, two scores are to be asked, and from these, points can be drawn for the Pareto front; while finding the best optimal combination of both.

The Bayesian Optimization itself remains mostly unchanged from A2a, with simply an addition of a second variable during the loop - meaning that, in actuality, two different optimizations are happening at the same time. This will be helpful in plotting the Pareto Front, as each axis will have followed an independent optimization process. Shortcomings in each of them can be independently analysed and, in the same way, best-type combinations of both processes can be also found. Also in a similar fashion to A2a, 5 initial samples will be taken and used as an optimization starter point for a 10-iteration loop; hopefully yielding meaningful results.

(a) Sample example 1.

(b) Sample example 2.

Figure 2: Spending types examples for optimization.

3 Pareto Front

To test this approach with input from someone other than the author; the author's partner was asked to, with limited context, answer the rating questions typically asked a designer. As previously described, the user is asked to independently rate each of the moving parts: the "confirm add type button" in the top right corner of the window, and the RGB colour bar distribution of colours in the middle of the screen. In the background, the parallel optimizations are running, progressively outputting new design options until the 15 (5 initial plus 10 iterations) have been rated. This allows for the plotting of a Pareto Front (Figure 3) and an initial approximation of where the Front itself might be, though the data is limited as the exploitation capabilities of the implemented algorithm have ample room for improvement. Nevertheless, the performed experiment does shed some light on where the actual best points might be, and showcases the potential of this approach for context-based multi-objective iterative optimizations.

From this first approximation to the Pareto Front, it seems clear that a perfect $5-5$ relation might be challenging to find. The trade-offs between better button scores against better RGB bar ratings seem so be a bit unbalanced, with an approximate $2-1$ ratio. Basically, an improvement of 1 point in button rating results in a decrease of 0.5 in Bar rating. It is hard, however, to extract deep conclusions from this fact, at it is hard to compare both ratings when one is for a single button colour and the other is for a RBG bar with multiple moving parts; thus making it possible that this relation actually stems from a more fundamental balancing issue between the two scores. Nonetheless, and with this data being the only available, it is clear that it might be desirable to sacrifice, to some extent, some score points for the RGB bar to better the button rating, as that trade-off is a net gain in terms of overall scores.

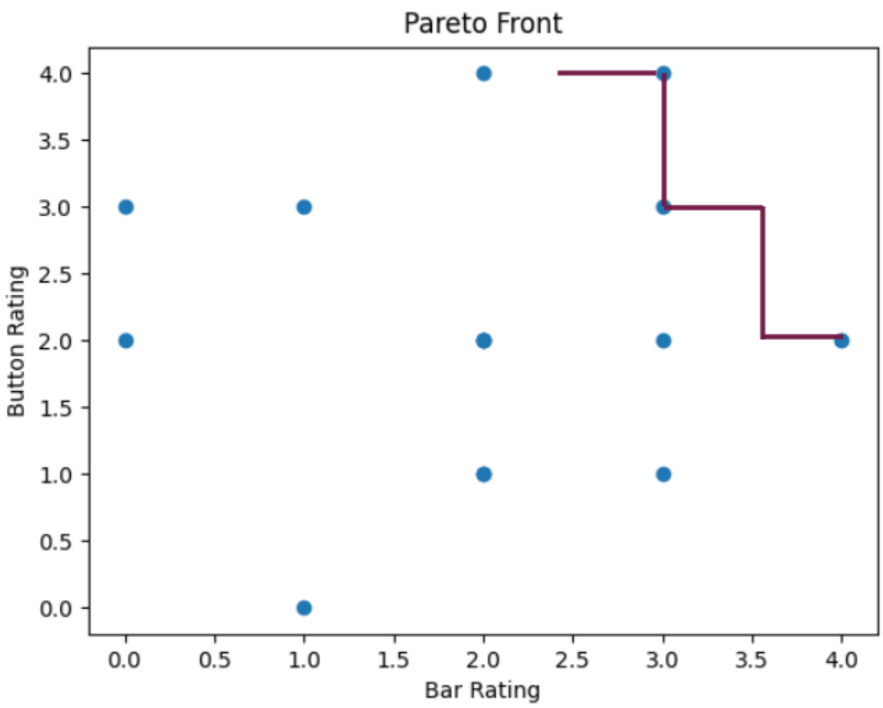


Figure 3: Result scatter plot and Pareto Front approximation

4 Conclusion

In Figure 4, a sample of 3 potentially best results have been generated from the optimized system after the experiment has been performed. Just as with assignment A2a, the fact that the optimization system does not properly exploit leads means the optimization could theoretically perform better given the same number of inputs by the user or designer. Be it as it may, however, Figure 4 shows how despite this shortcoming, the algorithm is able to find very satisfactory results with coherent combinations of button colour and RGB distributions. It is quite interesting how, despite this being two nominally independent optimizations, the fact that they are trained together result in coherence emerging from the best results. For instance, one can see how results in Figures 4a and 4b achieve tones for the button quite similar to the latter tones included in the RGB distributions, as the physical proximity between those might have prompted the user to rate higher such mental connections.

(a) Sample result 1.

(b) Sample result 2.

(c) Sample result 3.

Figure 4: Sample "best" results obtained.