PyDiet - Terms of Reference

[#project-pydiet]

Proposed Title

An Application of Computational Intelligence to Meal Optimisation

Background

> Why do people want to optimise their diets?

Diet has a powerful influence on many physiological and psychological characteristics of people. Buying food often represents a significant portion of an individual's living costs. Therefore, many people attempt to optimise their diet against goals and within constraints to manipulate physiological, psychological and financial components.

> How can diet planning be framed as an optimisation problem?

Goals may include targeting particular macronutrient, micronutrient and energy content, and minimising the financial cost of ingredients. Since most ingredient combinations do not form viable meals, any optimisation is constrained to viable recipes and their respective tolerance to variation of ingredient ratios. Consequently, designing optimal recipes is a complex optimisation problem, and a specific case of the ***Knapsack Problem***.

> Why is computational intelligence potentially well suited to dietary optimisation?

Human nutrition specialists typically charge a fee to apply experience, knowledge and heuristic methods to solve the problem. Computational intelligence techniques have been applied to derive satisfactory solutions to many incarnations of the knapsack problem. Most people dislike eating the same meals, so may frequently require fresh solutions. Additionally, diet optimisation goals may change as a function of body weight, training intensity, proximity to competition and other factors.

Therefore, obtaining different solutions from a nutritionist can become expensive. Alternatively, people may attempt to create their own meal plans, which is time consuming. The opportunity cost of this time to the individual must be considered.

> How do you propose to address to apply computational intelligence to dietary optimisation?

I propose to apply computational intelligence techniques, such as evolutionary algorithms, to derive solutions to the dietary optimisation problem. While the precise nature of the optimisation algorithm is currently undetermined, it is perhaps not too premature to crudely outline the architecture as follows:

- *Recipe Selection* - This may be a search heuristic or a brute force search, to determine which recipes in the database provide the most scope to meet the user's requirements.

- *Recipe Manipulation* - This may be an evolutionary algorithm which attempts to tune the ingredient ratios within the pre-selected recipes, within specified tolerances, to approach the user's goals.

- *Recipe Selection* - The best candidate from the population is selected. Ingredient cost discrimination may be implemented here.

- *Recipe Scaling* - All ingredient quantities within the candidate recipe are scaled by a constant factor to meet the energy content the user has specified for the meal.

The software would work along the user's specification list, for each meal, on each day, repeating this process and compiling results.

A mechanism will prevent the software from consecutively repeating similar meals from similar specifications.

Proposed Solution Specifications

> The following is a ***description*** of the proposed output of the work.

> Clarify the difference between diet recommendation and meal optimisation.

A brief review of the literature suggests that current dietary optimisation systems fall into two categories, which must be distinguished.

The first category aims to inform the user of the diet they should be following, based on a description of their physical characteristics or medical symptoms. For the remainder of the work, this category will be termed ***dietary recommendation.***

The second category aims to inform the user of the foods they should be eating to adhere to a particular diet. The work proposed here falls into the second category. For the remainder of the work, this category will be termed ***meal recommendation.***

> An overview of what I propose to produce.

The author proposes to create a diet planning application using the Python language. To maintain focus on the optimisation task, the software shall be controlled via configuration files and a basic tkinter interface.

The software shall be capable of designing weekly meal plans with different dietary goals for each day.

The user shall be able to describe their dietary requirements for each day via configuration files.

There shall be a non-deterministic factor, to promote meal variation.

If possible, the non-determinism should be controllable by tuning discrimination parameters.

> What are the software ***inputs***?

The software shall accept macronutrient, micronutrient and caloric quantities describing the user's diet, on each day of the week.

The user shall be allowed to omit specifications for macronutrient and micronutrient quantities. In these situations, the software shall fall back on EFSA dietary reference values \cite{dietary-reference-values}.

> How this fall-back should work isn't particularly clear to me. Obviously the EFSA values won't match the full set of micronutrient values I have in the software.
>
> I think instead, a good approach would be to have some mandatory nutrients, which, if not populated, the software will not run. Any additional targets beyond those will be included in the optimisation, but the algorithm can run without them. See [algorithm implementation]

The software shall accept a cost-sensitivity parameter, allowing the user to balance meal variation against the financial cost of ingredients.

> The effectiveness of the cost-variation compromise strategy will depend on the ability of the software to locate the true global optimum. If the algorithm routinely fails to find more cost-effective ***and*** more nutritionally optimal recipes, then the principle would be undermined.

The program will operate on a database of recipes, to be populated gradually by the author over the duration of the project (see [Time Plan](#Time Plan)).

Each recipe will specify a list of ingredients, ingredient ratios and their respective tolerable variations.

> Recipes (in this context) do not specify a total final quantity of food, so they are essentially collections of ingredient ***ratios***.

The variation tolerance will describe the allowable variation of the ingredient mass against the total mass of the meal before the meal becomes impracticable.

The macronutrient and micronutrient properties of each ingredient will be stored in a separate database, thus preventing information duplication across recipes including the same ingredient.

> What are the software outputs?

The program shall output a text file presenting a summary of the optimised meals for each day of the week, a weekly ingredient shopping list, and estimated costs.

Academic Objectives

> This section describes what I hope to learn throughout the work.

> What are the abstract technical skills I hope to develop and learn?

Primarily, the author hopes to develop the ability to apply computational intelligence techniques to real-world optimisation problems. The opportunity to practice architecting a non-trivial software solution is of particular value.

Additionally, the author wishes to exercise the capability of applying research methods to current problems.

> What are the secondary practical skills you hope to develop and learn?

Furthermore, the process of developing and documenting solutions to a relatively complex problem will provide opportunities to develop secondary skills required to support academic activity.

These skills include academic writing skills, proficiency with the LaTeX typesetting system and the Python programming language, and practice using numerous supporting software packages.

Research Questions

Primary Questions

The primary research questions are focussed on the technical aspects of developing a meal optimisation algorithm. The author hopes to identify a suitable computational intelligence strategy.

From here, additional evaluative questions will arise, regarding the relative strengths and limitations of candidate algorithm designs. These evaluative questions will, in turn, require answers to explanatory questions, for example: "Why can algorithm A often locate more optimal meals than algorithm B?".

Secondary Research Questions

Collaterally to the technical aspects of development, the author has an interest in the opportunities for the commercial deployment of a meal optimisation program. As such, the work aims to address the comparative question around the differences, from a user's perspective, between a qualified human nutritionist, and a software solution capable of emulating a human nutritionist.

Furthermore, the work will begin to identify the most practical ways a user can describe their nutrition goals to a software system. As mentioned previously, at this stage, the system will only incorporate a very basic GUI component, but will still be capable of investigating quantitative and practical methods for the user to describe nutritional goals, and extend the recipe and ingredient database.

Background Objectives

> To conduct a decent literature review.

The author proposes to conduct a review of the existing literature, directly related to the application of computational intelligence to dietary optimisation. The author's initial reading suggests the literature approaches dietary optimisation from various directions and with numerous distinct motivations.

> To understand which CI techniques apply well to the different parts of the problem.

Conducting a review of the literature provides an opportunity to develop an understanding of the strengths and shortcomings of these approaches regarding their specific application to dietary optimisation.

Moreover, since problems in CI typically belong to abstract categories, the work will provide an opportunity to develop an understanding of different techniques against different categories.

> Understanding the commercial and practical motivations behind dietary optimisation.

In addition to the technicalities of dietary optimisation, the author wishes to examine the motivations and implications behind developing a practical solution to dietary optimisation.

Specifically, if a practical and deployable solution were to be developed:

- Where would the target markets lie?
- What would be the criteria for practicality?
- Where would the ingredient and recipe data come from?
- How could a single system meet the requirements of different markets, such as medical and athletic?
- How could such a system be deployed?

Resources Required

No non-standard resources are required.

Risk Assessment

> Risks associated with the usage of the finished product.

A number of risks could conceivably be associated with the finished software.

Several scenarios could lead a user to attempt to follow an inappropriate dietary recommendation. These include logical faults within the algorithm, incorrect goals and incorrect data in the ingredient database. These risks can be mitigated by informing potential users that, at present, the software is for research purposes only, and any meal recommendations should not be implemented without prior consultation with a qualified nutritionist. Furthermore, the program will be designed to insert this cautionary statement into the results output page.

> Risks associated with accidental loss of work.

Additionally, risks to the project include the accidental loss of data due to information technology failure. The project data can be roughly divided into two categories: research and documentation as notes and written prose, and code.

To protect the research and documentation from accidental loss, the author will take regular weekly backups, and utilise the version control and backup capabilities in Google Drive.

To protect the code for accidental loss, the author will take regular weekly backups and utilise the cloud-based code repository, GitHub.

> Risks associated with failing to complete the project on time.

Poor time management could lead to progress falling behind, and the completed work being unavailable at the submission deadline.

The author will mitigate this risk by creating and following a high-level progress plan (see Section \ref{sec:time-plan}.

Additionally, the author will maintain an appropriate level of communication with the supervising lecturer. Specifically, this communication will include making progress reports, and work in progress available as appropriate.

Time Plan

The time plan is available online here.