Modelling the Design of a Nutritionally Optimal Meal Plan on a Budget as a Partially Observable Markov Decision Process

MAAME AKUA KOME-MENSAH and HILLARY HERMAWAN
Stanford University, Symbolic Systems Program

0. ABSTRACT

Preparing nutritious meals on a budget poses a challenge to many households, from those of recent college graduates to those of busy families. In this paper, we leverage nutrition research to develop several partially observable Markov decision processes (POMDP), Fast-Informed Bound with one-step lookahead and Branch and Bound. We use these to model the design of a nutritionally optimal and cost-efficient meal plan. We find that Branch and Bound yields a meal plan with a high expected utility, while FIB provides a reasonable upper bound for the optimal value.

1. INTRODUCTION

There is a popular conception that healthy eating is not readily accessible because of factors like perceived cost (shopping at grocery stores such as Whole Foods that stock "healthy" foods such as kale and quinoa is often associated with more affluent households) and relative effort (cooking at home versus picking up a fast food meal). Being able to maintain a balanced and nutritious diet on a budget of both time and money may seem impossible. However, we believe that it can be done and that knowledge of this possibility needs to be publically available to encourage more widespread access to better health.

This classic problem of resource allocation has been explored extensively through linear programming. Here, we aim to model the problem instead as a partially observable Markov decision process (POMDP) to determine the fifteen meals a young adult should eat in a 5-day week while optimizing for cost and nutritional content.

2. RELATED WORK

There are several studies that explore the generation of optimal meal plans based on certain constraints. J. Bulka et. al uses artificial intelligence algorithms to model glucose and insulin fluxes in people with diabetes and prepare meal plans adapted to their tastes while staying within blood glucose limits [1]. SmartDiet [2] proposes an interactive diet consultant built on optimization algorithms that recommends nutrient-balanced meals by taking into account individual needs.

These works, and other works in the field, focus on specific health conditions and standalone meals. We hope to navigate more of the general health space, gearing our results towards households that prepare their own meals using ingredients from local markets. We also hope to generate meal plans that span a period of time to encourage flexibility, as households may purchase all their ingredients for a week or several weeks at a time.

3. DATASET AND PREPROCESSING

We acquired a dataset from Kaggle with information on the ingredients, nutritional content, and user ratings of around 10,000 meals, represented as recipes. For each meal (row), there is information on its protein, sodium, calorie and fat content and whether is can be eaten for breakfast, lunch, or dinner. Since the dataset did not contain cost information, we compiled data on costs of the ingredients manually to calculate the costs of preparing each meal. We deleted samples that had poorly recorded ingredients and nutritional content information, as well as those that required ingredients that are difficult to find in the average supermarket, such as venison.

In order to formulate a reward function, we conducted research into optimal nutrition and nutritional decay. We studied suggested daily nutrition intakes of different demographics, compiling tables of daily nutritional requirements and noting optimization opportunities in recommended total fat and sodium ceilings. As an potential time factor in our model, we also looked into how nutrients decay in the body, which involved considering the basal metabolic rates involved in food intake.

4. MODELING THE PROBLEM AS A POMDP

Our goal is to generate a 5-day meal plan for a young adult. We assume a finite horizon of 15 steps, as the adult eats 3 times a day. At each time step (mealtime), we observe the young adult as happy or unhappy and suggest a meal for the current time step based on our observation. Since each action is a meal we recommend, there are 2,620 possible actions.
However, the time of the day restricts the meals we can suggest to the adult. For example, the adult must have breakfast at the first time step, lunch at the second, and dinner at the third. This sequence of breakfast-lunch-dinner repeats till we get to the 15th step, which represents dinner at the end of the work week. The young adult can be in any one of seven states that are not directly observable to us: healthy, low_protein, high_protein, low_sodium, high_sodium, low_fat and high_fat. These states are a noisy indication of the health of the adult.

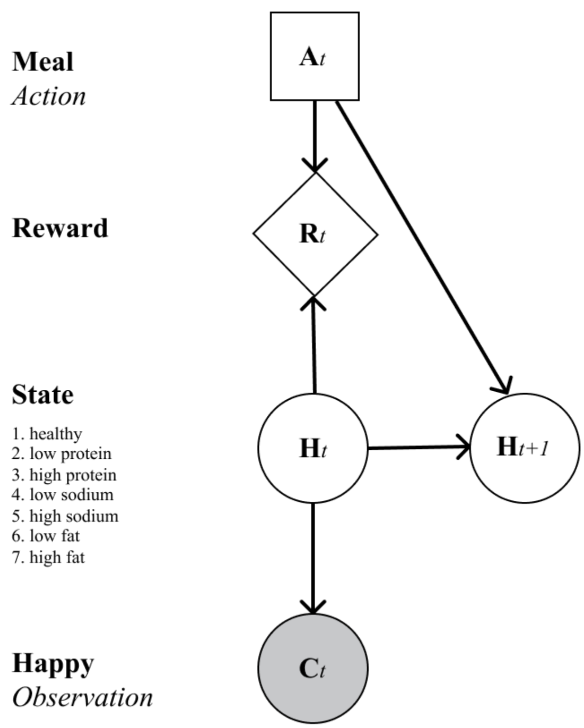


Figure 1. The problem structure, modeled off of the crying-baby problem [3].

We built our observation model $O(o \mid s, a)$ and transition matrix $T(s' \mid s, a)$ based on the nutritional value of each action (meal) and used these to update our belief about the states according to the equation

$$b'('s) \propto O(o \mid s \ , \ a) \Sigma_s \, T(s' \mid s, \ a) b(s) \qquad \text{(Eq. 1)}$$

We used Python to generate sample action-observation pairs based on the reward function we formulated. Since our actions are the meals we decide to take, the reward from each action is the nutritional value obtained from the particular meal. Our reward function assigns positive rewards to meals that were within a certain distance from required daily amounts regarding cost and nutritional breakdown and negative costs to meals that were not. It also assigns positive value to meals that are reasonably priced. To encourage exploration, we ensure that our model does not recommend a meal more than once.

Our simulator observes a happy adult when the reward is a non-negative integer and an unhappy adult otherwise. Of our 2,620 actions, 612 yield a "happy" observation. Although this is unbalanced, we did not delete any of the surplus samples because we wanted to be able to choose from a wide variety of meals.

5. METHODS

5.1. FIB with One-Step Lookahead

We first executed a simple QMDP technique which created a set of alpha vectors based on the value function $Q(s, a)$ under full observability. Since QMDP is an offline method, it is executed prior to placement in the environment. Using value iteration, we initialized $\alpha_a^{(0)}(s) = 0$ for all s and iterate:

$$\alpha_a^{(k+1)}(s) = R(s, \ a) + \gamma \sum_{s'} T(s' \mid s, \ a) \, max_{a'} \alpha_{a'}^{(k)}(s') \qquad \text{(Eq. 2)}$$

We used these alpha vectors to approximate the optimal value function and used one-step look ahead at each step to calculate the optimal policy:

$$\pi(b) = arg\ max_a \left[R(b, \ a) + \gamma \, \Sigma_o \, P(o \mid b, \ a) U(UpdateBelief(b, \ a, \ o)) \right] \qquad \text{(Eq. 3)}$$

The utility of a policy was calculated using the alpha vector for that action and the associated belief. The optimal policy for 15 steps yielded a total utility of 1.28×10^{19}; this raised some concerns because it appeared abnormally large. We used Fast Informed Bound (FIB) in an attempt to attain a tighter bound of this value. FIB with one-step lookahead yielded a utility of 9.7×10^{14}, so we expected the optimal value to be just a little lower than this value.

5.2. Branch and Bound Forward Search

We then implemented forward search, an online method. We expected faster performance, as this method only computes values for states that are reachable from the current state in the environment. We used the following equation to evaluate the value of an action a and chose the action that returns the highest value for a given depth d:

$$R(b, \ a) + \gamma \, \Sigma_o \, P(o \mid b, \ a) U_{d-1}(UpdateBelief(b, \ a, \ o)) \qquad \text{(Eq. 3)}$$

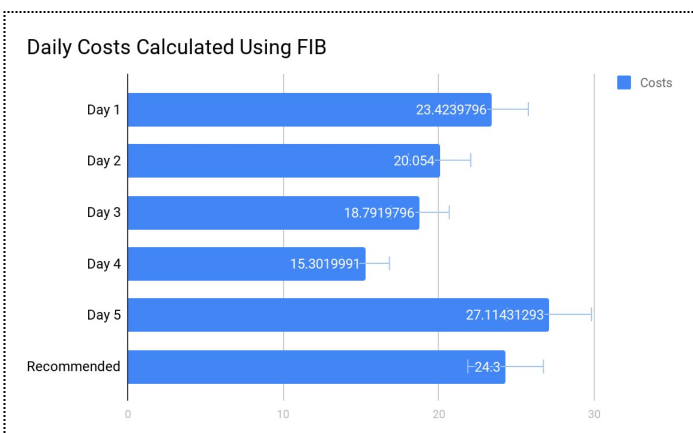
Forward Search, however, was very slow because of the large number of possible actions. To improve upon this performance, we implemented Branch and Bound policy search to prune the less useful searchspace. We expected Branch and Bound to greatly improve upon Forward Search because most of the meals in our data are unhealthy; the meals contain more nutrients than is recommended. To efficiently prune the searchspace, we arranged the actions in order of decreasing upper bounds. We calculated the upper bounds using FIB for tightness and we used a blind policy for the lower bound. Though Branch and Bound was much quicker than ordinary Forward Search, it was still slow. It yielded an optimal value of 6.6×10^9 which was notably less than the value obtained from FIB.

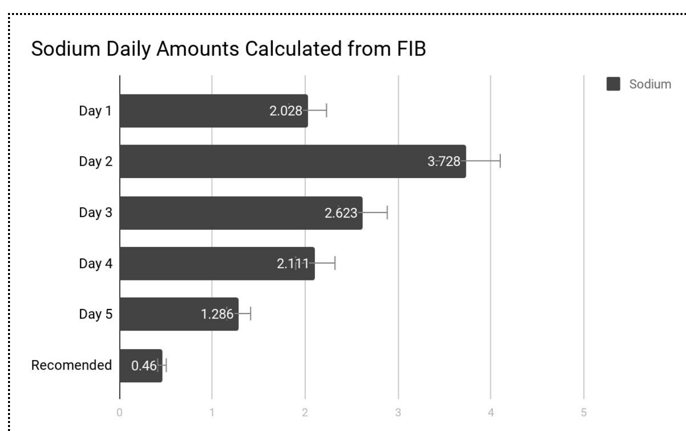
6. RESULTS

After comparing optimal meal plans generated from different implementations of POMDPs, we found our most optimal meal plan through Branch and Bound [Figure 2]. Interestingly, although all our methods yielded positive utility, the compositions of the meal plan varied greatly, with very few meals suggested by more than one method. This is perhaps due to the large number of meals in our dataset. We find that as the days go on, the model better fits the daily intake of nutrients to the recommended amounts.

Meal	Day 1	Day 2	Day 3
Breakfast	Ham and Swiss Puff-Pastry Quiche	Eggs Carbonara with Basil and Parmesan	Whole-Wheat Pancakes with Blackberry Syrup
Lunch	Shrimp Cakes with Andouille Sausage	Carrot Cardamom Soup	Cheddar Potato Soup with Bacon
Dinner	Coriander Chicken Tostadas with Refried Beans and Grilled Fennel	Sheet-Pan Chicken Saltimbocca With Roasted Potatoes and Crispy Kale	Springtime Pasta Primavera

Figure 2. The first three days of the optimal meal plan proposed by Branch and Bound.





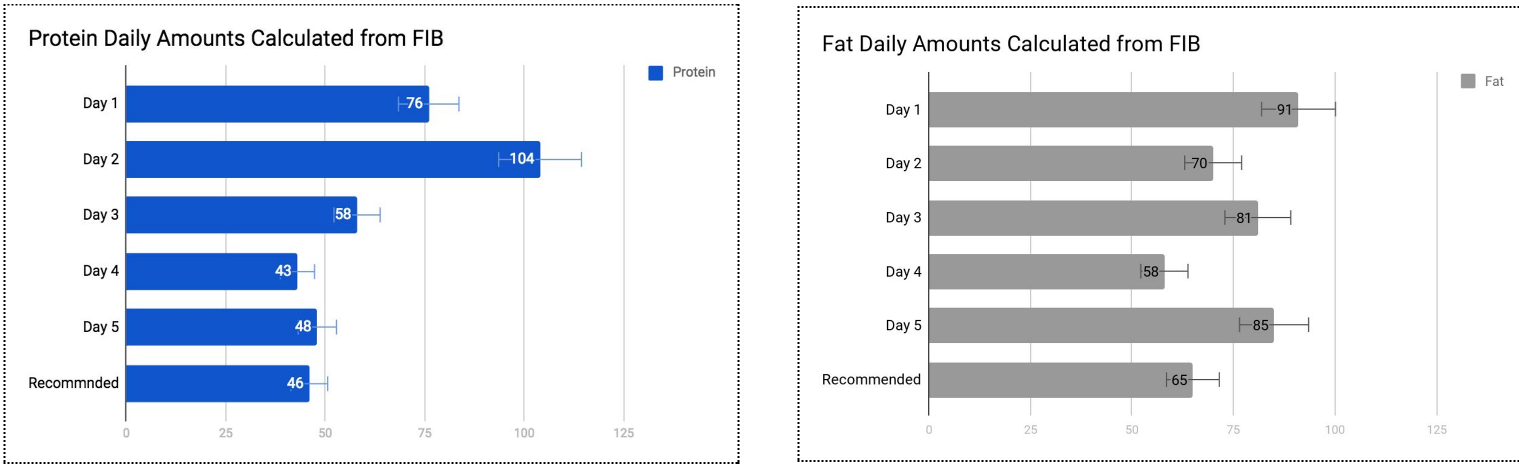


Figure 3. Daily nutrition intake provided by the optimal meal plan, compared to recommended amounts.

7. DISCUSSION

The results of this work suggest that there exists multiple possible meal plans that range in relative utility and, depending on the recipe pool, vary widely in content, providing flexibility and variety. We imagine that our model can be adapted in terms of different budgets, nutritional needs, and possible meals, and these different inputs will output reactive results.

One possible application for this work, as explored in SmartDiet [2], is incorporation into a mobile application that households can use to plan out their meals. Another possible application is incorporation into a tool for systems that require producing food for large amounts of people on budget constraints, such as school cafeterias.

We have made many assumptions in the course of designing our model for simplification purposes, notably that prices of meal ingredients would not fluctuate significantly from store to store, that there is a standard amount of an ingredient in all recipes, and that there is one optimal nutritional mix. For future consideration, we hope to extend this work the following ways: revisit the assumptions we have made and refine them to improve the accuracy of our model and results; introduce level of meal enjoyment, measured through recipe ratings, as a factor to be optimized; consider how ingredients can be combined and used across recipes - i.e. buying one dozen eggs and using varying numbers of eggs in different meals over the span of a week; and adapt our model to account for varying nutritional needs - i.e. those of a grown adult man versus those of a child.

8. WORK BREAKDOWN

We worked together to develop and refine the problem and its scope in order to fit it to POMDP constraints, which involved several iterations of brainstorming; researching the feasibility, details, and implementation of ideas; and getting feedback from course assistants. We developed the reward function, grouped food items and manually inputted costs for our dataset, and trained and evaluated policies together.

Maame acquired and cleaned the dataset; wrote code for the observation model, simulator, FIB with One-Step Lookahead, and Forward Search with Branch and Bound; wrote certain sections of the final paper; and outlined the final paper. Hillary conducted nutrition research and surveyed relevant literature; wrote code for the Monte Carlo tree search; wrote certain sections of the final paper; and edited the final paper.

9. REFERENCES

[1] J. Bulka et al. “Automatic meal planning using artificial intelligence algorithms in computer aided diabetes therapy.” International Conference on Autonomous Robots and Agents, 2009.

[2] Hisao, Jen-Hao and Henry Chang. “SmartDiet: A personal diet consultant for healthy meal planning.” International Symposium on Computer-Based Medical Systems, 2010.

[3] Kochenderfer, M.J. *Decision Making Under Uncertainty: Theory and Application*. MIT Press, 2015.

[4] 2005. Health Facts. Available at: https://health.gov/dietaryguidelines/dga2005/toolkit/healthfacts/nutrition.htm. [Accessed November 21, 2018].

[5] 2018. Daily Nutrient Requirements Calculator. Available at: https://www.eatforhealth.gov.au/page/eat-health-calculators/calculated/1543034779. [Accessed November 21, 2018].

[6] Goodman, Barbara E. “Insights into digestion and absorption of major nutrients in humans.” *The American Physiology Society*, 2010.

10. APPENDIX

Please refer to our code at https://codeshare.io/24QvX3.