Personalized Diabetes Management

by

Allison Kelly O'Hair

B.S. Mathematics, University of California Davis (2009)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author . . . .
Sloan School of Management
May 17, 2013

Certified by . . . .
Dimitris Bertsimas
Boeing Leaders for Global Operations Professor
Co-Director, Operations Research Center
Thesis Supervisor

Accepted by . . . .
Patrick Jaillet
Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center

Personalized Diabetes Management

by

Allison Kelly O'Hair

Submitted to the Sloan School of Management
on May 17, 2013, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

In this thesis, we present a system to make personalized lifestyle and health decisions for diabetes management, as well as for general health and diet management. In particular, we address the following components of the system: (a) efficiently learning preferences through a dynamic questionnaire that accounts for human behavior; (b) modeling blood glucose behavior and updating these models to match individual measurements; and (c) using the learned preferences and blood glucose models to generate an overall diet and exercise plan using mixed-integer robust optimization.

In the first part, we propose a method to address (a) above, using integer and robust optimization. Despite the importance of personalization for successful lifestyle modification, current systems for diabetes and dieting do not attempt to use individual preferences to make suggestions. We present a general approach to learning preferences, that includes an efficient and dynamic questionnaire that accounts for response errors, and robust optimization models using risk measures to account for the commonly seen human behavior of loss aversion.

We then address part (b) of our system, by first modeling blood glucose behavior as a function of food consumed and exercise performed. We rely on known attributes of different foods as well as individual data to build these models. We also show how we use optimization to dynamically update the parameters of the model using new data as it becomes available.

In the third part of this thesis, we address (c) by using mixed-integer optimization to find an optimal meal and exercise plan for the user that minimizes blood glucose levels while maximizing preferences. We then present a robust counterpart to the formulation, that minimizes blood glucose levels subject to uncertainty in the blood glucose models.

We have implemented our system as an online application, and conclude by showing a demonstration of the overall program.

Thesis Supervisor: Dimitris Bertsimas
Title: Boeing Leaders for Global Operations Professor
Co-Director, Operations Research Center

Acknowledgments

I gratefully acknowledge my advisor Dimitris Bertsimas for the critical role he played in the production of this dissertation. In addition to helping guide this research, he has helped me to find a career path that I am passionate about, and has challenged me to achieve goals that I never thought would be possible. I am truly honored to have him as my advisor.

In addition, I acknowledge the other MIT faculty and staff who have taught and supported me. In particular, I would like to thank Georgia Perakis and Karen Zheng for being on my thesis committee and for serving as such great role models not only for me, but for many other students.

I would also like to thank the other members of the LiA team: Manolis Tsangaris and Sypros Potamianos for their help in implementing the online software; and Anna Jasonides for her invaluable nutrition and dietary advice. Thank you for believing in this work as much as I do.

My experience at MIT would not have been the same without my friends and colleagues at the ORC. In particular, I would like to thank: Vishal Gupta for serving as a very thoughtful sounding board when I was stuck in my research; Andre Calmon for encouraging me to pursue a research project I am passionate about; Ross Anderson and Adam Elmachtoub for always making me laugh; Kris Johnson, Angie King, and Fernanda Bravo for their invaluable friendship and support; and John Silberholz and Stephen Relyea for being such a fun and inspiring research team.

Finally, I am extremely grateful to my family for all of their support. They have been my support system and my motivation throughout my time here. Thank you: to my Mom for her incredible advice and ability to listen and help me through anything; to my Dad for inspiring me to pursue mathematics and computer science, and for reminding me to work hard, but have fun too; and to my sister, who has always been my best friend and “my person”. And thank you to Alex, who has inspired, challenged, and encouraged me to aim higher and achieve more.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1998, about 10.4 million people in the United States had diabetes [21]. Today, about 25.8 million people in the U.S. have diabetes (8.3% of the population), and by 2050, it is projected that as many as one in three U.S. adults could have diabetes [24]. These statistics also reflect the growth of diabetes internationally. An estimated 285 million people had diabetes worldwide in 2010, and it is predicted that as many as 438 million people will have diabetes by 2030.

This alarming increase in diabetes rates is due to several factors, including the current obesity epidemic. Figure 1-1 shows how the obesity rates in the United States have changed in the past twenty years. In 1990, no state had an obesity rate of greater than 15%. Today, no state has an obesity rate of less than 20%, and many states have an obesity rate of greater than 30%. Type II diabetes, the type that is causing the majority of new cases, is known to be an obesity-related condition. Because of this, the number of people with type II diabetes is projected to continue increasing, and will only get worse as the obesity epidemic continues.

Diabetes has very serious complications if it is not controlled, including heart disease, kidney disease, blindness, and amputation. It was also responsible for $245 billion in healthcare costs in 2012 [2]. It has become a national priority to help reduce the number of new cases of diabetes and obesity. The 2013 Standards of Medical Care for Diabetes recommends weight loss, regular physical activity, and dietary changes to individuals with prediabetes or diabetes [8]. Additionally, there is evidence that low-glycemic index diets can improve blood glucose control in people with diabetes, as well as help prevent the development of type II diabetes [22, 70].

Motivated by the severity of diabetes and obesity, the focus of this thesis is a comprehensive system for personalized diabetes and diet management. We believe that optimization and analytics can help treat and prevent diabetes and obesity. It can be challenging for medical professionals to give personalized advice to patients and to give day to day guidance to help them manage their health. On the contrary, an interactive online system that uses data, optimization, and analytics has the power to make personalized suggestions on a patient by patient basis.

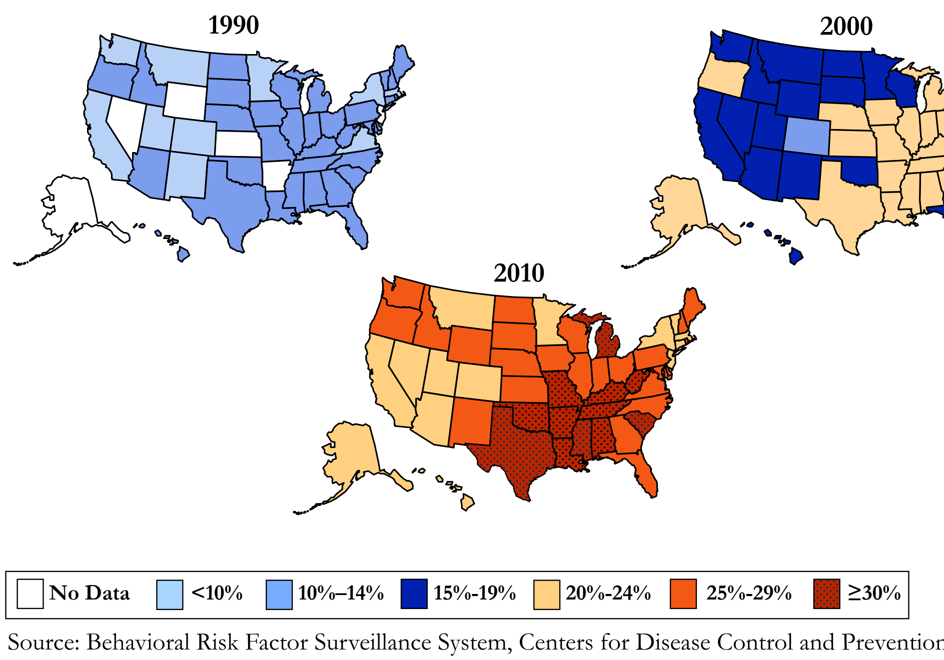


Figure 1-1: Obesity trends among U.S. adults, from 1990 through 2010. The color of each state gives the percentage of adults who are considered obese (BMI $>$ 30) in that state.

1.1 Contributions

In this thesis, we propose a mixed-integer robust optimization approach for personalized diabetes and diet management that suggests meals and exercise times to individuals that are customized to their preferences, attributes, and blood glucose measurements. We minimize blood glucose levels, while maximizing preferences and accounting for nutritional constraints. Our approach includes the following key steps:

(a) We generate personal preferences for the user based on their responses to carefully selected comparison questions.

(b) We develop models to estimate how a person's blood glucose levels will change after eating and exercising. We then use optimization to update our knowledge of the individual's blood glucose behavior using data from the user.

(c) Using the blood glucose models, the user's preferences, and a comprehensive recipe and nutritional database, we generate a daily meal and exercise plan using mixed integer optimization. We incorporate robust optimization into the plan generation to account for uncertainty in the blood glucose parameters.

The most important contribution of this work is combining these individual pieces into a comprehensive system for diabetes and diet management. We have implemented this system as an online application, and use this implementation to provide empirical evidence for the strength of our methodology. In this chapter, we give an introduction to the contents of the subsequent chapters.

In Section 1.2, we give a literature review of the field of preference learning, and introduce the methodology we have developed to learn preferences, which is the focus of Chapter 2. In Section 1.3, we introduce Chapter 3, specifically, the idea of modeling blood glucose dynamics and an optimization approach to updating the model parameters, and the recipe database that we constructed. In Section 1.4, we introduce the idea of using mixed-integer optimization to find a diet and exercise plan, which is described in detail in Chapter 4. We also discuss previous work that has been done in this area. We give a demonstration of the overall system and provide concluding remarks in Chapter 5.

1.2 Learning Preferences

While it is widely believed that diabetic patients with good eating and exercise habits can significantly improve the state of their disease, it is also known that patients with diabetes are particularly prone to regimen adherence problems [29]. It has been argued that one way to improve adherence rates may be to increase the number of diet options available, and to better match individual food preferences [26].

For these reasons, in Chapter 2 we present our approach to learning individual food preferences. To learn preferences, we wanted our system to have the following characteristics: (a) it should be efficient and quick for the user to complete; (b) it should be natural and enjoyable for the user to answer the questions; and (c) it should account for observations that have been made about human behavior.

Learning preferences or utilities for items has been a topic of interest for many years. The goal is to learn user preferences or utilities for a set of items given limited information, with the final outcome being a recommendation made to the user for the item with the highest utility. Some of the earliest work in preference learning was in the field of economics, in which the emphasis was placed on learning how people behave when faced with a choice between different options. One of the original theories was that of expected utility theory, which claimed that rational people maximize the expectation of a utility function [13, 82]. Since then, there have been several influential works disputing the claims of expected utility theory, and instead claiming that users behave in what could be considered "irrational" ways [5, 49, 79]. Evidence given in this research shows that when describing their preferences, users are often inconsistent, loss averse, influenced by the framing of the questions, and define outcomes with respect to a reference point.

The field of preference learning has also recently become more popular in machine learning and operations research due to its importance in many web applications, including search and recommendation systems [31, 37]. The emphasis here is not placed on observing how people behave, but on how to build a preference model given limited data on preferences. This model is often built to predict revenues from offering a subset of products to customers, or to rank a fixed set of options [35, 46]. Preference learning is also used to find weights for multiple criteria optimization problems, or multiple criteria decision-making (MCDM) problems [88]. Several interactive approaches to learning weights in MCDM problems have been proposed [57], and there is some work on incorporating robustness into these approaches [28, 69, 85]. However, there has been limited work on incorporating human behavior into the preference learning and final weight selection [33, 84].

While there have been many methods proposed to learn preferences, we feel that there is a need for a systematic and comprehensive methodology to algorithmically derive preferences and ultimately make suggestions to users that adhere to human behavior. Our approach uses an adaptive comparison questionnaire to find utilities for the attributes of items. The idea of using an adaptive questionnaire comes from the marketing literature [77, 78]. Adaptive questionnaires are increasingly being used, but the results often suffer from response errors to early questions that influence the selection of later questions. Some work has been done using complexity control or a Bayesian framework to be robust to response error [1, 76]. We instead address the inconsistency and response errors of users through a self-correcting mechanism that uses integer optimization. This accounts for the fact that while transitivity (if $\mathbf{x} > \mathbf{y}$ and $\mathbf{y} > \mathbf{w}$ then $\mathbf{x} > \mathbf{w}$) is rational and ideal, people are often inconsistent. Additionally, people may incorrectly answer questions, resulting in response errors that add noise to the data.

In order to model the commonly seen human behavior of loss aversion, we solve a robust optimization problem to optimize the worst utility within the uncertainty set of utilities. This differs from previous work, in which the standard approach was to select a particular estimate for the utilities out of all possible choices in the feasible set. Additionally, we propose a robust optimization method to control the tradeoff between robustness and optimality. In this method, we maximize the Conditional Value at Risk ($CVaR$) of the utility function.

In our empirical study at the end of Chapter 2, we use $CVaR$ to compute the loss aversion of each method quantitatively, which demonstrates the edge our methods have over the traditional approach. We feel this work also makes the following methodological contributions: (a) to the robust optimization literature by proposing to derive uncertainty sets from adaptive questionnaires, (b) to the marketing literature by using the analytic center of discrete sets (as opposed to polyhedra) to capture inconsistencies and errors, and (c) to the risk modeling literature by using efficient methods from computer science for sampling to optimize $CVaR$.

1.3 Blood Glucose Modeling and Nutritional Data

In Chapter 3, we describe models we have developed to capture blood glucose dynamics. The monitoring of blood glucose is critical for the treatment of diabetes, due to the nature of the disease. Type II diabetes is characterized by an insufficient production of insulin; either the body does not produce enough insulin, or the cells ignore the insulin that is produced. Typically, the body breaks down sugars and starches into glucose and then uses insulin to absorb this glucose from the bloodstream into the cells. In someone with diabetes, this glucose stays in the bloodstream which causes abnormally high blood glucose levels. This can lead to very serious complications, including glaucoma, cataracts, amputation, kidney disease, nerve damage, and heart disease [2]. If a person with type II diabetes can successfully lower their blood glucose levels through diet and exercise, they not only lower their risk for these complications, but they can avoid the side-effects of the various diabetic medications.

Our models predict the blood glucose levels of a person, given the food they have consumed and the exercise they have performed. The models use parameters that have initial starting values, defined using the glycemic index and medical studies, but can be personalized to the individual [36, 40, 65, 67, 75, 25, 64]. These medical studies have measured blood glucose levels of individuals and provided aggregate blood glucose curves, but to our knowledge, no one has developed mathematical models to capture the blood glucose dynamics.

We then propose a way to update the blood glucose parameters used in the modeling of blood glucose dynamics. For each user, we store a set of blood glucose parameters to capture how their blood glucose changes after exercising or eating certain foods. While we can initialize these parameters to default values, we would ideally like them to match the user's personal blood glucose measurements. In this chapter, we provide a method for adjusting the parameters to reflect new measurements provided by the user. We present computational evidence to argue that this technique successfully learns the user's true parameters, even in the presence of noise.

Lastly, we discuss a comprehensive nutritional database that we have constructed. This includes over 600 recipes, nutritional information for each ingredient and each recipe, and other attributes of the recipes such as the glycemic index. We also discuss the nutritional guidelines we use to ensure that we recommend nutritious meal plans. The database and nutritional guidelines were constructed with the assistance of a registered dietician and nutritionist.

1.4 An Optimization Approach to Personalized Diabetes and Diet Management

In Chapter 4, we use the preferences generated using our methodology of Chapter 2, and the blood glucose models and the comprehensive food database described in Chapter 3, to generate an overall meal and exercise plan for the user. We do this using mixed-integer optimization.

We then present a robust optimization counterpart of the mixed-integer optimization problem for finding an eating and exercise plan. The robust optimization problem is robust to the parameters defining the blood glucose models, to add protection against high blood glucose values. With the robust approach, we are able to find a meal plan that is almost as appealing to the user, but keeps the blood glucose values even lower than in the nominal problem. We introduce a parameter to control the tradeoff between robustness and optimality. We first present the optimization formulations, and then give empirical evidence for the strength of both models.

To the best of our knowledge, work of this kind has not been done before. There has, however, been a significant amount of work in the area of diet optimization where the goal is to find a solution that minimizes cost, what is commonly referred to as the "diet problem". Researchers studied the diet problem in the 1940s, when the Army wanted to meet the nutritional requirements of soldiers while reducing the cost. The basic statement of the problem is to decide how much of different foods to eat on a daily basis so that nutritional intakes are at least equal to the recommended dietary allowances and so that the cost of the diet is minimal [73]. They were not concerned with the appeal or appropriateness of the solution, just with the cost and nutritional requirements. This problem was not very useful for many human diet problems for which cost was not critical, but it led to a wide variety of successful applications of linear optimization to deal with cattle and chicken feed, fertilizer, and general ingredient mixing problems. To make the meals more appealing in terms of preferences and variety, many others have developed extensions of the basic problem [11, 39]. However, this problem minimizes cost as the objective function, while we are interested in minimizing blood glucose levels and maximizing preferences.

The importance of incorporating uncertainty into the diet problem is discussed by Mulvey, Vanderbei and Zenios [59]. They consider nutritional content uncertainty, the importance of which is also mentioned by Stigler and Dantzig [27, 73]. Stochastic programming and artificial intelligence techniques have also been proposed to deal with the uncertainty in optimization models of diet problems [52, 66, 54]. We instead use the robust optimization approach proposed by Bertsimas and Sim [16] to model uncertainty in the blood glucose models.

There have also been some medical studies in optimizing blood glucose control with insulin therapy for type I diabetics [32, 12], or measuring blood glucose deviations after consuming different foods for type II diabetics [36, 40, 65, 75, 67]. The only work that we have found that addresses the use of a personalized diet and data to reduce blood glucose levels is an individual's efforts and tests to prove that this method worked for him personally [64]. In this chapter, we aim to extend his personal observations into a tool that uses optimization to find the best meal and exercise program for any individual. Our main contribution in this chapter is a method for generating a meal and exercise plan that is feasible and appealing in practice, and accounts for the preferences and blood glucose behavior of individuals.

Chapter 2

Learning Preferences Under Noise and Loss Aversion

In this chapter, we present an algorithmic approach to learning preferences. This work was motivated by the specific problem of diabetes and diet management, since people who are struggling with diabetes or obesity are particularly prone to regimen adherence problems [29]. It is believed that more appealing meal plans in terms of preferences could improve adherence rates [26].

However, this work is more general, and can be applied to many other areas in addition to food preferences. It could be used to learn user preferences for cars, vacations, or other items that can be described by a set of attributes. For this reason, we present our approach in general terms in this chapter, and then provide empirical evidence for the specific problem of food preferences at the end of the chapter.

This chapter is structured as follows. In Section 2.1, we discuss the topics that motivated this work and give an introduction to our overall approach. In Section 2.2, we introduce the adaptive questionnaire used to inform the feasible set of utilities and the mixed integer optimization model to address human inconsistencies and response error. In Section 2.3, we present a robust optimization approach to address loss aversion and a new $CVaR$ approach to provide less conservative robust solutions. In Section 2.4, we describe the online system we have implemented to learn preferences for our diabetes and diet application. Lastly, in Section 2.5, we give some empirical evidence to support the use of our strategy in practice.

2.1 Motivation and Overview

To learn preferences, we wanted our system to have the following characteristics: (a) it should be efficient and quick for the user to complete; (b) it should be natural and enjoyable for the user to answer the questions; and (c) it should account for observations that have been made about human behavior. In this section, we discuss the areas of research that motivated this work, and give an overview of our approach.

2.1.1 Behavioral Economics and Prospect Theory

Learning preferences or utilities for items has been a topic of interest for hundreds of years. Some of the earliest work in preference learning was in the field of economics, in which the emphasis was placed on learning how people behave when faced with a choice between different options.

This work started in the 18th century, when Bernoulli initiated expected utility theory by positing that rational behavior can be described as maximizing the expectation of a function $u(\cdot)$ defined on possible outcomes [13]. Two hundred years later, Von Neumann and Morgenstern [82] proposed four axioms of rationality under which expected utility theory holds:

- **Completeness.** When presented with two items A and B, either item A is preferred, item B is preferred, or there is no preference between the two items. This axiom assumes that an individual has well-defined preferences.

- **Transitivity.** If item A is preferred to item B, and item B is preferred to item C, then it must be true that item A is preferred to item C. This axiom assumes that preferences are consistent across any three options.

- **Continuity.** If item A is preferred to item B which is preferred to item C, then the user's preference for item B can be described as a weighted sum of the preferences of items A and C. This axiom assumes that there is a "tipping point" between being better than and worse than a given middle option.

- **Independence.** If the user prefers item A to item B, this preference order still holds regardless of any third item C that could be introduced. This axiom assumes that preferences hold independently of the possibility of another outcome.

Since expected utility theory was proposed, there have been several influential works disputing the axioms of expected utility theory, and instead claiming that users behave in what could be considered "irrational" ways [5, 49, 79]. Evidence given in these works shows that when describing their preferences, users are often inconsistent, loss averse, influenced by the framing of the questions, and define outcomes with respect to a reference point.

Specifically, Kahneman and Tversky observed that several of the axioms defined by Von Neumann and Morgenstern are inconsistent with human behavior, and proposed prospect theory as a psychologically more accurate description of preferences compared to expected utility theory [49]. They also extended prospect theory to riskless choice, for situations in which there is no uncertainty in the choices that have to be made [79]. They made the following observations:

- **Reference dependence.** Gains and losses are determined relative to a reference point. This differs from expected utility theory, in which a rational agent is indifferent to the reference point.

- **Loss aversion.** Losses hurt more than gains feel good. This is the strongest observation that Kahneman and Tversky made, and they observed that all of the empirical evidence they found supports the existence of loss aversion. They describe loss aversion as "a bias that favors retention of the status quo over other options," and as a reaction to change that is "expected to be more intense when the changes are unfavorable (losses) than when they are for the better."

- **Diminishing sensitivity.** The gain or loss of an extra unit results in a lesser impact as the amount of the overall gain or loss gets larger.

- **Framing effects.** The rational theory of choice assumes that equivalent formulations of a choice problem should give rise to the same preference order. Contrary to this assumption, the framing of questions to determine preferences matters. This relates to loss aversion in that the same difference between two options will be given greater weight if it is viewed as a difference between two disadvantages than if it is viewed as a difference between two advantages.

- **Human inconsistency.** Due to these behavioral observations, people are often inconsistent in their preferences.

The value function $v(\cdot)$ proposed by Kahneman and Tversky is depicted in Figure 2-1. It has three key characteristics: (1) it passes through the reference point; (2) it is asymmetric, representing loss aversion; and (3) it is s-shaped, showing diminishing sensitivity.

Prospect theory made a seminal contribution to behavioral economics and enlightened our understanding of preferences. While Kahneman and Tversky and others [19] have proposed methods to calculate utilities, we feel that there is a need for a systematic and comprehensive methodology to algorithmically derive preferences according to these observations about human behavior. We aspire to develop and implement this methodology to learn preferences and make decisions.

2.1.2 Conjoint Analysis

Preference learning is also very popular in marketing, in which preferences are typically learned through questionnaires [23, 42, 43, 78]. The understanding of consumer preferences is a central problem in marketing, and the most widely used method for doing so is by using conjoint analysis, or choice questionnaires.

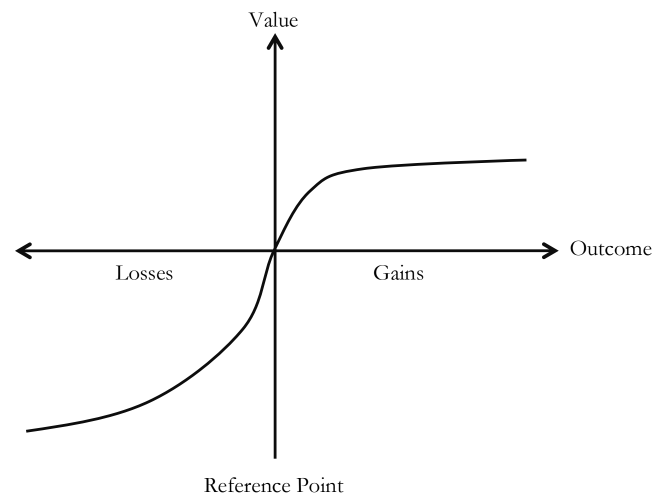


Figure 2-1: The value function in prospect theory.

It has been argued that choice questionnaires, ones in which the user is asked to pick between different options, are more natural than asking the user to rank items [44]. This inspired us to ask comparison questions to make our approach more natural and enjoyable for the users.

Additionally, the idea of using an adaptive questionnaire, or one in which the next question changes depending on the user's responses to previous questions, was recently introduced in the marketing literature [77, 78]. Adaptive questionnaires are increasingly being used, due to their ability to collect more information in a shorter amount of time, which is appealing to both the user (who has to spend less time on the questionnaire), and to the researcher (who gets more responses since the questionnaire is easier to complete). We will use adaptive choice-based questionnaires in our approach, and by doing so, extend this work.

2.1.3 Our Approach

We have developed a systematic and comprehensive methodology to algorithmically derive preferences and ultimately make suggestions to users that adhere to human behavior. In previous work, only one utility function or preference order is typically assumed as a result of the preference learning process, but there are often several that are consistent with the known data. We propose a methodology to robustly collect preference data from individuals, as well as ultimately make decisions using robust optimization. Since the selection of a single utility function or ranking order is often arbitrary, robust optimization improves on this approach by using all of the known information to make a decision. Our approach is based on robust and integer optimization, conjoint analysis and risk measures.

Our overall approach is as follows. We assume that any item or possible outcome $\mathbf{x}$ can be defined by attributes $x_1, \ldots, x_n$. We make the common assumption of a linear utility function $u(\mathbf{x}) = \mathbf{u}'\mathbf{x}$, for which we are trying to learn the weights $\mathbf{u}$. Prior to asking any questions, the weights $\mathbf{u}$ belong in an initial uncertainty set $\mathcal{U}^0 = [-1, 1]^n$. Thus, we consider a family of possible utilities

$$u(\mathbf{x}) = \mathbf{u}'\mathbf{x}, \quad \mathbf{u} \in \mathcal{U}^0.$$

We present the user with two items in the first question, $\mathbf{x}^1$ and $\mathbf{y}^1$, and ask the user to compare them, i.e., to tell us if he prefers item $\mathbf{x}^1$, prefers item $\mathbf{y}^1$, or has no preference (is indifferent). Based on his answer, we update the uncertainty set and use mixed-integer linear optimization to adaptively generate two new items to ask about in question two, $\mathbf{x}^2$ and $\mathbf{y}^2$. After a number k of such adaptively chosen questions, we have "decreased" the uncertainty set from $\mathcal{U}^0$ to $\mathcal{U}^k \subseteq \mathcal{U}^0$. At this point, we have a family of utilities

$$u(\mathbf{x}) = \mathbf{u}'\mathbf{x}, \quad \mathbf{u} \in \mathcal{U}^k.$$

Note that we address two separate types of potential noise in the data: human inconsistency and response error. Inconsistency refers to contradictory responses even though all of the responses are accurate. Response error refers to incorrect responses, which may not cause inconsistencies, but will increase the difficulty of capturing the user's true utilities. For these reasons, we also ask the user to indicate if they "feel strongly" about their response to a question to more accurately capture the incorrect answers.

In order to model loss aversion, we propose to solve the following robust optimization problem:

$$\max_{\mathbf{x}\in\mathbf{X}} \min_{\mathbf{u}\in\mathcal{U}^k} \mathbf{u}'\mathbf{x},$$

where X is the feasible space of outcomes, and k is the number of questions that have been asked. Note that we are taking the perspective that the adaptive process has reduced uncertainty to the set $\mathcal{U}^k$, and, as we are loss averse, we are optimizing the worst utility within the uncertainty set $\mathcal{U}^k$.

Additionally, we propose a robust optimization method to control the tradeoff between robustness and optimality. In this method, we maximize the Conditional Value at Risk ($CVaR_\alpha$) of the utility function, which is defined to be the expected value of the worst α% of the utilities [15, 50, 63, 68]. Since the feasible set $\mathcal{U}^k$ is a projection of a mixed integer set and therefore $CVaR_\alpha$ of the set is defined by an integral, we use random sampling of $\mathcal{U}^k$ to approximate $CVaR_\alpha$. Specifically, we use the "Hit-and-run" method of randomly sampling points from a convex body in $\mathbb{R}^n$, introduced by Smith [71]. This method has been shown to perform well in practice, and allows us to efficiently optimize $CVaR_\alpha$ [81].

$CVaR_\alpha$ is known as a second-order quantile risk measure, a concept that has been introduced in many ways by many different authors [9, 34, 62, 68]. We use $CVaR_\alpha$ and not standard deviation or Value at Risk (VaR_α) since we want to capture the amount of losses incurred. Standard deviation measures variation in both losses and gains, and VaR_α, or the α-quantile of the utilities, only captures the number of times you lose, not the severity of the losses. Additionally, neither standard deviation or VaR_α are coherent risk measures, meaning that they violate basic properties that capture the idea of risk. By adjusting the value of α, we are able to select how conservative we would like our robust solution to be.

Overall, we model human behavior in three main ways. First, we model inconsistent behavior and account for the importance of question framing in the way in which we learn individual preferences. Second, we incorporate loss aversion when making decisions for the individual. Lastly, we account for reference dependence in the overall design of the system. A quick and easy questionnaire to learn preferences means that we can ask the questionnaire repeatedly over time as the user's reference point changes.

In summary, our method uses integer optimization, robust optimization, $CVaR_\alpha$, adaptive conjoint analysis, and linear optimization. In our empirical study at the end of this chapter, we use $CVaR_\alpha$ to compute the loss aversion of each method quantitatively. We feel this work makes the following methodological contributions: (a) to the robust optimization literature by proposing to derive uncertainty sets from adaptive questionnaires, (b) to the marketing literature by using the analytic center of discrete sets (as opposed to polyhedra) to capture inconsistencies and errors, and (c) to the risk modeling literature by using efficient methods from computer science for sampling to optimize $CVaR_\alpha$.

2.2 Building Self-Correcting Utilities using Adaptive Questionnaires

In this section, we will address how we adaptively ask the user questions to learn a small set of possible utilities, while using integer optimization to account for the inconsistent behavior and response errors of people.

We will denote items by vectors of attributes with superscripts indicating the question. Thus, $\mathbf{x}^i = (x_1^i, x_2^i, \ldots, x_n^i)$ is the vector of attributes for one item (item $\mathbf{x}$) that is asked about in question i. The goal of the optimization problem is to suggest the best item for the user, given the utilities $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ for the attributes of the items. For example, the items could be different recipes, for which the attributes are the ingredients in the recipes, or the items could be different cars, for which the attributes are different features that can be selected (leather seats, navigation system, etc.). We will assume that there is a finite set of possible attributes, and that the user's utility function is linear in the attributes, $u(\mathbf{x}) = \mathbf{u}'\mathbf{x}$. Suppose we ask the user about two items in question i, $\mathbf{x}^i$ and $\mathbf{y}^i$. If the user indicates that they prefer $\mathbf{x}^i$ (or $\mathbf{y}^i$), then we say that $\mathbf{x}^i > \mathbf{y}^i$ (or $\mathbf{y}^i > \mathbf{x}^i$). If the user indicates that they are indifferent between the two items, we say that $\mathbf{x}^i = \mathbf{y}^i$. Note that this is the result of the user's response; since the user could make errors in their responses, this is the indicated preference of the user, not necessarily the true preference of the user.

We address the inconsistency and response errors of users through a self-correcting mechanism. This accounts for the fact that while transitivity (if $\mathbf{x} > \mathbf{y}$ and $\mathbf{y} > \mathbf{w}$ then $\mathbf{x} > \mathbf{w}$) is rational and ideal, people are often inconsistent. Additionally, people may incorrectly answer questions, resulting in response errors that add noise to the data. Another possible reason is that the user actually has conflicting preferences. For example, the user might indicate that she likes item $\mathbf{x}$ more than item $\mathbf{y}$ ($\mathbf{x} > \mathbf{y}$), item $\mathbf{y}$ more than item $\mathbf{w}$ ($\mathbf{y} > \mathbf{w}$), but item $\mathbf{w}$ more than item $\mathbf{x}$ ($\mathbf{w} > \mathbf{x}$). In this case, it is impossible to find utilities that are consistent with all three responses.

In the adaptive questionnaire, we start with no information about the user's utilities, and we would like to adaptively learn his or her utilities with a series of comparison questions. Furthermore, we would like to select the comparison questions to ask the user so that the space of different possible utility vectors is reduced as quickly as possible. Since the response to each of the questions is unknown, we would like to ask questions that give us the most information possible regardless of the response.

The method that we will describe here builds on the method described in [78, 77], but we use integer optimization to account for inconsistencies and response error in the process of selecting the next question. Additionally, we make some changes to the basic algorithm: we select the next question that minimizes the distance to the analytic center[1] of the remaining feasible space, we add an indifferent option, and instead of selecting a particular utility vector at the end of the algorithm, we keep the entire feasible space when we optimize over the utilities. We will discuss the reasons for these changes at the end of this section.

[1]The analytic center of a set of linear inequalities $a_i'x \leq b_i$, $i = 1, \ldots m$, is defined to be the solution of the following unconstrained optimization problem: $\min -\sum_{i=1}^{m} \log(b_i - a_i'x)$.

2.2.1 Modeling Questionnaire Responses with Inequalities

Suppose that we have already asked the user k comparison questions and received responses of the form $\mathbf{x}^i > \mathbf{y}^i$, $\mathbf{y}^i > \mathbf{x}^i$, or $\mathbf{x}^i = \mathbf{y}^i$. Consider the case where the user indicates that they prefer $\mathbf{x}^i$ to $\mathbf{y}^i$ ($\mathbf{x}^i > \mathbf{y}^i$) in question i. We would like the total utility of item $\mathbf{x}^i$ to be larger than the total utility of item $\mathbf{y}^i$, or $\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) > 0$. We model the strict inequality here by using a small number $\epsilon > 0$, so we have the constraint $\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) \geq \epsilon$. However, we would like to account for the possibility of inconsistencies or response error. We do this by introducing a binary variable ϕ_i for question i. Instead of using the constraint above, we use the constraints

$$\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) + (n + \epsilon)\phi_i \geq \epsilon,$$
$$\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) + (n - \epsilon)\phi_i \leq n,$$

for question i if the user indicates that $\mathbf{x}^i > \mathbf{y}^i$, where n is the total number of possible attributes. Similarly, we use the constraints

$$\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) - (n + \epsilon)\phi_i \leq -\epsilon,$$
$$\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) - (n - \epsilon)\phi_i \geq -n,$$

for question i if the user indicates that $\mathbf{y}^i > \mathbf{x}^i$. In both sets of constraints, if $\phi_i = 0$, then the utility vector is consistent with the user's response to question i (the first constraint enforces the correct response and the second constraint is redundant). If $\phi_i = 1$, then we "flip" the constraint and assume that the user either introduced an inconsistency or made a response error (the first constraint becomes redundant and the second constraint forces the inequality to flip).

If the user indicates that they are indifferent to question i ($\mathbf{x}^i = \mathbf{y}^i$), then we add the constraints

$$-\epsilon \leq \mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) \leq \epsilon.$$

In this case, we do not allow the inequalities to flip, since we assume that an indifferent response is more likely to be truthful. Even if the user actually prefers one item over the other, the difference in utilities is probably small, which is captured here. If an indifferent response is given for question i, we set $\phi_i = 0$. We will set $\epsilon = 0.1$ for the empirical evidence given in Section 4, but our model is robust to the value of ϵ.

Since we expect the user to respond incorrectly to only a small fraction of the questions, we also use the constraint

$$\sum_{i=1}^{k} \phi_i \leq \gamma k,$$

where γ is a parameter that indicates the maximum fraction of responses that we allow to be incorrect. For example, if $\gamma = 0.1$, we allow at most 10% of the user's responses to be flipped. In this work, we only give an upper bound on the sum of the ϕ_i variables. This is due to the desire to be robust to a certain number of response errors and inconsistencies that could occur. An alternative would be to add a term to the objective that penalizes "flipping." However, this would minimize the number of response errors we assume happen, instead of the number of response errors that actually happen. We would like to be robust to a certain number of response errors potentially occurring in practice.

These constraints provide a description of the feasible space of utilities. Throughout the remainder of this paper, we will also restrict the vector of utilities $\mathbf{u}$ to be in $[-1, 1]^n$, without loss of generality.

2.2.2 Selecting the Next Question

We would now like to select the next question to ask using this feasible space of utilities. We do this by finding the analytic center of the feasible set for $\mathbf{u}$, which is the projection of a mixed integer set. To find the analytic center of this set, we solve the optimization problem (2.1), where we use the notation $\mathbf{z}^i = \mathbf{x}^i - \mathbf{y}^i$ for the difference between the attribute vectors of the items asked about in question i:

$$\text{maximize} \sum_{\substack{i=1\\x_i\neq y_i}}^{k} \log(s_i)+\sum_{\substack{i=1\\x_i=y_i}}^{k}[\log(s_i^1)+\log(s_i^2)]+\sum_{j=1}^{2n}\log(t_j) \tag{2.1}$$

$$\begin{aligned}
\text{s.t. } & -\mathbf{u}'\mathbf{z}^i-(n+\epsilon)\phi_i+s_i^1=-\epsilon, && \forall i\in\{1,\dots,k\} \text{ s.t. } \mathbf{x}^i>\mathbf{y}^i, && (2.1.1)\\
& \mathbf{u}'\mathbf{z}^i+(n-\epsilon)\phi_i+s_i^2=n, && \forall i\in\{1,\dots,k\} \text{ s.t. } \mathbf{x}^i>\mathbf{y}^i, && (2.1.2)\\
& \mathbf{u}'\mathbf{z}^i-(n+\epsilon)\phi_i+s_i^1=-\epsilon, && \forall i\in\{1,\dots,k\} \text{ s.t. } \mathbf{y}^i>\mathbf{x}^i, && (2.1.3)\\
& -\mathbf{u}'\mathbf{z}^i+(n-\epsilon)\phi_i+s_i^2=n, && \forall i\in\{1,\dots,k\} \text{ s.t. } \mathbf{y}^i>\mathbf{x}^i, && (2.1.4)\\
& \mathbf{u}'\mathbf{z}^i+s_i^1=\epsilon, && \forall i\in\{1,\dots,k\} \text{ s.t. } \mathbf{x}^i=\mathbf{y}^i, && (2.1.5)\\
& -\mathbf{u}'\mathbf{z}^i+s_i^2=\epsilon, && \forall i\in\{1,\dots,k\} \text{ s.t. } \mathbf{x}^i=\mathbf{y}^i, && (2.1.6)\\
& \phi_i=0, && \forall i\in\{1,\dots,k\} \text{ s.t. } \mathbf{x}^i=\mathbf{y}^i, && (2.1.7)\\
& -M\phi_i\le s_i-s_i^1\le M\phi_i, && i=1,\dots,k \text{ s.t. } \mathbf{x}^i\neq\mathbf{y}^i, && (2.1.8)\\
& -M(1-\phi_i)\le s_i-s_i^2\le M(1-\phi_i), && i=1,\dots,k \text{ s.t. } \mathbf{x}^i\neq\mathbf{y}^i, && (2.1.9)\\
& -u_j+t_j=1, && j=1,\dots,n, &&\\
& u_j+t_{n+j}=1, && j=1,\dots,n, &&\\
& \sum_{i=1}^{k}\phi_i\le\gamma k, && &&\\
& \phi_i\in\{0,1\}, && i=1,\dots,k, &&\\
& s_i\ge 0, && i=1,\dots k, &&\\
& t_j\ge 0, && j=1,\dots 2n. &&
\end{aligned}$$

The objective of (2.1) is to maximize the log of the slack variables, as defined by the analytic center [20]. Constraints (2.1.1)-(2.1.7) represent the question constraints defined previously in this section, but with slack variables added in. Note that the slack variables are slightly more complicated than those in a typical optimization problem for the questions for which the user makes a choice (is not indifferent).

In our formulation, each question defines two constraints, but only one of them is nontrivial, depending on the value of ϕ_i. If $\phi_i = 0$, then the first question constraint, (2.1.1) or (2.1.3), defines the feasible utilities for that question, so we would only like to consider the slack variable for the first constraint (s_i^1), when maximizing the sum of the log of the slacks. If $\phi_i = 1$, then the second question constraint, (2.1.2) or (2.1.4), defines the feasible utilities for that question, so we would only like to consider the slack variable for the second constraint (s_i^2), when maximizing the sum of the log of the slacks. Constraints (2.1.8)-(2.1.9) use the "big-M" approach to set $s_i = s_i^1$ or $s_i = s_i^2$, depending on the value of ϕ_i, where s_i is the slack variable that contributes to the objective. The remaining constraints define the feasible region as described previously.

In practice, we solve this problem by first finding a strictly interior feasible point and then using Newton's method (or a suitable approximation of Newton's method) to iteratively compute the analytic center (for a more detailed explanation, see [77, 78]). Note, however, that unlike [77, 78], we compute the analytic center of a set that involves continuous and discrete variables.

Denote the optimal solution for $\mathbf{u}$, or the analytic center, by $\mathbf{c}^*$. To try and cut the feasible region as much as possible, we select as the next question the one whose hyperplane is as close to $\mathbf{c}^*$ as possible. This is the solution to the following problem:

$$i^* = \arg\min_i \frac{|(\mathbf{c}^*)'\mathbf{z}^i|}{||\mathbf{z}^i||},$$

where the minimization is over all possible questions $\mathbf{z}^i = \mathbf{x}^i - \mathbf{y}^i$ that can be asked between two items $\mathbf{x}^i$ and $\mathbf{y}^i$. Then question $k+1$ is defined by $\mathbf{z}^{i^*}$. Note that while this problem requires an enumeration of all possible question pairs ($O(n^2)$), it is fast for reasonably sized practical problems (hundreds of items and attributes). After asking question $k+1$ and receiving the response, we add two new question constraints and a variable ϕ_i to (2.1) and repeat the procedure described here to find question $k+2$.

At any point, we may want to stop asking questions, or the remaining feasible region might be so small that it is impractical or unproductive to continue asking questions. A common strategy is to compute the analytic center one last time, and use this as an estimate for the utilities. We propose a new approach that uses robust optimization over the entire feasible set, which we describe in Section 2.3.

In addition to the methodological change of finding the analytic center of a mixed continuous and discrete set, we made a few additional changes to the methodology proposed in [77, 78]. The first is that we select the next question that minimizes the distance to the analytic center of the remaining feasible space, instead of selecting a hyperplane that goes through the analytic center and is parallel to the shortest axis of the bounding ellipse as proposed by [77, 78]. The reason for this change is that we assume that we have a fixed set of items that we can ask about. We do not have the freedom to construct the best possible item to ask about, but instead have to pick from one of the fixed options. This means that we are not able to ask a question that necessarily goes through the analytic center of the polyhedron and is parallel to the shortest axis of a bounding ellipse. By selecting the question that has a hyperplane closest to the analytic center, we are using a variation of the idea proposed by [77, 78], with the same goal of reducing the feasible space by as much as possible with each question, regardless of the response given by the user.

We also add the option for a user to answer that they are "indifferent" between the two items, instead of forcing the user the pick between the two items. This adds an extra level of complexity to the problem, but also makes the questionnaire more user-friendly.

Additionally, instead of selecting a particular utility vector at the end of the algorithm, we keep the entire feasible space when we optimize over the utilities. The reason for this change is that we would like our approach to be more robust to error. We discuss this further in Section 2.3.

2.2.3 Adding a "Feel Strongly" Option

In the optimization problem (2.1), we assume that all questions are equally likely to contain response errors (unless the user selected the indifferent option) and thus any question constraints can be flipped. However, it is more likely that the user "feels strongly" about some responses, and is more ambivalent about others. We add an additional option for the user to indicate that they "feel strongly" about a response, in which case we do not include constraints (2.1.2) or (2.1.4) of (2.1). By doing this, if $\phi_i = 1$, the only constraint for question i becomes trivial, and thus we are not considering that question when computing the utilities. We don't want to force the inequality to flip, since the user felt strongly about their response, but we may need to relax the constraint due to inconsistencies in the responses.

Thus, when faced with two items, $\mathbf{x}^i$ or $\mathbf{y}^i$, the user can indicate that (a) he prefers $\mathbf{x}^i$ over $\mathbf{y}^i$, (b) he prefers $\mathbf{y}^i$ over $\mathbf{x}^i$, (c) he strongly prefers $\mathbf{x}^i$ over $\mathbf{y}^i$, (d) he strongly prefers $\mathbf{y}^i$ over $\mathbf{x}^i$, or (e) he is indifferent. We will denote a strong preference for item $\mathbf{x}^i$ over item $\mathbf{y}^i$ by $\mathbf{x}^i >> \mathbf{y}^i$.

With this additional option, we solve (2.1) with constraints (2.1.1)-(2.1.4) replaced by the following constraints:

$$\begin{aligned}
-\mathbf{u}'\mathbf{z}^i - (n+\epsilon)\phi_i + s_i^1 &= -\epsilon, && \forall i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{x}^i > \mathbf{y}^i \text{ or } \mathbf{x}^i >> \mathbf{y}^i, \quad (2.2)\\
\mathbf{u}'\mathbf{z}^i + (n-\epsilon)\phi_i + s_i^2 &= n, && \forall i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{x}^i > \mathbf{y}^i,\\
\mathbf{u}'\mathbf{z}^i - (n+\epsilon)\phi_i + s_i^1 &= -\epsilon, && \forall i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{y}^i > \mathbf{x}^i \text{ or } \mathbf{y}^i >> \mathbf{x}^i,\\
-\mathbf{u}'\mathbf{z}^i + (n-\epsilon)\phi_i + s_i^2 &= n, && \forall i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{y}^i > \mathbf{x}^i,\\
s_i^2 &= 0, && \forall i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{x}^i >> \mathbf{y}^i \text{ or } \mathbf{y}^i >> \mathbf{x}^i.
\end{aligned}$$

Note that we set $s_i^2 = 0$ if the user feels strongly about the response to question i, since we don't have a second constraint. By using the new constraints (2.2), we are able to correct for inconsistencies or errors in the responses, while decreasing the chance of violating the inequalities that we know the user feels strongly about, and are therefore most likely to be correct.

2.3 Loss Averse Solutions with Robust Optimization

Up to this point, we have been concerned with estimating utilities. However, our main concern is with selecting the appropriate item in order to maximize a user's utility subject to a set of constraints on the items. This is a common problem in many applications; a decision needs to be made among a set of items, and a natural goal is to maximize utility. This problem can be modeled by the following optimization problem:

$$\begin{aligned} \text{maximize } & \mathbf{u}'\mathbf{x} \\ \text{s.t. } & \mathbf{x} \in X, \end{aligned}$$

where X is our feasible set, $\mathbf{u}$ is our utility vector, and each possible choice is modeled by a vector of attributes denoted by $\mathbf{x} \in X$. Note that we have not made any assumptions regarding the feasible set X.

This optimization problem assumes that the utility vector $\mathbf{u}$ is fixed, but in many situations the utility vector is unknown. In the previous section, we proposed a strategy for learning $\mathbf{u}$ which, after a number k of questions, gives us a feasible region that is continuous in $\mathbf{u}$ and discrete in $\boldsymbol{\phi}$. Denote by $\mathcal{U}^k$ the feasible set of $\mathbf{u}$ for all possible values of $\boldsymbol{\phi}$ after k questions (note that we will not use the feel strongly option throughout this section, but the formulations can easily be extended to include it):

$$\begin{aligned} \mathcal{U}^k = \{\mathbf{u} \in \mathbb{R}^n \mid \; & \mathbf{u}'\mathbf{z}^i + (n+\epsilon)\phi_i \geq \epsilon, && \forall i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{x}^i > \mathbf{y}^i, \\ & \mathbf{u}'\mathbf{z}^i + (n-\epsilon)\phi_i \leq n, && \forall i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{x}^i > \mathbf{y}^i, \\ & \mathbf{u}'\mathbf{z}^i - (n+\epsilon)\phi_i \leq -\epsilon, && \forall i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{y}^i > \mathbf{x}^i, \\ & \mathbf{u}'\mathbf{z}^i - (n-\epsilon)\phi_i \geq -n, && \forall i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{y}^i > \mathbf{x}^i, \\ & -\epsilon \leq \mathbf{u}'\mathbf{z}^i \leq \epsilon, && \forall i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{x}^i = \mathbf{y}^i, \end{aligned}$$

$$
\begin{aligned}
&\phi_i = 0, && \forall i \in \{1,\dots,k\} \text{ s.t. } \mathbf{x}^i = \mathbf{y}^i, \\
&-1 \le u_i \le 1, && i = 1,\dots,n, \\
&\sum_{i=1}^{k} \phi_k \le \gamma k, && \\
&\phi_i \in \{0,1\}, && i = 1,\dots,k && \}.
\end{aligned}
$$

2.3.1 A Robust Optimization Approach

In previous work, the standard approach was to select a particular estimate for $\mathbf{u}$ out of all possible choices in $\mathcal{U}^k$, where the most typical choice has been to select the analytic center of $\mathcal{U}^k$. Here, we instead solve a robust optimization problem that considers the entire set $\mathcal{U}^k$:

$$\max_{\mathbf{x}\in X} \left[\min_{\mathbf{u}\in\mathcal{U}^k} \mathbf{u}'\mathbf{x} \right]. \tag{2.3}$$

For a fixed $\mathbf{x} \in X$, the inner optimization problem selects the worst possible utility vector that is in the set $\mathcal{U}^k$. Since the set $\mathcal{U}^k$ came from the adaptive questionnaire, it is possible that any vector in $\mathcal{U}^k$ is the user's true utility vector, and so we are looking at the worst case outcome. The outer optimization problem then tries to find the best item $\mathbf{x} \in X$ given this worst case approach. This problem is thus robust in the sense that we are trying to maximize the worst case scenario, and so we aim to be robust to error. This approach is typically referred to as Wald's maximin model [83], and is one of the most important models in robust optimization. Note that for a given $\mathbf{x} \in X$, the objective of (2.3) is a concave function, which captures the risk adverse utilities often exhibited by individuals.

Typically, this problem is solved by taking the dual of the inner problem, resulting in an optimization problem that is no more difficult than the original optimization problem [16]. However, in our case, the inner problem is a mixed-integer optimization problem, since the ϕ_i variables are binary. Since in many applications, the set X only specifies that one item should be selected, we will solve this problem by enumeration when we report empirical results in Section 2.5. Thus, for each possible $\bar{\mathbf{x}} \in X$, we solve

$$\min_{\mathbf{u} \in \mathcal{U}^k} \mathbf{u}'\bar{\mathbf{x}},$$

and then select the $\mathbf{x}$ for which the objective function value is the largest. This problem can be solved with mixed integer optimization techniques.

2.3.2 A CVaR Approach

While the robust optimization approach provides a loss averse method, it tends to produce solutions that may be too conservative. In this section, we present a method to model loss averse behavior with robust optimization using the concept of Conditional Value at Risk ($CVaR$), or expected shortfall [9]. $CVaR$ is a risk measure often used in finance to evaluate the market risk of a portfolio. The "$CVaR$ at the α% level" or $CVaR_\alpha$ is the average value of the worst α% of the cases. It is an alternative to value at risk (VaR_α), the α-quantile, but it is more sensitive to losses [9]. Mathematically, $CVaR$ can be defined as follows:

$$CVaR_\alpha = \frac{1}{\alpha} \int_0^\alpha VaR_\gamma d\gamma.$$

The motivation for this approach is similar to the trade-off proposed by [16], in that we would like to adjust the conservatism of the robust approach. The analytic center approach often yields better objective function values than the robust approach on average, but it also tends to have greater losses in the domain of losses than the robust version (it is not very loss-averse). By using $CVaR$, we are able to maintain the best features of both approaches. This will be shown in the empirical evidence in Section 2.5.

We would like to maximize the $CVaR$ of the possible utility vectors in the space of feasible utilities $\mathcal{U}^k$. Since this region includes discrete values for ϕ_i, we first fix these values to the values at the analytic center, which is found after the final question is asked using the procedure described in Section 2.2. We assume that these values

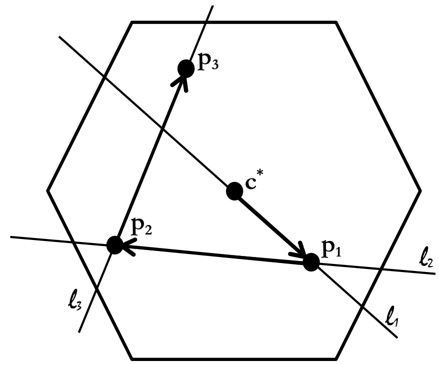


Figure 2-2: An example of the Hit and Run algorithm.

capture good values for ϕ_i, but in practice they could be varied between all possible values.

Since precisely modeling $CVaR$ would require an integral in the objective, we use random sampling to approximate the problem. Our feasible region $\mathcal{U}^k$ becomes a polytope when we fix the values of ϕ_i. There are several different ways to randomly sample from a polytope [81]. We use the "Hit-and-run" algorithm, starting from the analytic center c^* (which has been computed in the computation of $\mathcal{U}^k$) to find N different utility vectors $u_1, \ldots, u_N \in \mathcal{U}^k$. The Hit-and-run algorithm was introduced by Smith [71], and is defined as follows:

- Pick a uniformly distributed random line ℓ through the current point.
- Move to a uniform random point p along the chord $\ell \cap \mathcal{U}^k$.

These two steps are repeated until the random point can be assumed to come from the stationary distribution of the algorithm. Figure 2-2 gives an example of the first three steps of the algorithm in two dimensions. Starting from the analytic center c^*, a uniformly distributed line ℓ_1 through c^* is selected, and then we move to the random point p_1. Then from p_1, we select the uniformly distributed line ℓ_2 through p_1, and move to the random point p_2, and so forth.

Smith [71] proved that the stationary distribution of the hit-and-run walk is the uniform distribution over $\mathcal{U}^k$. Lovász and Vempala [55] showed that the hit-and-run walk mixes (becomes stationary) in $O(n^4 ln^3(\frac{n}{d}))$ steps starting from a point at distance d from the boundary, and is thus a polynomial time algorithm. Other random walks, including the well-known "ball walk", can potentially take exponentially many steps from some starting points. In addition to hit-and-run being a polynomial time algorithm, it is also known to perform very well in practice.

Given the random sample of utility vectors $\{u_1, \ldots, u_N\}$, we then maximize the $CVaR$ of this representative sample of utility vectors by using the robust optimization problem (2.4).

$$\begin{aligned} \max_{\mathbf{x} \in X} \ \min_{\mathbf{y}} \ & \frac{1}{\alpha N} \sum_{j=1}^{N} (\mathbf{u}_j' \mathbf{x}) y_j & & (2.4) \\ & \sum_{j=1}^{N} y_j = \alpha N, & & \\ & 0 \leq y_j \leq 1, & j = 1, \ldots, N, & \end{aligned}$$

where α controls how conservative, or loss averse, we would like to be. If we set $\alpha = \frac{1}{N}$ for $N \to \infty$, this is equivalent to the robust approach.

We can represent this problem as a linear optimization problem by taking the dual of the inner minimization problem [14]. We define dual variables θ and $\mathbf{w}$. Then, the problem can be reformulated as the linear optimization problem (2.5).

$$\begin{aligned} \max_{\mathbf{x}, \theta, \mathbf{w}} \ & \theta + \frac{1}{\alpha N} \sum_{j=1}^{N} w_j & & (2.5) \\ & \theta + w_j \leq \mathbf{u}_j' \mathbf{x}, & j = 1, \ldots, N, & \\ & w_j \leq 0, & j = 1, \ldots, N, & \\ & x \in X. & & \end{aligned}$$

Although our approach requires random sampling of the feasible region, it successfully provides loss averse solutions without being overly conservative. We will show the benefits of this formulation in Section 2.5.

2.4 An Online System to Learn Preferences

We have implemented an online software that uses the proposed approach to select preferred recipes for the user of a personalized dieting application. A snapshot of the online software is shown in Figure 2-3. Users are asked to answer comparison questions as described in Section 2.2 given the title of the recipes, pictures, descriptions of the recipes, and the ingredients. They are also given the option to indicate that they strongly prefer one recipe over the other, by selecting the "double thumbs-up" button (at any time, users can click on a "Help" button to read the instructions or a "Stop" button to terminate the survey). We would like to learn the user's utilities for the ingredients (or other attributes) of the recipes, and then ultimately suggest a meal plan that is appealing to them.

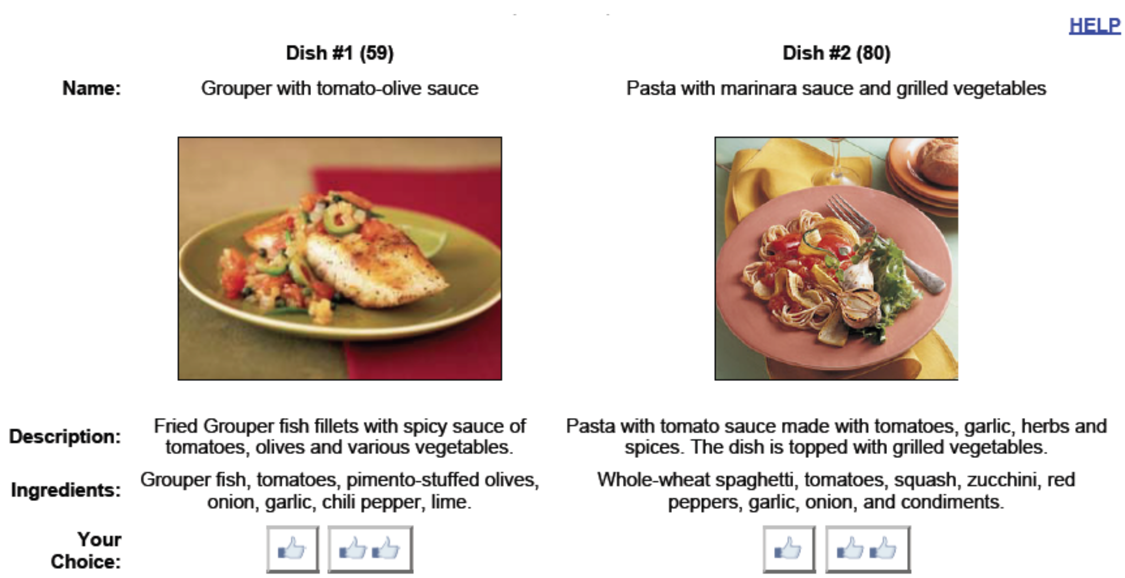
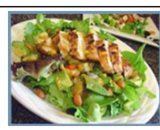


Figure 2-3: A snapshot of the adaptive questionnaire implemented in an online software.

To illustrate the system, we will give an example of the questions, responses, and learned preferences for a particular user. The system asks comparison questions for breakfast, lunch, and dinner separately, since the recipes served at each of these meals can vary significantly. Here, we will focus on the lunch survey. The example here uses 38 different recipes for lunch. For illustration purposes, assume that the user has a strong preference for vegetarian recipes, and prefers seafood over other meat dishes. The first ten questions that the user is asked and the user's responses are shown in Table 2.1. The first column gives the question number, the second column gives a picture and the name of the first recipe that is asked about in the question, and the third column gives a picture and the name of the second recipe that is asked about in the question. The user's response is indicated by either a single thumbs-up sign or a double thumbs-up sign in one of the two recipe columns.

Question	Recipe 1	Recipe 2
1	  Acorn Squash with Apples	 Black Bean Wrap
2	 Grilled Salmon on Sourdough Bread	   Zesty Tomato Soup
3	 Grilled Chicken Salad with Oranges	   Tuna Pita Pockets
4	 Pineapple Chicken Salad	   Rice and Beans Salad
5	 Chili	  Chicken and Coleslaw Wrap
6	 Quesadillas	  Spinach Frittata

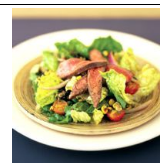
7	 Chicken Salad with Thai Flavors	  Salad Greens with Pears and Walnuts
8	  Pasta Salad with Mixed Vegetables	 Shrimp-Apple Salad
9	  Portobello and Blue Cheese Salad	 Grilled Flank Steak Salad
10	 Turkey Club Sandwich	  Crab Salad

Table 2.1: The first ten questions and responses of a sample user of the system.

Since the survey is adaptive, the questions are dependent on the answers to the previous questions. For example, if the user instead selected Recipe 2 for question 9 (Grilled Flank Steak Salad), question 10 would instead have been between "Caesar Salad with Grilled Chicken" and "Hot Ham and Cheese Sandwiches with Mushrooms."

After these ten questions, the system computes the user's preferences using our methodology. Table 2.2 gives a list of the recipes ranked the highest in terms of these preferences. After only ten questions, the system has captured the user's preferences very well. Recall that there are 38 lunch dishes total, of which 13 are vegetarian, 7 contain seafood, 12 contain chicken, and 6 contain red meat. Only 3 non-vegetarian dishes appear in the top ten list, one of which is a seafood dish. The two chicken dishes have a high preference due to the salad ingredients involved in them, but they are both near the bottom of the list. We have successfully learned that the user has a high utility for vegetarian dishes, including salads. In the next section, we present empirical evidence supporting our methodology using this online software.

Rank	Recipe
1	Salad Greens with Pears and Walnuts
2	Zesty Tomato Soup
3	Rice and Beans Salad
4	Acorn Squash with Apples
5	Spinach Frittata
6	Crab Salad
7	Grilled Pear and Watercress Salad
8	Chicken Caesar Pita
9	Portobello and Blue Cheese Salad
10	Chicken Coleslaw Wrap

Table 2.2: The top ten recipes according to the user's preferences computed after the first ten questions.

2.5 Empirical Evidence

In this section, we present evidence that a self-correcting and robust approach is feasible, realistic, and appealing in practice. We present empirical evidence comparing the analytic center approach, the robust approach, and the $CVaR$ approach.

2.5.1 The Experiment

To compare the validity of the different approaches, we performed the following experiment. We first generated a "true" utility vector $\mathbf{u}^*$ that a user could have, by randomly sampling a feasible utility vector from the initial feasible space of utilities $\mathcal{U}^0 = [-1, 1]^n$. Using this true utility vector, we generated the appropriate answers for a fixed number of comparison questions, where the questions were selected using the adaptive questionnaire described in Section 2.2. For each question, we inserted some normally distributed noise ζ with mean zero and standard deviation σ, into the response to that question. The reason for this is that we assume people make errors in their responses, either due to inaccurate responses, or ambiguous preferences. Thus, for $\zeta \sim N(0, \sigma)$, if

$$(\mathbf{u}^*)'\mathbf{z}^i + \zeta \geq \hat{\epsilon},$$

we record the user's response to question i as $\mathbf{x}^i > \mathbf{y}^i$. If

$$(\mathbf{u}^*)'\mathbf{z}^i + \zeta \leq -\hat{\epsilon}$$

we record the user's response to question i as $\mathbf{y}^i > \mathbf{x}^i$. Lastly, if

$$-\hat{\epsilon} \leq (\mathbf{u}^*)'\mathbf{z}^i + \zeta \leq \hat{\epsilon}$$

we record the user's response to question i as $\mathbf{x}^i = \mathbf{y}^i$.

In the empirical evidence presented in this section, we let $\hat{\epsilon} = 0.02$. This causes zero to four indifferent responses for every ten questions, depending on the true utility vector of the user. We assume here that while some indifferent responses might be made, the user will select between the two options most of the time. We have two reasons for this. The first is that the methodology proposed here is designed to ask comparison questions. If the user is answering indifferent for the majority of the questions, a comparison questionnaire is probably not appropriate for the specific application. The second reason is that our methodology is designed to handle response errors and inconsistencies, partly because we are forcing the user to pick between the two options.

Note that it is the self-correcting property of our approach that insures the feasibility of the algorithm regardless of how much noise is inserted into the responses. Even if inconsistencies are created, with γ large enough, the feasible space $\mathcal{U}^k$ will remain non-empty. However, recall that the adaptive questionnaire is designed so that inconsistencies will be avoided as much as possible, so we expect a small number of inconsistencies, if any. Since the next question is selected as one that cuts the current feasible space as close as possible to the analytic center, it will always leave a remaining feasible space unless the closest cut to the analytic center does not go through the current feasible space at all (leaving an empty feasible space with one response to the question, and the same feasible space as before with the other response to the question). While unlikely, if this does happen and the new feasible space turns out to be empty, the questionnaire will stop and the most recent non-empty feasible space will be the final feasible space. Throughout the empirical studies performed here, this situation has never occurred.

Using these responses, the ultimate optimization problem (that of finding an appealing meal plan for the user of a personalized dieting system) was then solved using the different methods discussed in this paper: the traditional analytic center approach with "slack" variables to account for inconsistencies, but without the self-correcting mechanism [78, 77], the new analytic center approach with the self correcting mechanism presented in (2.1), the robust approach given by (2.3), and the $CVaR$ approach given by (2.4). We will denote these methods by AC, ACSC, Robust, and CVaR, respectively. Note that there are two different analytic center approaches here - the traditional approach without accounting for inconsistencies and responses errors (AC), and the new approach with the self-correcting mechanism (ACSC).

We report the results of these methods as follows. We first rank all of the items (recipes) according to the "true" utilities, $\mathbf{u}^*$. Thus, the item with the highest true utility gets rank 1, and the item with the lowest utility gets rank 102 (the total number of recipes in the dataset). We then find the top five items according to the different optimization methods. The reason for this is that in this application and in many others, we will often suggest several items to the user that we think they will like. In the case of recipes, we will often want to make several suggestions since the user may like many different types of recipes, and we would like to make sure we suggest one that they are interested in on any particular day. Therefore, after solving the optimization problem corresponding to each of the methods, we eliminate the optimal solution, and solve again to get the next best solution. We repeat this three more times to get the top five solutions. We can then compare the selected solutions with the true rank of the items. We report the average true rank of the top five solutions found. Therefore, smaller values are preferred.

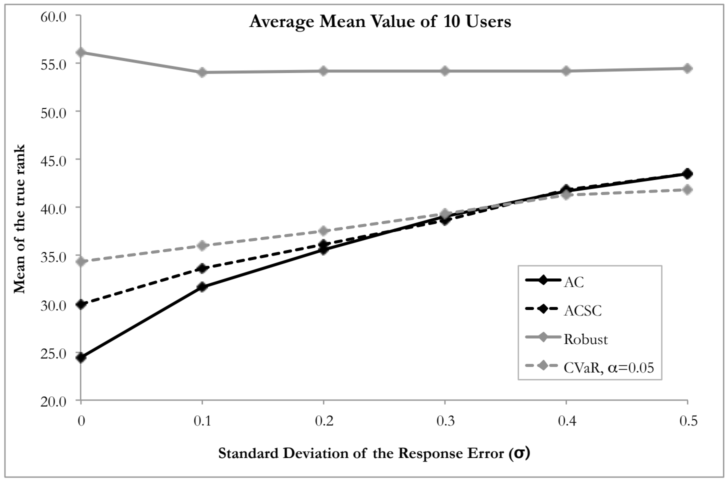
Throughout the rest of this section, we report results for ten different true utility vectors (or ten different "users"), and 200 runs per user for each value of $\sigma > 0$ to account for the noise in the responses. Our dataset consisted of 102 recipes for the main dish at dinner (so that the recipes were comparable), and were described by 186 attributes (the ingredients of the recipes).

2.5.2 Results

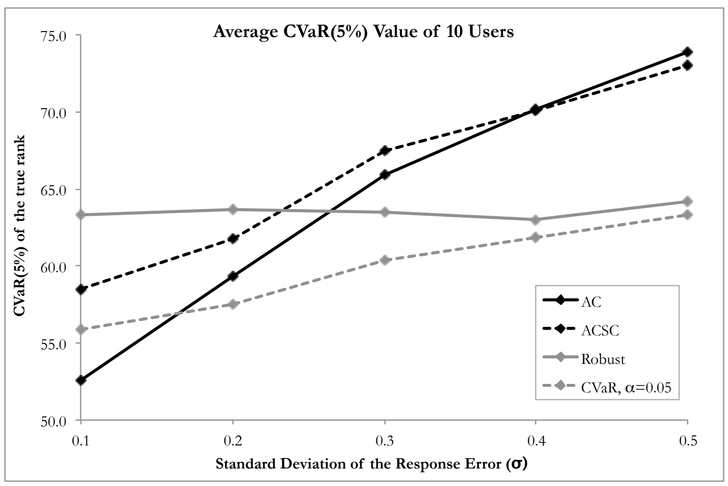
Figures 2-4(a), 2-4(b), and 2-4(c) show the results for 10 questions, and Figures 2-5(a), 2-5(b), and 2-5(c) show the results for 20 questions. For both sets of results, $\gamma = 0.1$, meaning that at most 10% of the responses are assumed to be inaccurate or inconsistent. Tables 2.3, 2.4 and 2.5 show the full numerical results. The bold entries in the tables highlight some of the best results and some of the interesting trends in the data.

In Figures 2-4 and 2-5, each line corresponds to a different optimization approach: the traditional analytic center approach (AC); the self-correcting analytic center approach (ACSC); the robust approach (Robust); and the CVaR approach (CVaR). The x-axis gives the value of σ, the standard deviation of the noise added to the responses. For $\sigma = 0$, there is no random noise added to the responses, so only one value is computed per user and CVaR(5%) and CVaR(20%) are not computed due to the small amount of data. For $\sigma > 0$, 200 runs are computed for each method, combination of parameters, and for each user.

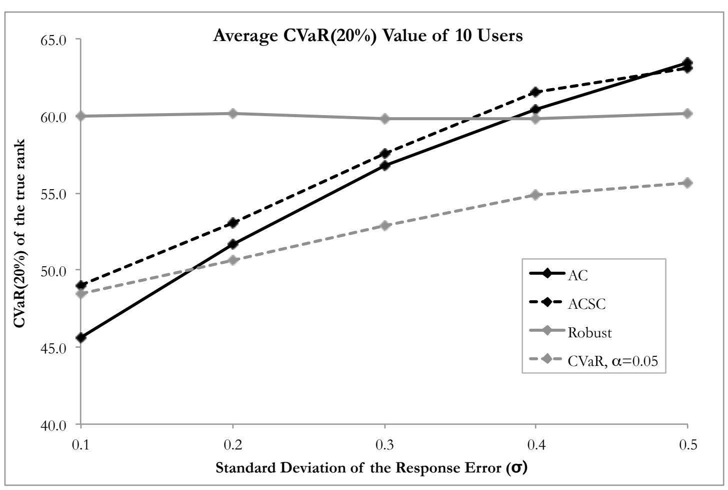
Figures 2-4(a) and 2-5(a) give the average rank of the top five solutions for AC, ACSC, CVaR (with $\alpha = 0.05$), and Robust. Recall that there are 102 total recipes, so the average rank can range from 3 to 100, where smaller values are preferred. As can be seen from Figures 2-4(a) and 2-5(a), the AC, ACSC and CVaR methods are competitive on average, with the AC method better than the other two for small σ. The ACSC and CVaR methods are also most robust to noise than the AC method, meaning that they are flatter than the AC curve. Since these methods better account



(a) Average rank.

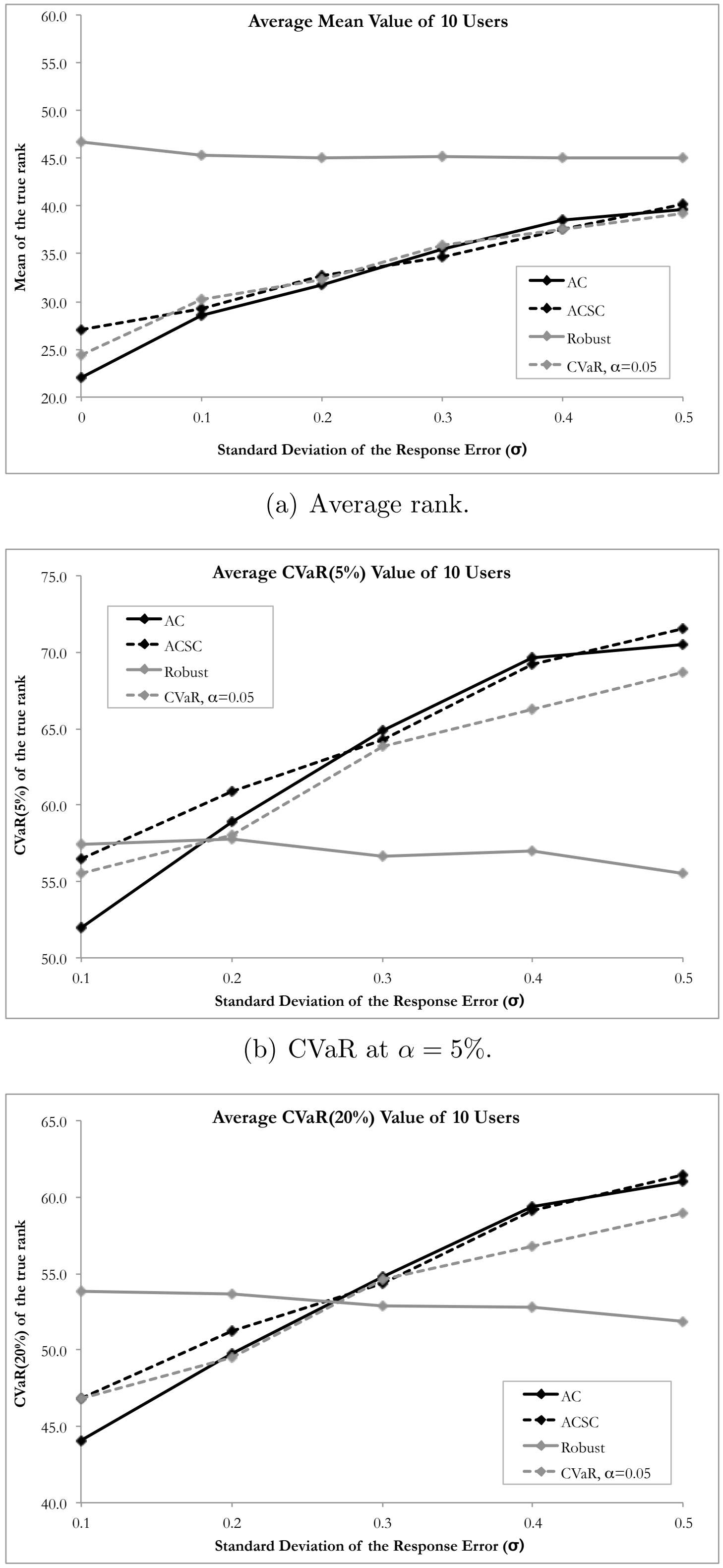


(b) CVaR at $\alpha = 5\%$.



(c) CVaR at $\alpha = 20\%$.

Figure 2-4: Results for 10 questions.



(a) Average rank.

(b) CVaR at $\alpha = 5\%$.

(c) CVaR at $\alpha = 20\%$.

Figure 2-5: Results for 20 questions.

for response error and inconsistencies, they logically perform better with $\sigma > 0$.

The Robust method is strictly dominated by the other methods, but it is very consistent regardless of the amount of noise. Additionally, Table 2.3 gives the standard deviation of the methods. The CVaR method has a consistently lower standard deviation than the traditional AC method, and the Robust method has a significantly lower standard deviation than all other methods. This provides evidence that the new methods are more risk averse than the traditional analytic center approach.

The largest different between asking ten questions (Figure 2-4(a)) and asking twenty questions (Figure 2-5(a)) is for the Robust method, which performs significantly better with twenty questions. Since the Robust is a worst case approach, it makes sense that it would perform much better with a smaller feasible space.

		Method			
Questions	σ	AC	ACSC	CVaR, $\alpha = 0.05$	Robust
10	0.0	**24.4**	30.0	34.4	56.1
10	0.1	$\mathbf{31.8 \pm 9.2}$	33.7 ± 10.5	36.0 ± 8.6	54.0 ± 5.5
10	0.2	$\mathbf{35.7 \pm 10.7}$	$\mathbf{36.2 \pm 11.3}$	37.5 ± 9.0	54.1 ± 5.7
10	0.3	$\mathbf{39.1 \pm 12.0}$	$\mathbf{38.6 \pm 12.6}$	$\mathbf{39.4 \pm 9.4}$	54.1 ± 5.8
10	0.4	$\mathbf{41.7 \pm 12.9}$	$\mathbf{41.9 \pm 13.4}$	$\mathbf{41.3 \pm 9.4}$	54.2 ± 5.7
10	0.5	43.5 ± 13.6	43.5 ± 13.7	$\mathbf{41.9 \pm 9.7}$	54.4 ± 5.8
20	0.0	**22.0**	27.1	24.4	46.7
20	0.1	$\mathbf{28.5 \pm 10.7}$	29.3 ± 11.8	30.2 ± 11.2	45.3 ± 6.3
20	0.2	$\mathbf{31.7 \pm 12.3}$	32.8 ± 12.6	32.3 ± 11.7	45.1 ± 6.4
20	0.3	35.4 ± 13.2	$\mathbf{34.6 \pm 13.3}$	35.9 ± 12.9	45.2 ± 5.9
20	0.4	38.5 ± 14.1	37.6 ± 14.5	$\mathbf{37.6 \pm 13.1}$	45.0 ± 6.0
20	0.5	39.7 ± 14.6	40.2 ± 14.6	$\mathbf{39.2 \pm 13.6}$	45.1 ± 5.3

Table 2.3: Average true rank of the solution to the optimization problem and the standard deviation of the values for different values of σ. For all methods, $\gamma = 0.1$. Results are averaged with respect to 10 different users, and 200 runs of the adaptive questionnaire per user when $\sigma > 0$.

Figures 2-4(b) and 2-5(b) give the conditional value at risk ($CVaR_\alpha$) for $\alpha = 5\%$ for 10 questions and 20 questions, respectively. Figures 2-4(c) and 2-5(c) give the conditional value at risk ($CVaR_\alpha$) for $\alpha = 20\%$ and for 10 questions and 20 questions, respectively. Tables 2.4 and 2.5 give the full numerical results. We use $CVaR_\alpha$ as a risk measure to compare how loss averse each of the methods are. The lower the $CVaR_\alpha$ value (meaning that the worst case solutions have better ranks), the more loss averse a method is considered to be.

Figures 2-4(b) and 2-5(b) show that for almost all levels of response error, the CVaR or Robust methods strictly dominate the other methods for the worst 5% of cases. The CVaR method is better with fewer questions, and the Robust method is better with more questions. The AC method is slightly better for small response error. Figures 2-4(c) and 2-5(c) show similar behavior. This shows us that the CVaR and Robust methods are significantly more loss averse, as designed.

		Method			
Questions	σ	AC	ACSC	CVaR, $\alpha = 0.05$	Robust
10	0.1	**52.6**	58.5	55.9	63.3
10	0.2	59.3	61.7	**57.6**	63.7
10	0.3	65.9	67.5	**60.4**	63.5
10	0.4	70.1	70.1	**61.9**	63.0
10	0.5	73.9	73.1	**63.3**	**64.1**
20	0.1	**51.9**	56.4	55.5	57.4
20	0.2	58.9	60.8	58.0	**57.8**
20	0.3	64.9	64.3	63.8	**56.7**
20	0.4	69.7	69.2	66.3	**57.0**
20	0.5	70.5	71.5	68.7	**55.5**

Table 2.4: $CVaR_\alpha$ at $\alpha = 5\%$ of the true rank of the solution to the optimization problem for different values of σ. For all methods, $\gamma = 0.1$. Results are computed with respect to 10 different true utility vectors, and 200 different runs of the adaptive questionnaire per user when $\sigma > 0$.

When considering all plots together, we would argue that for small response error ($0 < \sigma \leq 0.2$), the AC approach is the best on average, as expected. But the CVaR and Robust approaches start looking better as the response error increases in this interval. For moderate to high response error ($0.2 < \sigma \leq 0.5$), the CVaR or Robust method would be the best choice, depending on the number of questions. If only a few questions are asked (around 10), then the CVaR approach is probably better. If there is time for more questions, then Robust approach is better. While asking fewer questions is more appealing for the user, the Robust approach uses less computational

		Method			
Questions	σ	AC	ACSC	CVaR, $\alpha = 0.05$	Robust
10	0.1	**45.6**	49.0	48.4	60.0
10	0.2	**51.7**	53.1	**50.6**	60.1
10	0.3	56.8	57.5	**52.9**	59.8
10	0.4	60.4	61.5	**54.9**	59.8
10	0.5	63.5	63.1	**55.7**	60.2
20	0.1	**44.1**	46.8	46.8	53.9
20	0.2	49.7	51.2	**49.5**	53.7
20	0.3	54.8	54.4	54.6	**52.9**
20	0.4	59.4	59.1	56.8	**52.8**
20	0.5	61.1	61.4	58.9	**51.9**

Table 2.5: $CVaR_\alpha$ at $\alpha = 20\%$ of the true rank of the solution to the optimization problem for different values of σ. For all methods, $\gamma = 0.1$. Results are computed with respect to 10 different true utility vectors, and 200 different runs of the adaptive questionnaire per user when $\sigma > 0$.

power since random sampling is not required. These two approaches have the best performance when considering both the average value and CVaR. Overall, given that we expect some response error from the users, we would argue that the CVaR and Robust approaches show the best numerical results, in terms of losses and average performance.

2.6 Summary

We have developed an optimization-based approach for preference learning that incorporates techniques and observations from many different fields. Our approach addresses some of the key observations of preference learning in behavioral economics: people are loss averse, are inconsistent, and evaluate outcomes with respect to deviations from a reference point. We have shown how mixed binary optimization can be used to correct for inconsistent behavior, choice-based conjoint analysis can be used in an adaptive questionnaire to dynamically select pairwise questions, and robust linear optimization and $CVaR$ can model loss averse behavior.

Furthermore, we gave empirical evidence from an online software we developed that strives to model human preferences in a realistic situation. We have shown that the $CVaR$ and Robust approaches perform very well, and are more robust to noise in the responses and are more loss averse than the traditional analytic center approach. In Chapter 4, we will use the approach introduced here to find a meal and exercise plan than maximizes the user's food preferences.

Chapter 3

Blood Glucose Modeling and Nutritional Data

Our approach to personalized diabetes management requires several inputs. The first necessary input, user preferences, was the focus of Chapter 2. In this chapter, we discuss the other inputs necessary for our approach.

In Section 3.1, we introduce some basic terminology and concepts about diabetes. In Section 3.2, we describe a data-driven approach to estimate how a person's blood glucose levels will change after eating and exercising. In Section 3.3, we use optimization to update our knowledge of the individual's blood glucose behavior using data from the user. In Section 3.4, we describe the nutritional information we use, including a comprehensive recipe database and nutritional guidelines.

3.1 Diabetes Basics

Diabetes is a chronic disease characterized by excessive glucose in the bloodstream. The body breaks down sugars and starches into glucose, a form of sugar. This glucose then enters the bloodstream, causing blood glucose levels to rise. In a non-diabetic individual, the hormone insulin is produced to remove this glucose from the bloodstream, and convert it into energy to be used by the cells. In a diabetic individual, no insulin or not enough insulin is produced, so the blood glucose levels remain high. High blood glucose levels can lead to very serious complications, including glaucoma, cataracts, amputation, kidney disease, nerve damage, and heart disease [2].

There are several different types of diabetes. Type I diabetes is characterized by no insulin production. Insulin must be injected to absorb glucose from the bloodstream. Type I diabetes accounts for about 5% of all cases of diabetes, and is usually diagnosed in childhood. Type I diabetes is not associated with obesity, and while diet and exercise can definitely help, medication is required.

Type II diabetes is characterized by either not enough insulin production, or insulin that does not work the way it should. Type II diabetes accounts for about $90\% - 95\%$ of the diabetes cases worldwide, and is the type accountable for the alarming increase in diabetes rates. While type II diabetes tends to occur in adults, a growing number of children and adolescents are developing type II diabetes, most likely due to the increase in childhood obesity. Type II diabetes can be managed with diet and exercise alone in many individuals. For some, medication is needed, especially as the disease progresses or goes untreated. If a person with type II diabetes can successfully lower their blood glucose levels through diet and exercise, they not only lower their risk for diabetes complications, but they can avoid the side-effects of the various diabetic medications. For these reasons, the focus of our effort is on developing a personalized approach to use diet and exercise for diabetes management, specifically for type II diabetes.

The remaining cases of diabetes are either gestational diabetes, or rare manifestations of diabetes. Gestational diabetes is a form of type II diabetes and occurs during pregnancy. The risk factors for gestational diabetes are similar to those for type II diabetes, including being overweight before or during pregnancy. About 2% to 10% of pregnant women are diagnosed with gestational diabetes. A healthy meal and exercise plan can help treat gestational diabetes, and blood glucose levels typically return to normal after the baby is born. However, someone who has gestational diabetes is at higher risk for developing type II diabetes; women who have had gestational diabetes have a 35% to 60% chance of developing type II diabetes in the next ten to twenty years. In most cases, diabetes can be prevented in these individuals by maintaining a healthy weight, eating a healthy diet, and exercising regularly. This makes our system also very suitable for women who have or have had gestational diabetes.

Prediabetes is a condition that is also very relevant for our approach. Prediabetes refers to people who have higher than normal blood glucose levels, but not high enough to be diagnosed with diabetes. People with prediabetes are at very high risk for developing type II diabetes – $15\% - 30\%$ of people with prediabetes will develop type II diabetes within five years. Over a third of U.S. adults and over half of adults older than 65 have prediabetes. Lifestyle changes (diet and exercise) are particularly useful for treating prediabetes, and are the recommended treatment method. Our approach is also very applicable for people with prediabetes.

Normal, prediabetic, and diabetic ranges for blood glucose levels are described in Table 3.1. Fasting blood glucose is the blood glucose level of an individual when waking up in the morning or before meals, and is measured in miligrams per deciliter (mg/dL). Postprandial blood glucose is the blood glucose level of an individual up to two hours after eating, and is also measured in mg/dL. A hemoglobin A1C (HbA1C) test shows an average of blood glucose levels over the past three months, and is used as a measure of overall control. The American Diabetes Association recommends that someone with diabetes tries to keep their fasting blood glucose level below 130 mg/dL, postprandial levels below 180 mg/dL, and A1C below 7%.

Measurement	**Normal**	**Prediabetes**	**Diabetes**
Fasting Blood Glucose	$<$ 100 mg/dL	100 - 125 mg/dL	$>$ 125 mg/dL
Postprandial Blood Glucose	$<$ 140 mg/dL	140 - 200 mg/dL	$>$ 200 mg/dL
A1C	$<$ 5.7%	$5.7\% - 6.4\%$	$>$ 6.4%

Table 3.1: Ranges for blood glucose levels.

Blood glucose levels can be measured at home by using a blood glucose meter. This requires a finger prick to collect a small amount of blood, and then a hand-held device reads the blood sample and shows the blood glucose result. It is recommended that individuals monitor their blood glucose levels to see how food, exercise, or medications affect them.

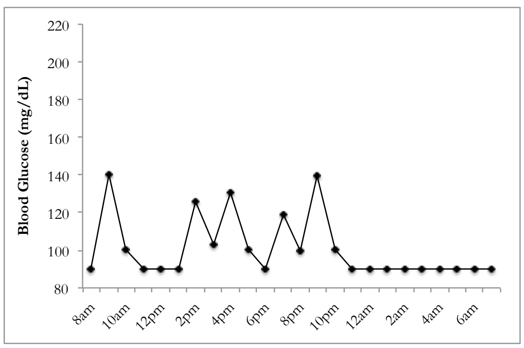
3.2 Blood Glucose Modeling

Due to the importance of blood glucose monitoring in diabetic individuals, a critical component of our approach is the modeling and data for blood glucose dynamics. It has been observed that an individual's blood glucose levels follow a trajectory like those shown in Figure 3-1 [36, 40, 65, 67, 75, 25, 64]. We will refer to this curve as a person's blood glucose (BG) curve.

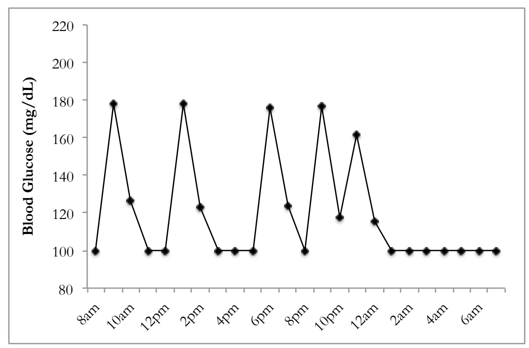
Depending on an individual's metabolism and diabetic state, the base level, height of the peaks, and duration of the peaks could be different. Figure 3-1(a) is representative of what a non-diabetic individual's BG curve might look like. It has a low fasting level (the blood glucose level of an individual when they wake up in the morning, or the base level of the curve), and low peaks that do not last very long. Figure 3-1(b) is representative of what a pre-diabetic individual's BG curve might look like. It has a slightly higher fasting level than that of Figure 3-1(a), and the peaks are higher, but still less than 180 mg/dL. Figures 3-1(c) and 3-1(d) are representative of what a diabetic individual's BG curve might look like. They both have a fasting level in the diabetic range (greater than 125 mg/dL), and the peaks are higher and last longer. Figure 3-1(d) shows the impact of two snacks in addition to the three meals. Because the consumption of food can be spread out more with snacks, this curve has lower peaks than those of Figure 3-1(c). We will exploit this fact by including snacks when we plan meals, if the user of the system is willing to eat snacks. Additionally, some of the peaks may have been lowered by exercise.

In addition to general trends seen in diabetic versus non-diabetic individuals, each individual's blood glucose reactions can be different. After eating the same food, two different individuals can have drastically different blood glucose responses, even if they are both in the same diabetic state. This can depend on body composition, activity levels, or genetics.

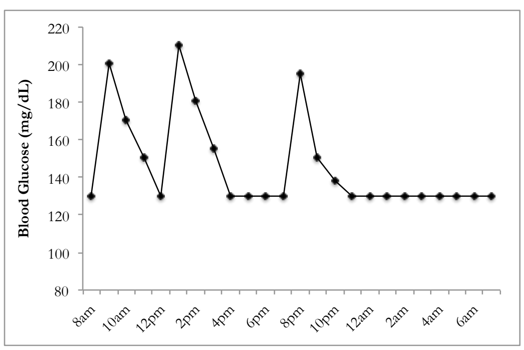
By analyzing the BG curve, we can model the blood glucose levels of an individual as a function of the food consumed and the exercise performed at certain times throughout the day. This is a key observation for our approach. Diabetic drugs and insulin can also affect an individual's blood glucose curve, but in this work, we will assume that the individual is not taking medication. By modeling the effect of medications on the fasting and postprandial peaks, the work here can be extended to include medications, which is an area of future work.



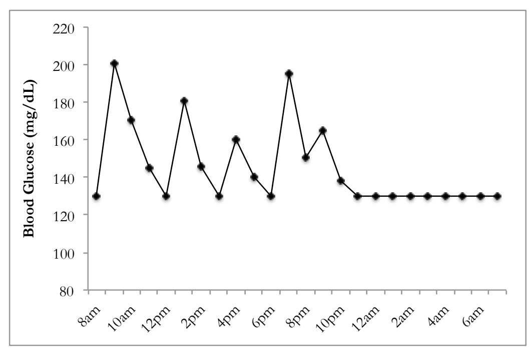
(a) An example BG curve of a non-diabetic.



(b) An example BG curve of a pre-diabetic.



(c) An example BG curve of a diabetic, with no snacks.



(d) An example BG curve of a diabetic, with two snacks.

Figure 3-1: Potential trajectories of the blood glucose levels of individuals.

3.2.1 The Glycemic Index

We construct a function to model blood glucose dynamics using the glycemic index data and our knowledge of the body's reaction to foods with various glycemic index values. The glycemic index (GI) measures the effects of carbohydrates on blood glucose levels. Since sugars and starches are the foods that raise blood glucose levels, carbohydrates are responsible for most of the changes in blood glucose levels. This concept was developed to find out which foods were best for people with diabetes [47]. The glycemic index of a food is defined as:

$$\text{GI} = \frac{\text{Area under the 2 hour BG curve after eating 50g of carbs of food}}{\text{Area under the 2 hour BG curve after eating 50g of carbs of pure glucose}} \times 100,$$

and can range from 0 to 100. The blood glucose (BG) curve is like the ones shown in Figure 3-1, and to compute the GI of a food, researchers measured this curve for 10 different test subjects and then used the average to compute the GI value.

As an example, consider Figure 3-2. This shows the BG curves for one individual and one test food. Suppose at 8am on day 1, the person ate 50 grams of carbohydrates of pure glucose, and his blood glucose followed the trajectory given by the gray area. Then at 8am on day 2, he ate 50 grams of carbohydrates of a test food we are interested in measuring the glycemic index of, and his blood glucose followed the trajectory given by the black area. We can then compute that:

$$\text{GI} = \frac{\text{Area(Test Food)}}{\text{Area(Pure Glucose)}} \times 100 = \tfrac{60}{100} \times 100 = 60.$$

Thus for this individual, we compute the glycemic index of the test food to be 60. We can then repeat this for nine other individuals, and the average of all ten GI values would be reported as the glycemic index of the test food.

We use glycemic index values from a published glycemic index database [10]. All foods containing carbohydrates are given a glycemic index value, and then can be classified as high glycemic index, medium glycemic index, or low glycemic index. Foods without carbohydrates do not have a glycemic index since they do not raise blood glucose levels. Foods with carbohydrates that break down quickly and thus release glucose into the bloodstream quicker tend to have a high glycemic index, such as white bread or potatoes. Foods that break down more slowly, releasing glucose into the bloodstream gradually tend to have a low glycemic index, such as strawberries or yogurt. Table 3.2 gives a summary of the glycemic index categories and some example foods in each group.

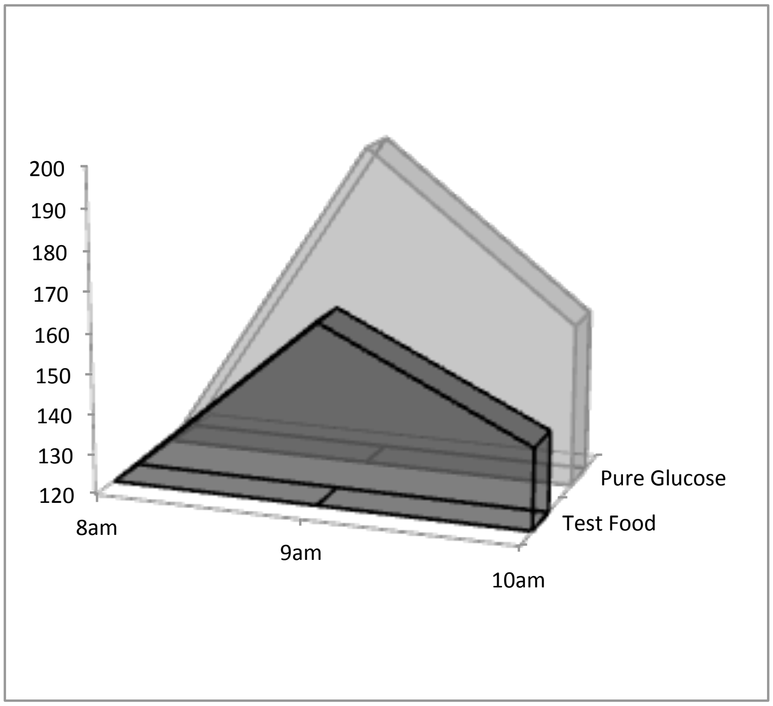


Figure 3-2: An example of the glycemic index computation.

Group	GI range	Example Foods
No GI	0	meat, fish, eggs, cheese, oil
Low GI	1 - 55	milk, yogurt, apples, berries, broccoli, beans, nuts
Medium GI	56 - 69	mango, raisins, whole-wheat bread, brown rice
High GI	70 - 100	potatoes, watermelon, white bread, rice cereal

Table 3.2: Categories of the Glycemic Index.

We use the glycemic index to aggregate foods into groups, and estimate the effect each group has on an individual's blood glucose levels. We know that consuming high glycemic index foods increases an individual's blood glucose levels the most, and that consuming foods without a glycemic index should not increase an individual's blood glucose levels at all.

3.2.2 The Models

We have developed the following equation to predict how the blood glucose levels of an individual will change after eating food and exercising:

$$BG_t = f + \sum_{i \in \mathcal{F}} \left[\alpha^i c^i x^i_{t-1} + \beta^i c^i x^i_{t-2}\right] - \alpha^e z_{t-1} - \beta^e z_{t-2}$$

where BG_t is the individual's blood glucose level at time t, f is the fasting level of the individual (the blood glucose level in the morning before consuming any food), $\mathcal{F}$ is the set of all foods that the user can consume, c^i is the amount of carbohydrates (in grams) in one serving of food i, x^i_t is the number of servings of food i consumed at time t, and z_t is the amount of exercise (in minutes) at time t.

Each of the constant multipliers α^i, β^i, α^e, and β^e model how each action changes the blood glucose levels: α^i is the blood glucose increase one hour after eating one gram of carbohydrates of food i, β_i is the blood glucose increase two hours after eating one gram of carbohydrates of food i, α^e is the blood glucose decrease one hour after exercising one minute, and β^e is the blood glucose decrease two hours after exercising one minute. Based on empirical evidence and the glycemic index, we estimate these values which then serve as starting points for a new user, but we believe that they are unique for each individual [36, 40, 65, 64, 67, 75]. We discuss how we can update these values with individual user data in Section 3.3.

This equation can be extended to include a longer time horizon (i.e. 3 hours) or shorter time intervals (i.e. every half hour). However, we have observed empirically that by three hours, the blood glucose levels of most individuals are back to fasting levels. Additionally, it is difficult to collect measurements more frequently than in one hour measurements, since each measurement requires a new finger prick. However, our models can easily be extended to include this additional information if available.

As an example, suppose the user ate breakfast at time t and consumed two foods: an apple (17 grams of carbohydrates) and a piece of whole-wheat toast (20 grams of carbohydrates). The apple has a low glycemic index, for which we estimate $\alpha^i = 1.33$ and $\beta^i = 0.33$. The toast has a medium glycemic index, for which we estimate

$\alpha^i = 2.0$ and $\beta^i = 0.67$. Suppose the user's fasting level is 120 mg/dL. Then we can calculate the user's blood glucose level at time $(t+1)$ to be:

$$BG_{t+1} = 120 + (1.33 * 17) + (2.0 * 20) = 182.6,$$

and their blood glucose level at time $(t+2)$ to be:

$$BG_{t+2} = 120 + (0.33 * 17) + (0.67 * 20) = 139.0.$$

Note that it is important to know both the glycemic index and the carbohydrate content of all foods consumed. Eating a large amount of a low glycemic index food can have the same effect on blood glucose levels as eating a small amount of a high glycemic index food. This concept is frequently referred to as *glycemic load*.

3.3 Personalization of the Blood Glucose Parameters

For each user, we store a set of blood glucose parameters: f, α^i, β^i, α^e, and β^e. These were defined in Section 3.2. While we can initialize these parameters to default values, we would ideally like them to match the user's personal blood glucose measurements. In this section, we describe a method for adjusting the parameters to reflect new measurements provided by the user.

As mentioned in Section 3.2, we will assume that each food falls into one of four categories, according to the glycemic index: high glycemic index, medium glycemic index, low glycemic index, or no glycemic index. We can extend this approach to assuming that the parameters for each food are different, but we make this assumption for simplicity and feasibility in practice. With this assumption, we have reduced the set of blood glucose parameters to the following nine parameters: f, α^H, α^M, α^L, β^H, β^M, β^L, α^e, and β^e, where H denotes high glycemic index, M denotes medium glycemic index, and L denotes low glycemic index. Note that we do not need parameters for the no glycemic index foods, since we assume they do not raise blood glucose levels.

We assume that each day, we will potentially receive new measurements from the user. These measurements are either of the fasting level, or of postprandial blood glucose levels (blood glucose levels one or two hours after a meal). In practice, blood glucose measurements are often up to 20% inaccurate for many reasons: the blood glucose meter used, the environment of the user, the adherence of the user to the protocol of the measurement device, and the adherence of the user to the eating and exercise plan [41]. Due to these sources of inaccuracy, we have to assume that any measurements we receive are noisy, and we should not completely replace our current knowledge with any new data. We will first discuss how we update the user's fasting level, and then discuss how we update the hourly blood glucose parameters.

3.3.1 Updating the Fasting Level

Most individuals with diabetes regularly check their fasting level when waking up in the morning. This is seen as a measurement of overall diabetes control, and is also frequently used to test for diabetes or prediabetes. Due to the potential for measurement error and noise in the user's fasting level, we don't want to completely replace our current fasting level f with any single new measurement. To account for this, we use the following procedure:

1. Retrieve any measurements from the user in the last 30 days. Call this set of measurements $\mathcal{S}$.
2. If $|\mathcal{S}| > 10$, replace $\mathcal{S}$ with only the 10 most recent measurements.
3. If $|\mathcal{S}| \geq 6$, let $f = \text{median}(\mathcal{S})$.
4. Else, remove values that are less than $0.8f$ or greater than $1.2f$, and let $f = \text{average}(\mathcal{S})$.

Steps 1 and 2 collect the measurements that the user has entered, and are meant to collect enough measurements, but not so many that we lose changes in the individual's fasting level over time. Step 3 computes the new fasting level to be the median, if there are enough measurements. The reasoning for this is that the median is more robust to outliers, but is only appropriate if there are enough measurements. Step 4 computes the new fasting level to be the mean after deleting outliers, if there are fewer measurements.

The reasoning for this procedure is that we want to capture increases or decreases over time, but we do not want to be too sensitive to any individual measurement. We will assume here that we perform this update each time we receive a new fasting measurement.

As an example, suppose that we currently believe a user's fasting level to be 120 mg/dL. Furthermore, suppose that we have received the following set of measurements from the user in the last 30 days, listed in the order received:

$$\mathcal{S} = \{121, 118, 125, 132, 120, 122, 119, 127, 118, 117, 123, 120\}.$$

Since this set contains 12 measurements, we remove the two earliest measurements. Our new set of measurements is:

$$\mathcal{S} = \{125, 132, 120, 122, 119, 127, 118, 117, 123, 120\}.$$

Since we have at least six measurements, we compute the new fasting level to be the median of this set, which is 121.

3.3.2 Updating the Food and Exercise Parameters

Our equation for the blood glucose level at time t is given by

$$BG_t = f + \sum_{i \in \mathcal{F}} (\alpha^i c^i x^i_{t-1} + \beta^i c^i x^i_{t-2}) - \alpha^e z_{t-1} - \beta^e z_{t-2}.$$

We can simplify this equation since we aggregated all foods into high, medium or low glycemic index (the no glycemic index foods are not considered since their terms are zero in this equation). Denote the total number of carbohydrates consumed of high, medium, and low glycemic index foods at time t by C_t^H, C_t^M, and C_t^L respectively. Then we can simplify this equation to:

$$BG_t = f + \alpha^H C_{t-1}^H + \alpha^M C_{t-1}^M + \alpha^L C_{t-1}^L + \beta^H C_{t-2}^H + \beta^M C_{t-2}^M + \beta^L C_{t-2}^L - \alpha^e z_{t-1} - \beta^e z_{t-2}.$$

Suppose we have received k postprandial blood glucose measurements, given by the set $\mathcal{M} = \{M_1, M_2, \ldots, M_k\}$. Additionally, denote the carbohydrates of each type (high, medium or low glycemic index) consumed one hour before measurement M_j by $C_j^{H,1}$, $C_j^{M,1}$ and $C_j^{L,1}$. Similarly, denote the carbohydrates of each type consumed two hours before measurement M_j by $C_j^{H,2}$, $C_j^{M,2}$ and $C_j^{L,2}$. Lastly, denote the fasting level at the time of each measurement M_j by f_j. We then solve the optimization problem (3.1) to update the blood glucose parameters.

$$\begin{aligned}
\min \quad & \Theta \sum_{j=1}^{k} |M_j - \widehat{BG}_j| + (1-\Theta)\delta && (3.1) \\
\text{s.t.} \quad & \widehat{BG}_j = f_j + \hat{\alpha}^H C_j^{H,1} + \hat{\alpha}^M C_j^{M,1} + \hat{\alpha}^L C_j^{L,1} + \hat{\beta}^H C_j^{H,2} \\
& \qquad + \hat{\beta}^M C_j^{M,2} + \hat{\beta}^L C_j^{L,2} - \hat{\alpha}^e z_{t-1}^j - \hat{\beta}^e z_{t-2}^j, \qquad j = 1, \ldots, k, && (3.1.1) \\
& |\alpha^k - \hat{\alpha}^k| \le \delta, \qquad \forall k \in \{H, M, L, e\}, && (3.1.2) \\
& |\beta^k - \hat{\beta}^k| \le \delta, \qquad \forall k \in \{H, M, L, e\}, && (3.1.3) \\
& 0 \le \hat{\alpha}^L \le \hat{\alpha}^M \le \hat{\alpha}^H, && (3.1.4) \\
& 0 \le \hat{\beta}^L \le \hat{\beta}^M \le \hat{\beta}^H, && (3.1.5)
\end{aligned}$$

where α^k and β^k are data denoting the current blood glucose parameter values, and $\hat{\alpha}^k$ and $\hat{\beta}^k$ are variables representing the new blood glucose parameter values. The constant θ controls how much emphasis we place on matching the new measurements, versus keeping the parameters similar to their old values. The variable δ represents the maximum difference between a new parameter value and a current parameter value, and is defined by constraints (3.1.2) and (3.1.3). Constraint (3.1.1) computes the blood glucose prediction corresponding to measurement M_j using the new parameters. Constraints (3.1.4) and (3.1.5) maintain the ordering of the parameters, in terms of glycemic index.

The optimization problem (3.1) is intended to personalize the blood glucose values, while accounting for noisy measurements. The objective tries to balance how different the new blood glucose predictions are from the actual measurements, and how different the new parameter values are from the current values.

Using common techniques, we can rewrite problem (3.1) as a linear optimization problem without the absolute values [18], which is given by (3.2):

$$
\begin{aligned}
\min \quad & \Theta \sum_{j=1}^{k} z_j + (1-\Theta)\delta && (3.2)\\
\text{s.t.} \quad & \widehat{BG}_j = f_j + \hat{\alpha}^H C_j^{H,1} + \hat{\alpha}^M C_j^{M,1} + \hat{\alpha}^L C_j^{L,1} + \hat{\beta}^H C_j^{H,2} \\
& \qquad + \hat{\beta}^M C_j^{M,2} + \hat{\beta}^L C_j^{L,2} - \hat{\alpha}^e z_{t-1}^j - \hat{\beta}^e z_{t-2}^j, && j = 1, \ldots, k,\\
& z_j \geq M_j - \widehat{BG}_j, && j = 1, \ldots, k,\\
& z_j \geq -M_j + \widehat{BG}_j, && j = 1, \ldots, k,\\
& \alpha^k - \hat{\alpha}^k \leq \delta, && \forall k \in \{H, M, L, e\},\\
& -\alpha^k + \hat{\alpha}^k \leq \delta, && \forall k \in \{H, M, L, e\},\\
& \beta^k - \hat{\beta}^k \leq \delta, && \forall k \in \{H, M, L, e\},\\
& -\beta^k + \hat{\beta}^k \leq \delta, && \forall k \in \{H, M, L, e\},\\
& 0 \leq \hat{\alpha}^L \leq \hat{\alpha}^M \leq \hat{\alpha}^H,\\
& 0 \leq \hat{\beta}^L \leq \hat{\beta}^M \leq \hat{\beta}^H.
\end{aligned}
$$

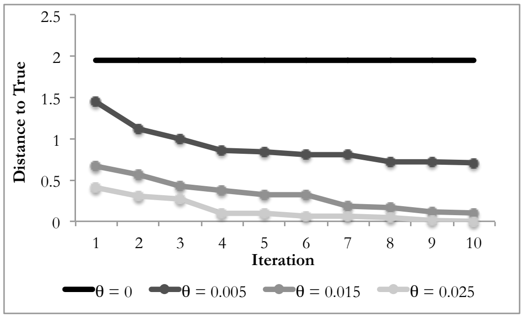
At the optimal solution, the variable z_j will be equal to $|M_j - \widehat{BG}_j|$, and the variable δ will be equal to the largest difference between the new parameters and old parameters. This is due to the fact that we are minimizing these variables. This problem can now easily be solved with general linear optimization solvers.

3.3.3 Empirical Evidence

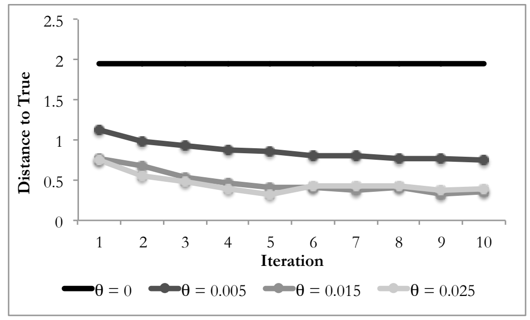
We will now present empirical evidence for the blood glucose personalization method. To show the ability of the method to learn new blood glucose parameters, we perform the following experiment. We first define "true" blood glucose parameters, which are the values we are trying to learn. We will present results here using five different "true" parameter values extracted from papers measuring postprandial (after meal) blood glucose values [36, 40, 65, 64, 75]. We then compute what the true blood glucose measurements would be using different amounts and types of carbohydrates. The carbohydrate content one hour and two hours before each measurement are randomly selected to have an amount of low, medium, and/or high carbohydrates from the set $\{0, 5, 10, 15, 20\}$, where at least one value is nonzero. This allows for different types of "meals". We then add uniformly distributed noise that is $\pm\sigma\%$ of the true measurement, where we vary σ from 0 to 20%. By adding uniformly distributed noise, we create a set of noisy measurements to represent what we might receive in practice.

Figure 3-3 presents results for which we use 10 measurements per iteration (instance of (3.1)) and 10 iterations. We selected 10 measurements here as a reasonable number of measurements that we could collect before updating the parameters. The results are averaged across five different true parameter values. Each subplot is for a different amount of noise, and the lines show how our learning of the true preferences changes with different values of θ, the constant in the objective that balances new measurements with current knowledge. The x-axis gives the iteration (the number of times we have solved (3.1), where each iteration is given 10 new measurements), and the y-axis gives the distance between the true parameter vector and the learned parameter vector.

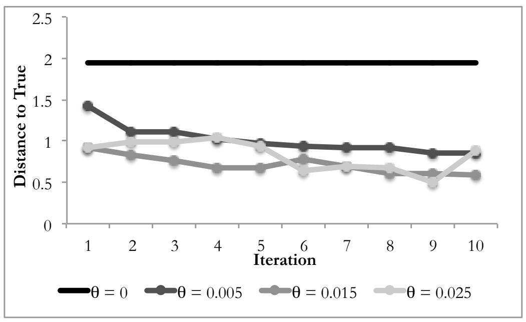
When there is no noise in the measurements, a larger θ is better (more emphasis on matching the measurements) and we learn quickly. However, with noise, $\theta = 0.015$ becomes more appealing, which means that it is advantageous to balance learning from the measurements with keeping some of our current knowledge of the parameters.



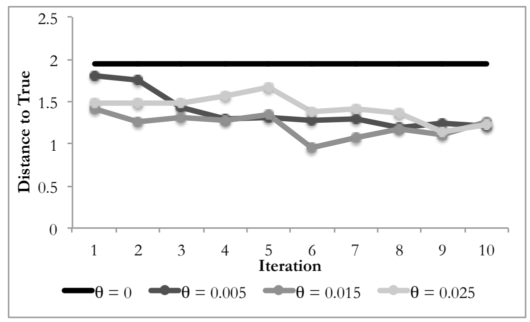
(a) $\sigma = 0\%$



(b) $\sigma = 5\%$



(c) $\sigma = 10\%$



(d) $\sigma = 20\%$

Figure 3-3: The distance between the learned parameter values and the true parameter values as a function of the number of iterations performed of the blood glucose update optimization problem (3.1).

Even with a large amount of noise, we are still learning over time and are closer to the true parameters than if we kept the initial parameters (which is shown by the solid black line in each plot).

3.4 Recipes and Nutritional Guidelines

Another critical input for our approach is recipe and nutritional information. This includes a database of recipes and foods, as well as a set of nutritional guidelines for each individual.

3.4.1 The Recipe Database

One of the goals of our system is to create healthy and appealing meal plans for real users. To do this, we have included recipes designed for diabetics from many different sources and selected with the assistance of a registered dietician [38, 7, 72, 86, 51, 45, 61, 6, 87, 56, 30, 74]. We have also included some prepared foods to reduce the preparation time for the users [60, 53], as well as some individual foods to serve as snacks or sides (fruit, nuts, yogurt, etc.). In total, our database of foods includes 678 recipes or individual foods for breakfast, lunch, dinner, and snacks. We construct each individual user's meal plans using a subset of these foods, depending on their food restrictions and preferences.

Each recipe has several features, described in Table 3.3. We computed the nutritional facts for each recipe by linking the ingredients to the United States Department of Agriculture (USDA) Nutrient Database, which is the standard reference for nutrient databases in the US [80]. The food group content information was added to insure that the user is not only meeting their daily nutritional requirements, but also their daily food group requirements [3, 48]. The food groups we consider are fruit, dairy, meat or meat substitutes, non-starchy vegetables, and starch.

Data	**Description**
Servings	The number of servings the recipe makes, and what constitutes a serving.
Ingredients	A list of ingredients in the recipe, giving the amount and description of each ingredient.
Allergens	Common allergens in the recipe, such as peanuts and shellfish.
Category	The type of recipe, such as main dish or side dish.
Nutrients	All nutritional information, computed from the ingredients.
Food Groups	All food group information, computed from the ingredients.

Table 3.3: Data stored for each recipe.

3.4.2 Nutritional Guidelines

We calculate the nutritional requirements for each individual using the Mifflin-St Jeor equation [58] to calculate the Basal Metabolic Rate (BMR) for an individual, and the number of calories they need using their activity level (the more active a person is, the more calories they need to consume). The BMR equation depends on an individual's weight, height, age and gender, and is given by the following:

$$
\begin{aligned}
\text{BMR} &= (10 \times \text{weight}) + (6.25 \times \text{height}) - (5 \times \text{age}) + 5, && \text{if male,} \\
\text{BMR} &= (10 \times \text{weight}) + (6.25 \times \text{height}) - (5 \times \text{age}) - 161, && \text{if female,}
\end{aligned}
$$

where weight is in kilograms and height is in centimeters. The necessary calories are then calculated based on the person's activity level as follows:

$$
\text{Total Daily Calories} = \begin{cases} 1.2 \times \text{BMR}, & \text{if sedentary,} \\ 1.375 \times \text{BMR}, & \text{if lightly active,} \\ 1.55 \times \text{BMR}, & \text{if moderately active,} \\ 1.7525 \times \text{BMR}, & \text{if very active,} \\ 1.9 \times \text{BMR}, & \text{if extra active.} \end{cases}
$$

We then calculate the amount of carbohydrates, fat, protein and other nutrients that each user should consume using the nutritional requirements described in Table 3.4. These guidelines were selected with the help of an experienced nutritionist and the American Diabetes Association. The first column lists the nutrient, the second column lists the units for each nutrient, and the third column lists the daily requirements for each nutrient. The units for the food groups are in servings, where the serving sizes are determined by the American Diabetes Association.

As an example, suppose the user is female, 40 years old, weighs 70 kilograms (154 pounds), is 170 centimeters tall (67 inches), and is lightly active. Then her BMR is:

$$
\text{BMR} = (10 \times 70) + (6.25 \times 170) - (5 \times 40) - 161 = 1401.5,
$$

and she should consume 1.375*1401.5 = 1,927 calories each day. Furthermore, we can use Table 3.4 to compute her other nutritional requirements. For carbohydrates,

Nutrient	Units	Requirements
Carbohydrates	grams (g)	35% – 45% of the total calories consumed, where each gram of carbohydrates contains 4 calories. If the user is vegetarian, the upper bound is increased to 60%.
Fiber	grams (g)	At least 14g per 1,000 calories consumed.
Total Fat	grams (g)	25% – 35% of the total calories consumed, where each gram of fat contains 9 calories.
Saturated Fat	grams (g)	Less than 7% of the total calories consumed.
Protein	grams (g)	25% – 35% of the total calories consumed, where each gram of protein contains 4 . If vegetarian, the lower bound is decreased to 10%.
Sodium	milligrams (mg)	Less than 1500mg.
Cholesterol	milligrams (mg)	Less than 300mg.
Fruit	ADA servings	2 – 4 servings.
Dairy	ADA servings	2 – 4 servings.
Meat & Meat Substitutes	ADA servings	5 – 10 servings. If vegetarian, 4 – 8 servings.
Nonstarchy vegetables	ADA servings	4 – 10 servings.
Starch	ADA servings	5 – 9 servings. If vegetarian, 5 – 12 servings.

Table 3.4: Nutritional Guidelines used to determine bounds and constraints for the optimization problem.

she should consume at least $\frac{(0.35*1,927)}{4} = 168.6$ grams, and at most $\frac{(0.45*1,927)}{4} = 216.8$ grams. For fiber, should should consume at least $14 * \frac{1,927}{1,000} = 27$ grams. Her requirements for total fat, saturated fat, and protein can be computed similarly.

3.5 Summary

In this chapter, we have presented models for blood glucose dynamics, a methodology for updating these models using data, and a description of the nutritional database we use. We showed that the blood glucose updates successfully learn individual parameter values, even in the presence of noise. Together with the preferences described in Chapter 2, these are the inputs needed for our optimization approach, which is the focus of the next chapter.

Chapter 4

Using Optimization to Generate Personalized and Robust Plans

Given the inputs discussed in Chapters 2 and 3, we can construct a mixed-integer optimization problem to find a personalized eating and exercise plan. In this chapter, we present the optimization problem, and then discuss how robust optimization can be used to lower the maximum blood glucose levels.

In Section 4.1, we generate a daily meal and exercise plan using mixed-integer optimization that minimizes blood glucose levels, maximizes personal preferences and accounts for nutritional constraints. In Section 4.2, we incorporate robust optimization into the plan generation to account for uncertainty in how blood glucose changes. Lastly, in Section 4.3, we present empirical evidence to show the strength of both the nominal and robust optimization problems.

4.1 The Nominal Problem

In this section, we present the mixed-integer optimization problem for finding a personalized meal and exercise plan. We will first give the notation for the data and the variables, and then describe the objective and constraints.

4.1.1 The Data and Variables

We consider a time horizon of one day, discretized into the set of hours $\mathcal{T} = \{1, \ldots, T\}$ where $T = 24$. We will assume that time $t = 1$ corresponds to 6am. We have a set of foods or recipes described in Section 3.4, which we denote by the set $\mathcal{F} = \{1, \ldots, F\}$. We categorize each food as a main dish, drink, extra item (fruits, nuts, yogurt, etc.), or side dish (salad, soup, vegetables, etc.). The main dishes are denoted by the set $\mathcal{M}$, the drinks are denoted by the set $\mathcal{D}$, the extra items are denoted by the set $\mathcal{E}$, and the side dishes are denoted by the set $\mathcal{S}$. Each food belongs to exactly one category.

Additionally, each food i has several nutritional attributes, including calories cal^i, carbohydrates c^i, and other nutrients which we will denote by $\{n_1^i, n_2^i, \ldots, n_N^i\}$. These include all of the nutrients and food groups discussed in Section 3.4. Table 4.1 gives a few examples of the nutritional data.

Nutrient	Apple	Walnuts	Chili	Smoothie	Egg Sandwich
Calories cal^i	65	196	148	207	271
Carbohydrates c^i	17	4	10	34	30
Fiber n_1^i	3	2	3	3	5
Total Fat n_2^i	0	19	4	1	8
Saturated Fat n_3^i	0	2	1	0	3
Protein n_4^i	0	4.5	19	17	21
Sodium n_5^i	1	0	435	216	395
Cholesterol n_6^i	0	0	49	6	218
Fruit n_7^i	1	0	0	1	0
Dairy n_8^i	0	0	0	1.5	0
Meat n_9^i	0	0.7	2.5	0	2
Non-starchy Vegetables n_{10}^i	0	0	1	0	0
Starch n_{11}^i	0	0	0	0	2

Table 4.1: Examples of the nutritional data for the optimization problem.

We also have preference data computed as described in Chapter 2. First, we will assume that we take the analytic center of the feasible utility space as the user's preferences, and then we will extend the formulation to include the robust approaches in Section 4.1.3. Denote the preferences computed using the analytic center by p_t^i for each time t and for each food i. To compute the preferences for different times, assume that five questionnaires are asked, for breakfast, lunch, afternoon snacks, dinner, and evening snacks. Each time falls into one of these categories, so there are five different preference values for each food. Differentiating between different times is important for preferences, since many users prefer different foods at different times (for example, someone might enjoy scrambled eggs at breakfast, but would not like scrambled eggs at dinner). The preferences take values in the range $[-1, 1]$, with -1 corresponding to the lowest possible preference, and 1 corresponding to the highest possible preference.

The following is a complete list of the data used:

$\mathcal{T} = \{1, \ldots, T\}$ = the set of time intervals (hours) in the time horizon, where $t = 1$ corresponds to 6am;

$\mathcal{F} = \{1, \ldots, F\}$ = the set of foods or recipes considered;

$\mathcal{M}$ = the set of foods or recipes that are main dishes;

$\mathcal{D}$ = the set of foods or recipes that are drinks;

$\mathcal{E}$ = the set of foods or recipes that are extra items (fruit, yogurt, nuts, etc.);

$\mathcal{S}$ = the set of foods or recipes that are side dishes (salad, soup, vegetables, etc.);

cal^i = the amount of calories (in kcal) in one serving of food i;

c^i = the amount of carbohydrates (in grams) in one serving of food i;

$\mathcal{N}^i = \{n_1^i, \ldots, n_N^i\}$ = the amount of each nutrient or food group in one serving of food i;

p_t^i = the preference of the user to eat food i at time t;

f = the fasting blood glucose level of the user;

λ = a parameter to represent the emphasis placed on minimizing blood glucose values in the objective;

α^i = the amount of increase in the blood glucose level of the user (in mg/dL) one hour after eating 1g of carbohydrates of food i;

β^i = the amount of increase in the blood glucose level of the user (in mg/dL) two hours after eating 1g of carbohydrates of food i;

α^e = the amount of decrease in the blood glucose level of the user (in mg/dL) one hour after exercising one minute;

β^e = the amount of decrease in the blood glucose level of the user (in mg/dL) two hours after exercising one minute;

u^i = the maximum number of servings that can be assigned of food i;

ℓ^i = the minimum number of servings that can be assigned of food i;

u^s = the maximum number of snacks that the user is willing to eat;

u^{exer} = the maximum number of minutes that the user is willing to exercise;

u^{BG} = the maximum allowable blood glucose level, in mg/dL.

Some of this data (e.g. the time period and nutrients) is the same for all users, and some of it (e.g. the set of foods allowed, the preferences, and the fasting level) is specific to each user.

The variables define which foods to eat, when, and how much to eat, as well as when to exercise and how much to exercise. The following is a list of the variables:

$$x_t^i = \begin{cases} \text{the amount in servings to eat of food } i \in \mathcal{F} \text{ at time } t \in \mathcal{T}, \\ \qquad \text{if food } i \text{ is consumed at time } t, \\ 0, \qquad \text{otherwise;} \end{cases}$$

$$y_t^i = \begin{cases} 1, & \text{if food } i \text{ is consumed at time } t, \\ 0, & \text{otherwise;} \end{cases}$$

$$m_t = \begin{cases} 1, & \text{if a meal is consumed at time } t, \\ 0, & \text{otherwise;} \end{cases}$$

$$s_t = \begin{cases} 1, & \text{if a snack is consumed at time } t, \\ 0, & \text{otherwise;} \end{cases}$$

$$z_t = \begin{cases} \text{the amount in minutes to exercise at time } t \in \mathcal{T}, \\ \quad \text{if exercise is scheduled at time } t, \\ 0, \quad \text{otherwise;} \end{cases}$$

$$w_t = \begin{cases} 1, & \text{if exercise is scheduled at time } t, \\ 0, & \text{otherwise;} \end{cases}$$

BG_t = the blood glucose level of the user at time t;

BG = the maximum blood glucose level of the user throughout the time horizon $\mathcal{T}$.

4.1.2 The Objective and Constraints

The nominal optimization problem is given by (4.1). For this problem, our objective has two components: minimizing blood glucose levels, and maximizing preferences. For the blood glucose component, instead of minimizing the sum of blood glucose levels throughout the day or the average blood glucose levels, we decided to minimize the maximum blood glucose level minus 140, which is the bound for tight blood glucose control [2]. This is due to our desire to create a plan without highly variable blood glucose levels, since highs and lows are very dangerous for diabetics [2]. The parameter λ controls how much emphasis we place on this piece of the objective function. With larger λ, the maximum blood glucose levels are typically lower, but the user does not prefer the plan as much. We will show another method for controlling this when we present the robust model.

For the preferences component, we maximize the sum of the learned preferences for the foods that the user consumes throughout the time period. This captures the user's preference for eating or not eating each food, and then the serving size is selected based on the nutritional and food group requirements, as well as the serving size bounds for each food, u^i and ℓ^i.

$$
\begin{aligned}
\text{minimize} \quad & \lambda(BG - 140) - \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{F}} p_t^i y_t^i && && (4.1)\\
\text{s.t.} \quad & 70 \le BG_t \le BG, && \forall t \in \mathcal{T}, && (4.1.1)\\
& BG_t \le u^{BG}, && \forall t \in \mathcal{T}, && (4.1.2)\\
& BG_1 = f, && && (4.1.3)\\
& BG_2 = f + \sum_{i\in\mathcal{F}}(\alpha^i c^i x_1^i) - \alpha^e z_1, && && (4.1.4)\\
& BG_t = f + \sum_{i\in\mathcal{F}}(\alpha^i c^i x_{t-1}^i + \beta^i c^i x_{t-2}^i) && && \\
& \qquad\qquad - \alpha^e z_{t-1} - \beta^e z_{t-2}, && \forall t \in \{3,\dots,T\}, && (4.1.5)\\
& \ell^i y_t^i \le x_t^i \le u^i y_t^i, && \forall t \in \mathcal{T}, i \in \mathcal{F}, && (4.1.6)\\
& \sum_{t\in\mathcal{T}} y_t^i \le 1, && \forall i \in \mathcal{M}, && (4.1.7)\\
& \sum_{t\in\mathcal{T}} y_t^i \le 2, && \forall i \notin \mathcal{M}, && (4.1.8)\\
& \sum_{t=1}^{5} m_t = 1, && && (4.1.9)\\
& \sum_{t=6}^{9} m_t = 1, && && (4.1.10)\\
& \sum_{t=12}^{17} m_t = 1, && && (4.1.11)\\
& \sum_{t\in\mathcal{T}} s_t \le u^s, && && (4.1.12)
\end{aligned}
$$

$$m_t + s_t \leq 1, \qquad \forall t \in \mathcal{T}, \quad (4.1.13)$$

$$\sum_{i \in \mathcal{M}} y_t^i = m_t, \qquad \forall t \in \mathcal{T}, \quad (4.1.14)$$

$$\sum_{i \in \mathcal{D}} y_t^i \leq m_t + s_t, \qquad \forall t \in \mathcal{T}, \quad (4.1.15)$$

$$\sum_{i \in \mathcal{E}} y_t^i \leq 2m_t + 2s_t, \qquad \forall t \in \mathcal{T}, \quad (4.1.16)$$

$$\sum_{i \in \mathcal{S}} y_t^i \leq 2m_t, \qquad \forall t \in \mathcal{T}, \quad (4.1.17)$$

$$Calories = \sum_{i \in \mathcal{F}} \sum_{t \in \mathcal{T}} cal^i x_t^i - \sum_{t \in \mathcal{T}} \mu * z_t, \quad (4.1.18)$$

$$\ell^{cal} \leq Calories \leq u^{cal}, \quad (4.1.19)$$

$$\ell^c * Calories \leq \sum_{i \in \mathcal{F}} \sum_{t \in \mathcal{T}} c^i x_t^i \leq u^c * Calories, \quad (4.1.20)$$

$$\ell^{n_j} * Calories \leq \sum_{i \in \mathcal{F}} \sum_{t \in \mathcal{T}} n_j^i x_t^i \leq u^{n_j} * Calories, \quad \forall j \in \{1, \ldots, N\}, \quad (4.1.21)$$

$$\sum_{t=0}^{T} z_t \leq u^{exer}, \quad (4.1.22)$$

$$10w_t \leq z_t \leq u^{exer} w_t, \qquad \forall t \in \mathcal{T}, \quad (4.1.23)$$

$$w_t \leq \sum_{i \in F} (y_t^i + y_{t-1}^i), \qquad \forall t \in \mathcal{T}, \quad (4.1.24)$$

$$x_t^i \in \mathbb{Z}, y_t^i \in \{0, 1\}, \qquad \forall i \in \mathcal{F}, t \in \mathcal{T}, \quad (4.1.25)$$

$$w_t \in \{0, 1\}, z_t \in \mathbb{Z}, m_t \in \{0, 1\}, s_t \in \{0, 1\}, \qquad \forall t \in \mathcal{T}. \quad (4.1.26)$$

The constraints (4.1.1)-(4.1.2) define the maximum blood glucose variable, and bound the blood glucose level BG_t at each time t. In practice, we define the upper bound u^{BG} as the maximum of 180 and $(f + 50)$, where f is the fasting level of the user. These bounds were selected using the advice of the American Diabetes Association [8]. The constraints (4.1.3)-(4.1.5) capture the blood glucose dynamics discussed in Section 3.2.

Constraints (4.1.6)-(4.1.8) restrict the amount of each food that the user can consume. Constraint (4.1.6) limits the number of servings of each food to be between the lower bound ℓ^i and the upper bound u^i. Constraints (4.1.7) and (4.1.8) limit the number of times the user eats each food throughout the day to at most once for the set of main dishes (denoted by $\mathcal{M}$), and at most twice for the set of non-main dishes. This prevents the same foods from being eaten too often. We refer to constraints of this type as "appeal" constraints, since they are added with the intention of making the plan more appealing to the user.

Constraints (4.1.9)-(4.1.17) define the meal and snack timing and composition. Constraints (4.1.9)-(4.1.11) force breakfast to be between 6am and 10am, lunch to be between 11am and 2pm, and dinner to be between 5pm and 10pm. Constraint (4.1.12) limits the number of snacks to be less than an upper bound number of snacks, which is user defined. This number is typically between zero and three. Constraint (4.1.13) prevents meals and snacks from occurring at the same time. Constraints (4.1.14)-(4.1.17) restrict the composition of each meal or snack. Breakfast, lunch, and dinner are constrained to have one main dish, at most one drink, and at most two optional extra items, such as fruits, yogurt or nuts. Dinner is also allowed to have up to two side dishes, which include vegetables or grains. Snacks are limited to only contain extra items. These constraints are necessary to create appealing menus for the users. The specific main dishes, drinks, extras, or side dishes that are allowed at each meal or snack are determined by the preferences, or can be further restricted to only allow dishes that are appropriate at each time.

Constraints (4.1.18)-(4.1.21) define the nutritional and food group requirements. Constraint (4.1.18) defines the total calories as the total calories consumed minus the total calories burned, where μ is a constant defining how many calories one minute of exercise burns. We compute this constant using research on Metabolic Equivalent of Task (MET), which defines the number of calories burned for different types of exercise [4]. Constraint (4.1.19) bounds the total calories, by upper and lower bounds that are determined by the Mifflin-St Jeor equation, as discussed in Section 3.4.2. Constraint (4.1.20) defines the carbohydrate constraint, which restricts the total carbohydrates to be between ℓ^c and u^c times the total calories. Constraints (4.1.21) are similar constraints for each of the other nutrients and food groups, where the upper and lower bounds often depend on the total calories or the attributes of the user (age, weight, or gender). For some nutrients, such as sodium, the lower bound is zero since there is no minimum requirement.

Constraints (4.1.22)-(4.1.24) restrict the amount and timing of exercise. Constraint (4.1.22) gives an upper bound on the total amount of exercise that is done in the time period. Constraint (4.1.23) limits the exercise at each time t to be between 10 minutes and u^{exer} minutes long. This prevents unreasonably short or unreasonably long exercise periods. In practice, the default value for u^{exer} is typically 30 minutes. Constraint (4.1.24) limits the exercise to only occur at times the user is eating food or ate food in the last hour. This constraint is necessary because of the sensitivity of diabetics to blood glucose changes. We use exercise to decrease blood glucose levels, but exercising without eating can decrease blood glucose levels to dangerous ranges [2].

Lastly, constraints (4.1.25) and (4.1.26) define the variables to be either binary or integer. In the remainder of this Chapter, we will denote constraints (4.1.6)-(4.1.26) by the feasible set $\mathbf{X}$.

In practice, we often want to produce solutions for multiple days, with variety between the days. We can do this in multiple ways, including solving for multiple days at once (increase the time horizon), or iteratively solving a day at a time while encouraging variety.

4.1.3 Incorporating the Robust Preference Models

In the previous section, we presented the optimization formulation assuming that we take the analytic center of the feasible preference space to be the user's preferences. However, we showed in Chapter 2 that the robust approaches, particularly the CVaR approach, produce more appealing solutions in terms of preferences. In this section, we show how (4.1) can be extended to include these approaches.

Suppose that the feasible set of utilities for time t is given by $\mathcal{U}_t$. The optimization problem with the robust approach to preferences is given by (4.2).

$$\min_{\mathbf{x},\mathbf{y},\mathbf{w},\mathbf{z},\mathbf{m},\mathbf{s}} \left[\max_{\mathbf{p}_t \in \mathcal{U}_t} \quad \lambda(BG - 140) - \sum_{t \in \mathcal{T}} \mathbf{p}_t' \mathbf{y}_t \right] \tag{4.2}$$

$$\begin{aligned}
\text{s.t.} \quad & 70 \leq BG_t \leq BG, && \forall t \in \mathcal{T}, \\
& BG_t \leq u^{BG}, && \forall t \in \mathcal{T}, \\
& BG_1 = f, \\
& BG_2 = f + \sum_{i \in \mathcal{F}} (\alpha^i c^i x_1^i) - \alpha^e z_1, \\
& BG_t = f + \sum_{i \in \mathcal{F}} (\alpha^i c^i x_{t-1}^i + \beta^i c^i x_{t-2}^i) - \alpha^e z_{t-1} - \beta^e z_{t-2}, && \forall t \in \{3, \ldots, T\}, \\
& (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \mathbf{m}, \mathbf{s}) \in \mathbf{X}.
\end{aligned}$$

Similarly, the optimization problem with the CVaR approach to preferences is given by (4.3). The vectors $\{\mathbf{p}_t^1, \ldots, \mathbf{p}_t^N\} \in \mathcal{U}_t$ are the sampled utility vectors for time t. The variables $\{v_t^1, \ldots, v_t^N\}$ represent the sampled points that are selected to be the worst $\alpha\%$ of utility vectors in the feasible space.

$$\min_{\mathbf{x},\mathbf{y},\mathbf{w},\mathbf{z},\mathbf{m},\mathbf{s}} \left[\max_{\mathbf{v}_1, \ldots, \mathbf{v}_T} \quad \lambda(BG - 140) - \sum_{t \in \mathcal{T}} \sum_{j=1}^{N} \frac{1}{\alpha N} ((\mathbf{p}_t^j)' \mathbf{y}_t) v_t^j \right] \tag{4.3}$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{j=1}^{N} v_t^j = \alpha N, && \forall t \in \mathcal{T}, \\
& 0 \leq v_t^j \leq 1, && \forall t \in \mathcal{T}, j = 1, \ldots, N, \\
& 70 \leq BG_t \leq BG, && \forall t \in \mathcal{T}, \\
& BG_t \leq u^{BG}, && \forall t \in \mathcal{T}, \\
& BG_1 = f, \\
& BG_2 = f + \sum_{i \in \mathcal{F}} (\alpha^i c^i x_1^i) - \alpha^e z_1, \\
& BG_t = f + \sum_{i \in \mathcal{F}} (\alpha^i c^i x_{t-1}^i + \beta^i c^i x_{t-2}^i) - \alpha^e z_{t-1} - \beta^e z_{t-2}, && \forall t \in \{3, \ldots, T\}, \\
& (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \mathbf{m}, \mathbf{s}) \in \mathbf{X}.
\end{aligned}$$

As was done in Chapter 2, we can take the dual of the inner optimization problem, and reformulate this as a mixed-integer problem. In the next Section, we will again assume that the preferences are represented as the analytic center of the feasible utility space, but by using formulations (4.2) or (4.3), we can include the Robust or CVaR approaches if desired.

4.2 The Robust Optimization Problem

In this section, we present the robust counterpart of formulation (4.1) that is robust to the blood glucose parameters. As mentioned in Chapter 3, blood glucose measurements are often up to 20% inaccurate for many reasons: the device itself, the environment of the user, the adherence of the user to the protocol of the measurement device, and the adherence of the user to the eating and exercise plan [41]. We also may not have learned enough about the user's personal measurements that the parameters match their personal blood glucose behavior exactly. For all of these reasons, we use robust optimization to guarantee that our blood glucose upper bound is still satisfied even with uncertain parameters.

The robust optimization problem is given by (4.4). The objective and constraints (4.4.1)-(4.4.4) and (4.4.7) are the same as their counterparts in (4.1). We altered constraints (4.4.5) and (4.4.6) to be robust to the blood glucose parameters in the uncertainty set $\mathcal{A}$. This technique is presented by Bertsimas and Sim [16].

$$\text{minimize} \quad \lambda(BG - 140) - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{F}} p_t^i y_t^i \tag{4.4}$$

$$\text{s.t.} \quad 70 \leq BG_t \leq BG, \qquad \forall t \in \mathcal{T}, \tag{4.4.1}$$

$$BG_1 = f, \tag{4.4.2}$$

$$BG_2 = f + \sum_{i \in \mathcal{F}} (\alpha^i c^i x_1^i) - \alpha^e z_1, \tag{4.4.3}$$

$$BG_t = f + \sum_{i \in \mathcal{F}} (\alpha^i c^i x^i_{t-1} + \beta^i c^i x^i_{t-2}) - \alpha^e z_{t-1} - \beta^e z_{t-2}, \qquad 3 \le t \le T, \tag{4.4.4}$$

$$\max_{\tilde{\alpha}^i, \tilde{\alpha}^e \in \mathcal{A}} \left[f + \sum_{i \in \mathcal{F}} (\tilde{\alpha}^i c^i x^i_1) - \tilde{\alpha}^e z_1 \right] \le u^{BG}, \tag{4.4.5}$$

$$\max_{\substack{\tilde{\alpha}^i, \tilde{\alpha}^e \in \mathcal{A} \\ \tilde{\beta}^i, \tilde{\beta}^e \in \mathcal{A}}} \left[f + \sum_{i \in \mathcal{F}} (\tilde{\alpha}^i c^i x^i_{t-1} + \tilde{\beta}^i c^i x^i_{t-2}) - \tilde{\alpha}^e z_{t-1} - \tilde{\beta}^e z_{t-2} \right] \le u^{BG}, \quad 3 \le t \le T, \tag{4.4.6}$$

$$(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \mathbf{m}, \mathbf{s}) \in \mathbf{X}, \tag{4.4.7}$$

where

$$\begin{aligned} \mathcal{A} = \{\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}} \mid\ & \tilde{\alpha}^i \in [\alpha^i - \hat{\alpha}^i, \alpha^i + \hat{\alpha}^i],\ \ \tilde{\alpha}^e \in [\alpha^e - \hat{\alpha}^e, \alpha^e + \hat{\alpha}^e], \\ & \tilde{\beta}^i \in [\beta^i - \hat{\beta}^i, \beta^i + \hat{\beta}^i],\ \ \tilde{\beta}^e \in [\beta^e - \hat{\beta}^e, \beta^e + \hat{\beta}^e], \\ & \left| \frac{\tilde{\alpha}^e - \alpha^e}{\hat{\alpha}^e} \right| + \left| \frac{\tilde{\beta}^e - \beta^e}{\hat{\beta}^e} \right| + \sum_{i \in \mathcal{F}} \left| \frac{\tilde{\alpha}^i - \alpha^i}{\hat{\alpha}^i} \right| + \sum_{i \in \mathcal{F}} \left| \frac{\tilde{\beta}^i - \beta^i}{\hat{\beta}^i} \right| \le \Gamma \}. \end{aligned}$$

The nominal values of the parameters are given by α^i, β^i, α^e and β^e; the robust values of the parameters are given by $\tilde{\alpha}^i$, $\tilde{\beta}^i$, $\tilde{\alpha}^e$ and $\tilde{\beta}^e$; and the allowable deviations from the nominal values (the uncertainty set ranges) are given by $\hat{\alpha}^i$, $\hat{\beta}^i$, $\hat{\alpha}^e$ and $\hat{\beta}^e$. The parameter Γ controls the degree of conservatism of the model. While we still have a parameter λ here that controls the tradeoff between blood glucose and preferences, we will show in Section 4.3 that Γ serves the same purpose and eliminates the need for λ.

Following the procedure outlined by Bertsimas and Thiele [17], we can define the *scaled deviations* as $q^i = \frac{\tilde{\alpha}^i - \alpha^i}{\hat{\alpha}^i}$, $q^e = \frac{\tilde{\alpha}^e - \alpha^e}{\hat{\alpha}^e}$, $r^i = \frac{\tilde{\beta}^i - \beta^i}{\hat{\beta}^i}$, and $r^e = \frac{\tilde{\beta}^e - \beta^e}{\hat{\beta}^e}$. Then we can reformulate (4.4) and the uncertainty set $\mathcal{A}$ as the optimization problem (4.5) and uncertainty set Ω.

$$
\begin{aligned}
\text{minimize}\quad & \lambda(BG-140)-\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{F}}p_t^i y_t^i && (4.5)\\
\text{s.t.}\quad & 70\le BG_t\le BG, && \forall t\in\mathcal{T},\\
& BG_1=f,\\
& BG_2=f+\sum_{i\in\mathcal{F}}(\alpha^i c^i x_1^i)-\alpha^e z_1,\\
& BG_t=f+\sum_{i\in\mathcal{F}}(\alpha^i c^i x_{t-1}^i+\beta^i c^i x_{t-2}^i)-\alpha^e z_{t-1}-\beta^e z_{t-2}, && 3\le t\le T,\\
& BG_2+\max_{q^i,q^e\in\Omega}\left[\sum_{i\in\mathcal{F}}(\hat{\alpha}^i c^i x_1^i q^i)-\hat{\alpha}^e z_1 q^e\right]\le u^{BG},\\
& BG_t+\max_{\substack{q^i,q^e\in\Omega\\ r^i,r^e\in\Omega}}\left[\sum_{i\in\mathcal{F}}(\hat{\alpha}^i c^i x_{t-1}^i q^i+\hat{\beta}^i c^i x_{t-2}^i r^i)-\hat{\alpha}^e z_{t-1}q^e-\hat{\beta}^e z_{t-2}r^e\right]\le u^{BG}, && 3\le t\le T,\\
& (\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{w},\mathbf{m},\mathbf{s})\in\mathbf{X},
\end{aligned}
$$

where

$$
\begin{aligned}
\Omega=\{\mathbf{q},\mathbf{r}\ \mid\ & |q^i|\le 1,\ \ |q^e|\le 1,\ \ |r^i|\le 1,\ \ |r^e|\le 1,\\
& |q^e|+|r^e|+\sum_{i\in\mathcal{F}}|q^i|+|r^i|\le\Gamma\}.
\end{aligned}
$$

Then by applying strong duality arguments to the maximization problems in the constraints, we can reformulate the robust optimization problem as (4.6), which is also a mixed-integer optimization problem.

$$
\begin{aligned}
\text{minimize}\quad & \lambda(BG-140)-\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{F}}p_t^i y_t^i && (4.6)\\
\text{s.t.}\quad & 70\le BG_t\le BG, && \forall t\in\mathcal{T},\\
& BG_1=f,\\
& BG_2=f+\sum_{i\in\mathcal{F}}(\alpha^i c^i x_1^i)-\alpha^e z_1,
\end{aligned}
$$

$$
\begin{aligned}
& BG_t = f + \sum_{i \in \mathcal{F}} (\alpha^i c^i x^i_{t-1} + \beta^i c^i x^i_{t-2}) - \alpha^e z_{t-1} - \beta^e z_{t-2}, && 3 \le t \le T, \\
& BG_2 + \Gamma \rho_2 + \sum_i \delta^i_2 - \delta^e_2 \le u^{BG}, && \\
& BG_t + \Gamma \rho_t + \sum_i (\delta^i_t + \gamma^i_t) - \delta^e_t - \gamma^e_t \le u^{BG}, && 3 \le t \le T, \\
& \delta^i_t + \rho_t \ge c^i x^i_{t-1} \hat{\alpha}^i, && \forall i \in \mathcal{F}, 2 \le t \le T, \\
& \gamma^i_t + \rho_t \ge c^i x^i_{t-2} \hat{\beta}^i, && \forall i \in \mathcal{F}, 3 \le t \le T, \\
& -\delta^e_t + \rho_t \ge z_{t-1} \hat{\alpha}^e, && 2 \le t \le T, \\
& -\gamma^e_t + \rho_t \ge z_{t-2} \hat{\beta}^e, && 3 \le t \le T, \\
& \delta^i_t \ge 0, && \forall i \in \mathcal{F}, 2 \le t \le T, \\
& \gamma^i_t \ge 0, && \forall i \in \mathcal{F}, 3 \le t \le T, \\
& \delta^e_t \le 0, && 2 \le t \le T, \\
& \gamma^e_t \le 0, && 3 \le t \le T, \\
& \rho_t \ge 0, && 2 \le t \le T, \\
& (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \mathbf{m}, \mathbf{s}) \in \mathbf{X}, &&
\end{aligned}
$$

where ρ, δ, and γ are the dual variables for the inner maximization problems.

4.3 Empirical Evidence

We have implemented our approach as an online software, and in this section we present empirical evidence for the different optimization methodologies presented. All of the optimization problems were solved using Gurobi 5.0.

4.3.1 The Nominal Problem

We will first present results for the nominal problem. Although the basic formulations and some of the more universal data does not change for each individual user, the personal data does change, which alters the solution time and quality for each user. For this reason, we will report results for 10 different users. We generated input data for 10 real users of the system, who entered their personal information and completed the food survey. They had a wide range of personal attributes and preferences, which are summarized in Table 4.3. The BMI column gives the Body Mass Index of each user, which is a measure of the user's body shape, using their height and weight. Normal BMI range is below 25.0, a user with a BMI between 25.0 and 30.0 is considered overweight, and a user with a BMI greater than 30.0 is considered obese. As can be seen in Table 4.3, most of the users of our system are either overweight or obese. NumTimes gives the number of times the user allowed for eating meals. A value of three means that they specified their desired breakfast, lunch, and dinner times exactly. A value greater than three means they allowed more flexibility for their meal times. NumFoods is the number of recipes or foods that were available for the user after defining their preferences and food restrictions. Calories gives the number of daily calories the user should consume, accounting for any weight loss goals the user might have. To account for weight loss goals, for every pound that the user wants to lose per week, we subtract 500 calories from the total number of calories the user should consume, within reason (a person can lose up to two pounds per week, as long as the calories stay above 1500 for men, and 1200 for women).

User	Age	Gender	BMI	NumTimes	NumFoods	Calories
1	40	Female	33.2	3	358	1437
2	38	Female	29.6	7	511	1838
3	49	Male	29.4	3	482	1912
4	28	Male	35.5	4	342	2228
5	36	Male	28.0	5	373	1989
6	27	Female	33.5	3	480	1777
7	26	Female	22.9	3	388	1760
8	47	Female	33.7	8	482	1922
9	40	Male	28.0	3	354	1500
10	37	Female	22.0	5	181	1452

Table 4.3: Attributes of 10 different users of the system.

Figure 4-1 gives the LP Bound Gap and the Optimality Gap as a function of time, averaged across the 10 users, where the gaps are defined as:

$$\text{LP Bound Gap} = \frac{|\text{Best Integer} - \text{LP Bound}|}{|\text{LP Bound}|}$$

$$\text{Optimality Gap} = \frac{|\text{Best Integer} - \text{Optimal}|}{|\text{Optimal}|}$$

Best Integer denotes the objective function value of the best feasible integer solution that has been found, and Optimal denotes the objective function value of the optimal solution. The problem instances after presolve in Gurobi 5.0 had on average 6,225 rows and 4,633 columns.

The Optimality Gap quickly decreases, and by 200 seconds, we have found the optimal solution for almost all of the users. By 120 seconds, we are within 10% of the optimal solution on average. The LP Bound Gap takes longer to decrease, and it takes a significant amount of time for most users to close the gap, even though we have found either the optimal solution, or a solution very close to the optimal solution. This indicates that finding good solutions is easy, but proving optimality can be hard.

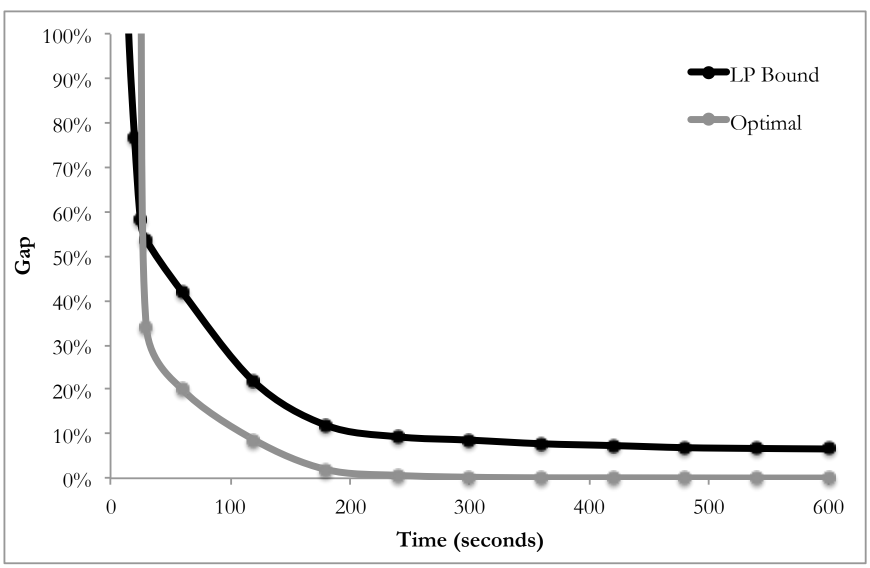


Figure 4-1: Plot of the LP Bound Gap and the Optimality Gap, averaged across 10 users.

Table 4.4 gives more information about the solution times (in seconds) for all users. While the time to a provable optimal solution can be long, the optimal solution is often found much earlier. Furthermore, the time to a "good" feasible solution (defined as one within 10% of the optimal solution) is often much shorter. Additionally, for several users, the optimal solution is found quickly. The longer solution times are highly correlated with the number of feasible eating times allowed by the user. The users who specify the exact times they would like to eat breakfast, lunch, and dinner (users 1, 3, 7, and 9) have much faster solution times, while the users who allow many times for meals (users 2 and 8) have much slower solution times. We can improve this by restricting the user to select fewer feasible times.

User	Time Optimal Proven (sec)	Time Optimal Found (sec)	Time to 10% Opt Gap (sec)
1	25	25	5
2	5825	970	30
3	47	40	5
4	1962	130	130
5	3041	473	10
6	343	342	240
7	31	31	5
8	7605	2302	60
9	71	65	5
10	86	66	15
AVG	1904	444	51

Table 4.4: Nominal solution times for all users, in seconds.

Table 4.5 gives the optimal solution for User 9, and a description of the key nutrients, preferences and blood glucose predictions for each meal or snack. The timing of the meals and the number of snacks were selected by the user. The calories, carbs, protein, and fat give the total amount of each nutrient in each meal or snack. The Pref column gives the average preference of the user for the foods in each meal or snack, on a scale from -1 to 1 (-1 is the least preferred, and 1 is the most preferred). The BG column gives the predicted blood glucose level of the user after eating each meal or snack, in mg/dL.

Table 4.6 gives a "greedy" solution for the same user, where the most preferred main dishes were forced when solving the optimization problem. This allows us to increase the preferences slightly for the meals, but at a significant cost in terms of blood glucose levels. This shows the ability of our approach to balance the two objectives.

Classification (Time)	Food or Exercise	Cal	Carbs	Prot	Fat	Pref	BG
Breakfast (6am)	Egg and Cheese Sandwich Plain Greek Yogurt	371	37	39	8	0.55	157
Lunch (12pm)	Pasta Salad with Mixed Vegetables Brazil Nuts Celery Exercise (30 min)	438	52	15	23	0.38	151
Snack (3pm)	Orange Soymilk	137	28	7	0	0.04	147
Dinner (7pm)	Spinach Stuffed Sole Broccoli Rabe with Toasted Garlic Quinoa Hazelnuts Apricot	463	35	34	24	0.51	155
Snack (10pm)	Orange Plain Greek Yogurt	169	25	19	0	0.02	143

Table 4.5: The Optimal Meal and Exercise Plan for User 9 ($\sim$ 1600 calories)

4.3.2 The Robust Optimization Problem

We solved the robust optimization problem (4.6) for all 10 users, with varying values of Γ. We selected the uncertainty sets so that the parameters can be up to 25% larger than the nominal values, which is a little more than the uncertainty seen in practice [41].

The results for some values of Γ are presented in Table 4.7. The first column for each value of Γ gives the total preference of the generated plan, divided by the total preference of the nominal plan. This is the piece of the objective that we expect to get worse (smaller) with larger Γ, since we expect the user to prefer the nominal plan

Classification (Time)	Food or Exercise	Cal	Carbs	Prot	Fat	Pref	BG
Breakfast (6am)	All Bran with Nonfat Milk Plain Greek Yogurt Hazelnuts Exercise (10 min)	435	64	32	10	0.56	179
Lunch (12pm)	Philly Steak Sandwich Plain Greek Yogurt Cucumber	462	35	50	14	0.52	171
Snack (3pm)	Raspberries Graham Crackers	153	30	4	3	0.04	170
Dinner (7pm)	Spinach Stuffed Sole Black-Eyed Peas Sauteed Zucchini Coins Carrot Spice Quickbread Plum	406	42	29	15	0.51	180
Snack (10pm)	Raspberries Cucumber	72	17	2	1	0.02	132

Table 4.6: A Greedy Meal and Exercise Plan for User 9 ($\sim$ 1600 calories)

to a plan that has lower blood glucose values. The second column for each value of Γ is the maximum blood glucose value of the robust solution, and what we hope to decrease with larger Γ. The bold rows are those for which the maximum blood glucose level in the nominal problem is close to the upper bound of 180.

User	$\Gamma = 0$		$\Gamma = 1$		$\Gamma = 2$		$\Gamma = 3$	
	Pref	MaxBG	Pref	MaxBG	Pref	MaxBG	Pref	MaxBG
1	1.00	148	1.00	148	1.00	148	1.00	148
2	1.00	144	1.00	144	0.99	143	0.98	143
3	1.00	166	0.99	160	0.99	156	0.98	152
4	**1.00**	**179**	**0.96**	**171**	**0.95**	**167**	**0.90**	**162**
5	1.00	155	1.00	155	0.99	154	0.92	154
6	**1.00**	**180**	**0.94**	**169**	**0.84**	**167**	**0.90**	**163**
7	1.00	150	1.00	150	1.00	150	0.99	150
8	1.00	168	0.98	161	0.91	154	0.91	154
9	1.00	158	0.99	148	0.99	148	0.99	148
10	**1.00**	**175**	**0.98**	**164**	**0.91**	**163**	**0.92**	**154**
AVG	1.00	162	0.99	157	0.96	155	0.96	153

Table 4.7: The preferences and maximum blood glucose values for the robust problem with various values of Γ.

For most users, the maximum blood glucose value either stays the same (if it was low in the nominal problem) or decreases with increasing Γ. For the users with high blood glucose values in the nominal problem, the robust approach serves to decrease the maximum blood glucose value, at only a minor cost to the overall preferences.

The average results for a wide range of Γ are shown in Figure 4-2. To visualize the change in preferences and the change in maximum blood glucose in the same plot, we normalized both values by dividing by the nominal value. The x-axis gives the value of Γ, and the y-axis gives the normalized outcome, either preferences or blood glucose. A value of 1.0 means that the preferences or maximum blood glucose values are the same as those in the nominal solution. As expected, both curves decrease with larger Γ. In general, the maximum blood glucose value decreases more as Γ increases, which is a nice property of this approach. For large Γ, the blood glucose curve flattens out since it becomes infeasible to reduce the blood glucose values further without violating nutritional constraints. The preferences curve becomes more noisy with large Γ. Looking at this curve, we would recommend a value of Γ less than or equal to three, since this is when the blood glucose curve flattens out and very little is gained from increasing Γ.

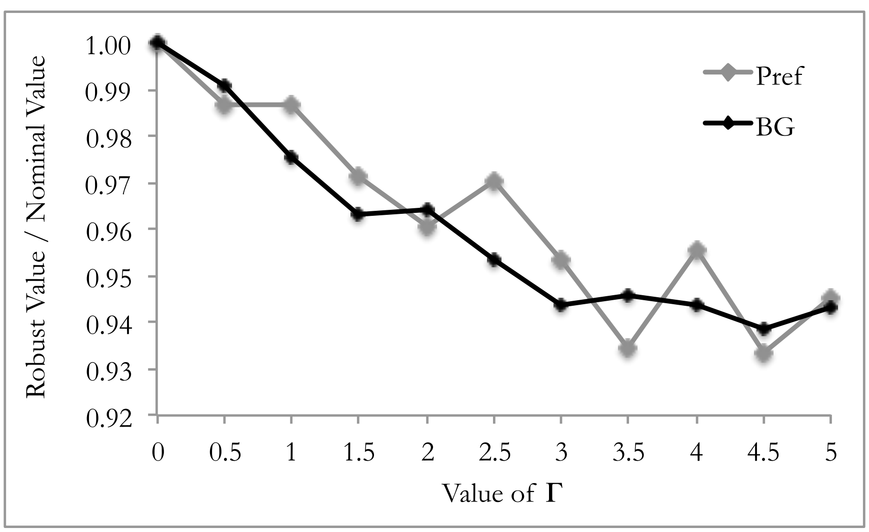


Figure 4-2: Plot of the normalized preferences and maximum blood glucose values as functions of Γ.

The specific value of Γ used in practice should be selected by a medical decision maker, and should depend on the individual in question. One strategy would be to slowly decrease Γ over time. When the user first starts using the generated plan, they will have higher blood glucose values, but will prefer the plan more. As they continue to use the plan and Γ is increased, they will gain better blood glucose control at a slight cost in terms of preferences.

4.4 Summary

In this chapter, we presented a mixed-integer robust optimization approach to finding a meal and exercise plan for a diabetic individual. This optimization approach relied on the inputs presented in Chapters 2 and 3. We also provided empirical evidence that the approach produces high quality solutions in a reasonable amount of time, and that the robust approach successfully controlled the trade-off between blood glucose levels and preferences.

The use of the optimization approach presented here can help create individualized treatment plans for each patient, catered to their preferences, nutritional requirements, and blood glucose measurements. This can be very challenging for healthcare providers, since each patient generates a significant amount of data. We believe that this approach has the ability to more efficiently provide lifestyle treatment regimens for diabetic individuals.

Chapter 5

Conclusion

In this chapter, we conclude by showing the overall system we have developed, and then we provide some concluding remarks.

5.1 The Overall System

We have implemented our approach as an online application for diabetes and diet management. In this section, we give an overview of the online system and how each of the individual pieces described in this thesis come together to make a comprehensive system.

Our online application is called LiA, which stands for "Lifestyle Analytics". Figure 5-1 shows the homepage, or menu. This is the first page a user sees when they register for LiA. Each of the icons takes the user to another webpage to fill out information or to perform an activity.

The first step for a new user is to visit the preferences page, shown in Figure 5-2. This is where the user enters their personal statistics (age, height, weight, and gender), which are used when computing nutritional needs. They also have the ability to select the number of snacks they are willing to eat, and specify their preferred meal times. Lastly, they enter their goals, either managing blood glucose levels, managing weight, or both. Based on their goals, we ask for additional information about their blood glucose levels and their weight goals.

After entering their preferences, the user visits the food restrictions page, shown in Figure 5-3. This asks the user about common food restrictions, whether or not they want to use a specific brand of prepared foods (My Fit Foods), and whether or not they would like alcoholic beverages in their plans. Any information provided on this page influences the foods and recipes shown throughout the rest of the system.

Figure 5-1: The menu webpage for the online application LiA.

The next step is the food questionnaire, which was the topic of Chapter 2. Figure 5-4 gives a snapshot of one of the questions that could be asked in the breakfast component. We ask the user 10 questions for breakfast, 15 questions for lunch, and 20 questions for dinner. The survey is fast, and enjoyable for most users. This is the last step of the initial data collection, which only needs to be completed when the user first starts using the system.

After completing the survey, the user can go to the plans page to generate and look at their weekly meal and exercise plans. Figure 5-5 shows the weekly plan summary, and part of a daily plan. All plans are generated using the optimization methods presented in Chapter 4. Each meal or snack is listed, and there are places for the user

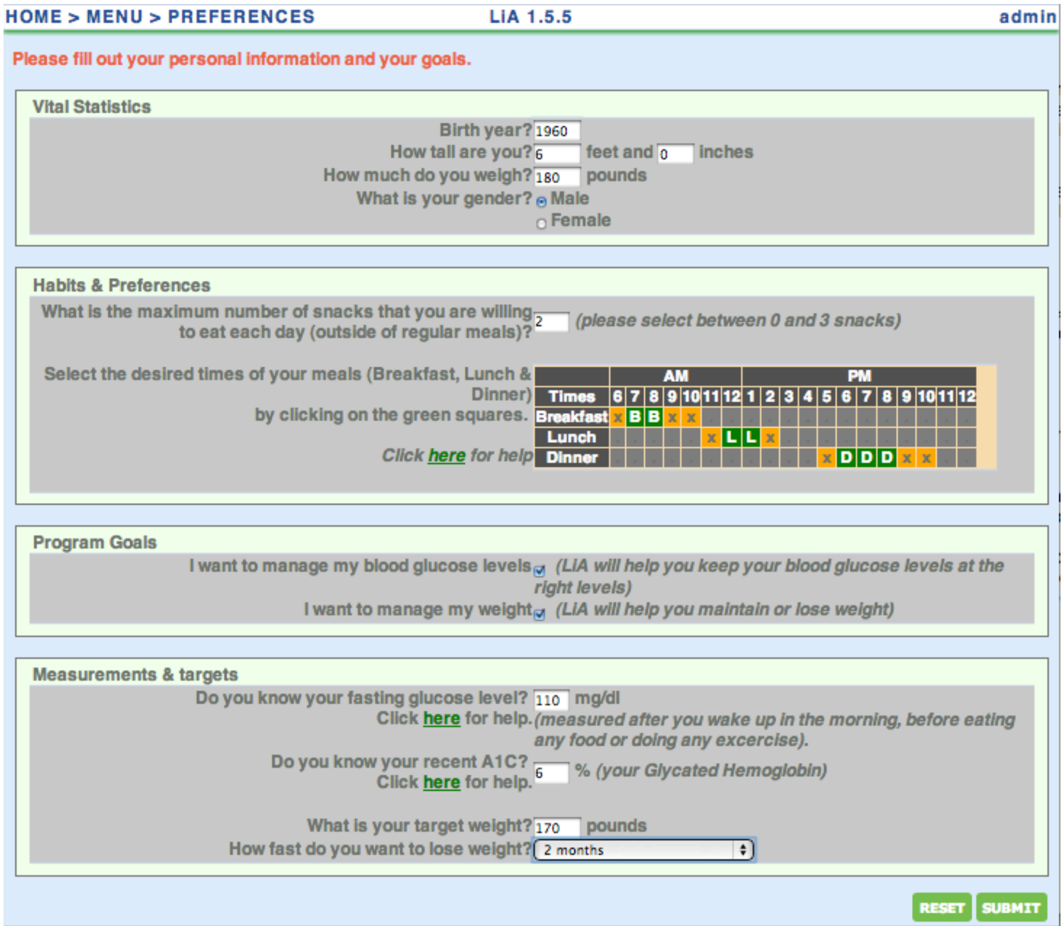


Figure 5-2: The preferences webpage for the online application LiA.

to enter blood glucose measurements and/or weight measurements. There are also many other features on the plans pages, including the ability to eliminate foods or recipes, print the weekly plan, and regenerate plans given any adjustments that have been made. The user also has the ability to switch to a different week, and generate plans for the future.

Figure 5-6 gives a snapshot of output displayed at the end of each daily plan. We give the user a nutritional summary, as well as a projected blood glucose curve given the meal and exercise plan. These were generated using the models and data discussed in Chapter 3.

The user can also see the entire library of recipes and foods, check their blood glucose and weight progress, and send messages to the administrative team.

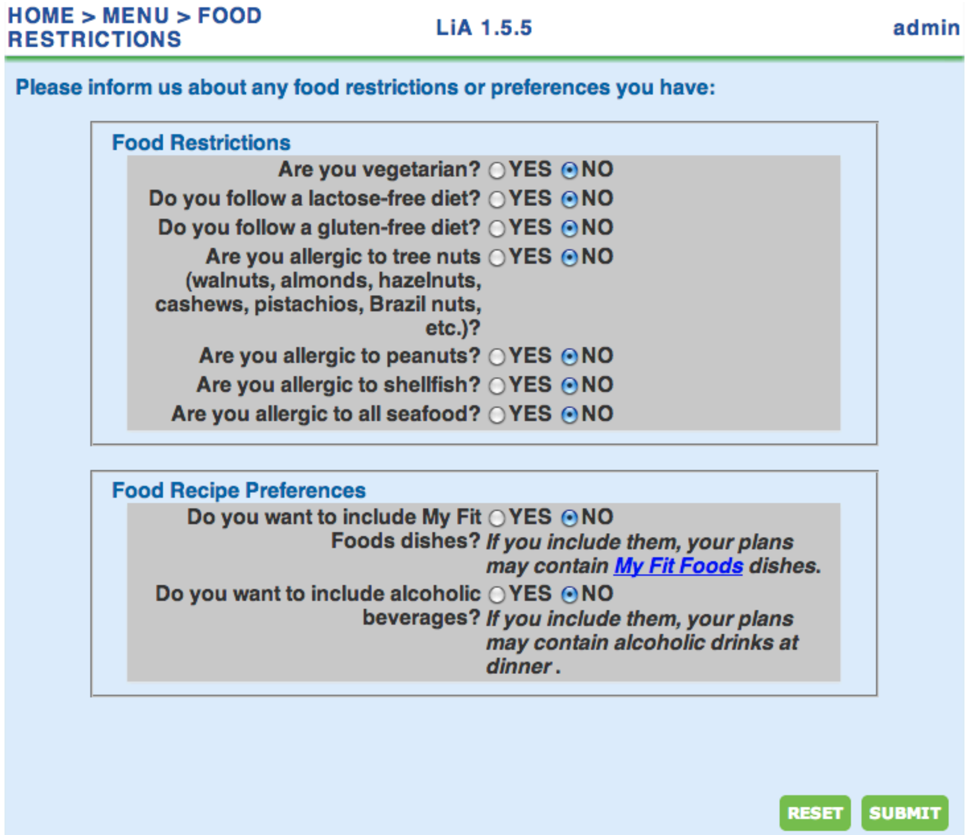


Figure 5-3: The food restrictions webpage for the online application LiA.

5.2 Concluding Remarks

The number of people with diabetes and obesity has risen to alarming rates. In this thesis, we have designed and implemented a system that uses optimization and analytics for personalized diabetes management. Through this work, we aspire to improve the lives of millions of people.

In each chapter, we described a different component of the system, each with its own contribution. In Chapter 2, we presented an algorithmic approach to learning preferences. Our approach made an important connection between observations of human behavior and general preference learning, as well as made several methodological contributions. In Chapter 3, we described models for blood glucose dynamics and an optimization approach to personalizing these models. This allows us to quantitatively minimize blood glucose levels, regardless of what food the user eats. In Chapter 4, we described optimization formulations for finding a

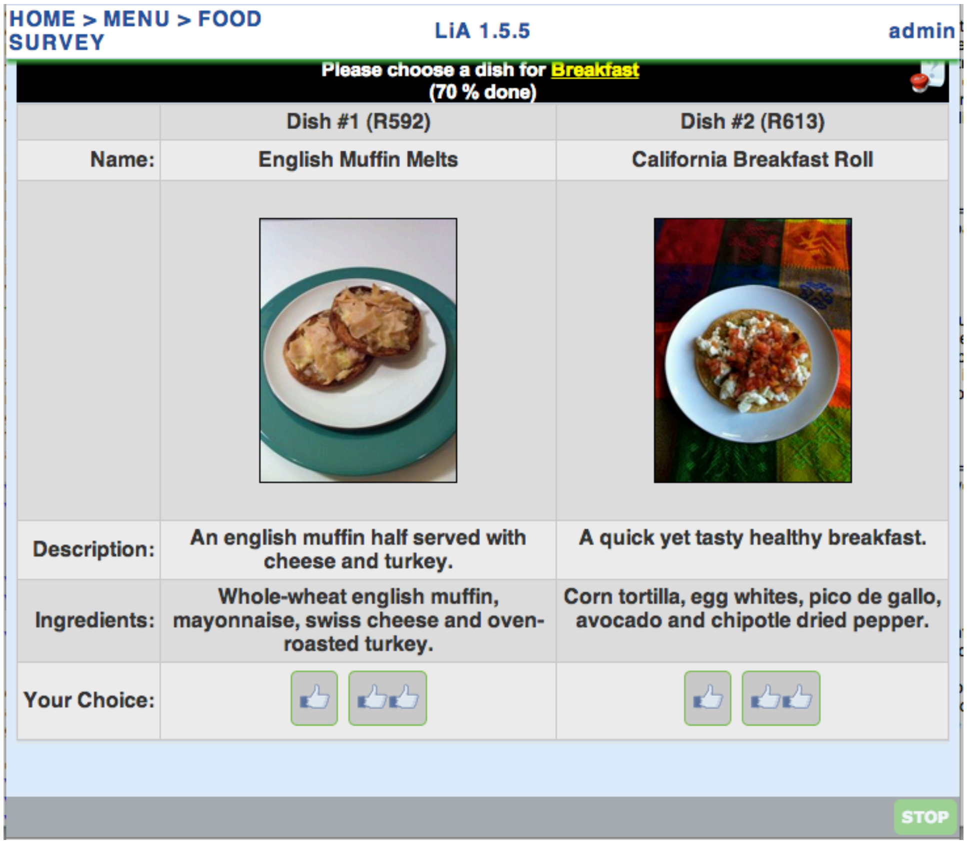


Figure 5-4: An example survey question for the online application LiA.

personalized meal and exercise plan for a diabetic patient. These formulations have the ability to turn general nutritional and lifestyle advice for diabetic patients into personalized tools for each individual patient.

The main contribution of this work is putting all of these pieces together into a personalized system that can be used by patients and healthcare providers. Given the importance of preventing and treating diabetes and obesity, we feel that this system has the potential to greatly improve diabetes management by making it more personal and dynamic.

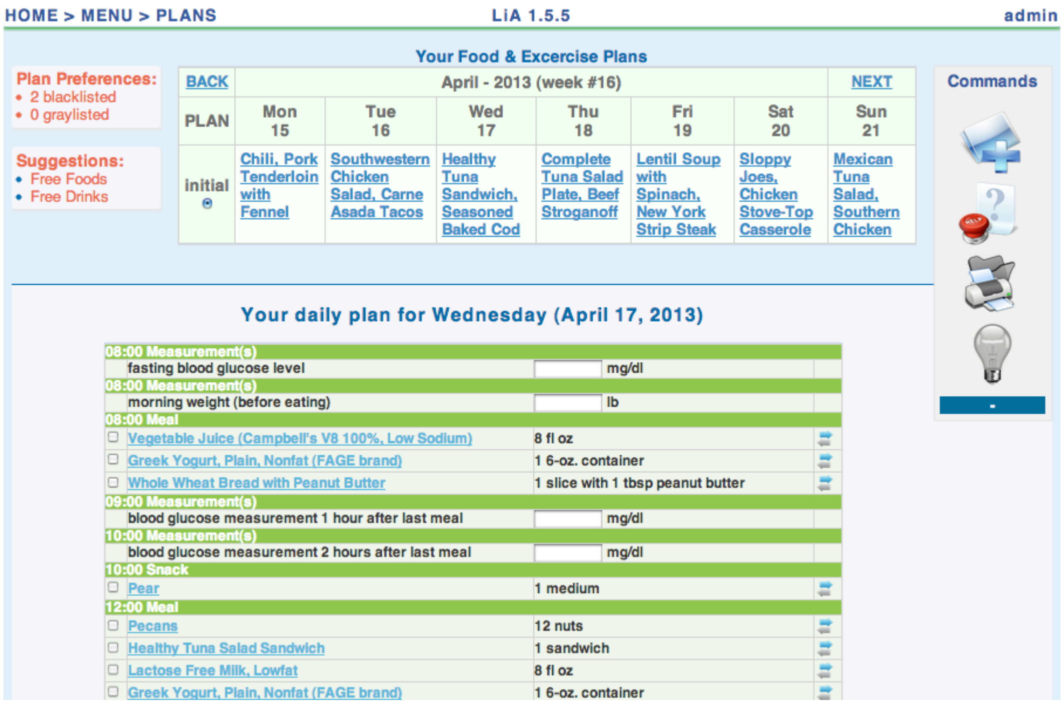


Figure 5-5: The meal planning page for the online application LiA.

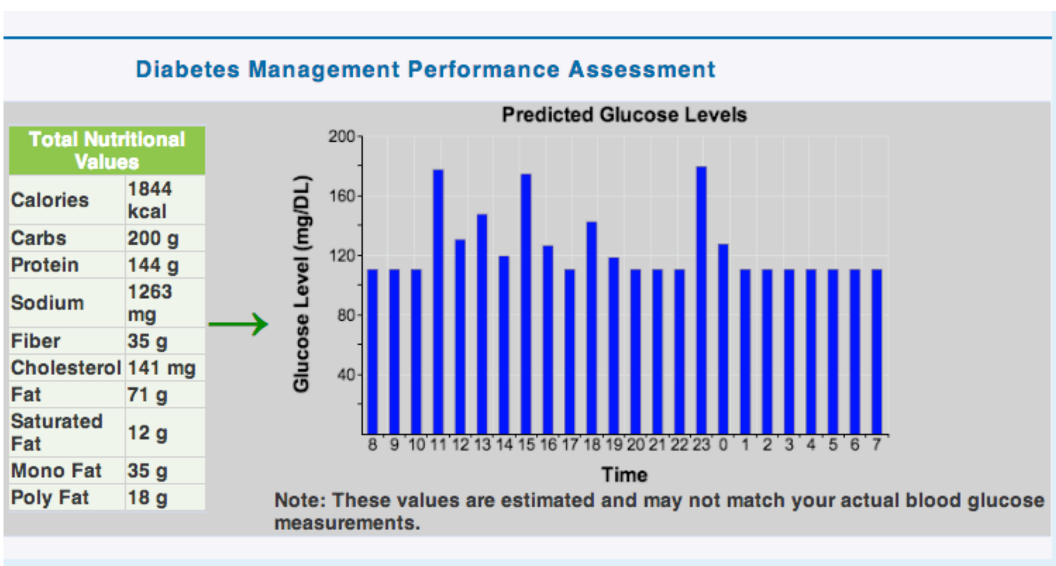


Figure 5-6: The nutritional and blood glucose information displayed in the online application LiA.

Bibliography

[1] Jacob Abernethy, Theodoros Evgeniou, Olivier Toubia, and Jean-Philippe Vert. Eliciting consumer preferences using robust adaptive choice questionnaires. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):145–155, 2008.

[2] American Diabetes Association. `http://www.diabetes.org/`, 2013.

[3] American Diabetes Association: Create your plate. `http://www.diabetes.org/food-and-fitness/food/planning-meals/create-your-plate/`, 2013.

[4] BE Ainsworth, WL Haskell, SD Herrmann, N Meckes, Jr DR Bassett, C Tudor-Locke, J Vezina JL Greer, MC Whitt-Glover, and AS Leon. 2011 compendium of physical activities: a second update of codes and met values. *Medicine and Science in Sports and Exercise*, 43(8):1575–1581, 2011.

[5] Maurice Allais. Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école américaine. *Econometrica*, 21(4):503–546, 1953.

[6] American Diabetes Association. *The New Family Cookbook for People with Diabetes*. American Diabetes Association, 2007.

[7] American Diabetes Association. *The American Diabetes Association Month of Meals Diabetes Meal Planner*. American Diabetes Association, 2010.

[8] American Diabetes Association. Executive summary: Standards of medical care in diabetes - 2013. *Diabetes Care*, 36(Suppl 1):S4–10, 2013.

[9] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[10] Fiona S. Atkinson, Kaye Foster-Powell, and Jennie C. Brand-Miller. International tables of glycemic index and glycemic load values: 2008. *Diabetes Care*, 31(12):2281–2283, 2008.

[11] L.J. Bassi. The diet problem revisited. *The American Economist*, 20(2):35–39, 1976.

[12] R. Bellazzi, G. Nucci, and C. Cobelli. The subcutaneous route to insulin dependent diabetes therapy. *Engineering in Medicine and Biology, IEEE*, 20(1):54–64, 2001.

[13] Daniel Bernoulli. Specimen theoriae novae de mensura sortis. *Comentarii Academiae Scientiarum Imperialis Petropolitanae*, 5:175–192, 1738. (Translation by Louise Sommer).

[14] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[15] Dimitris Bertsimas, Geoffrey J. Lauprete, and Alexander Samarov. Shortfall as a risk measure: properties, optimization and applications. *Journal of Economic Dynamics and Control*, 28(7):1353 – 1381, 2004.

[16] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[17] Dimitris Bertsimas and Aurelie Thiele. *Robust and Data-Driven Optimization: Modern Decision-Making Under Uncertainty*, chapter 4, pages 195–122. INFORMS, 2006.

[18] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Nashua, NH, USA, 1997.

[19] Han Bleichrodt, Jose Luis Pinto, and Peter P. Wakker. Making descriptive use of prospect theory to improve the prescriptive use of expected utility. *Management Science*, 47(11):1498–1514, 2001.

[20] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[21] James P. Boyle, Amanda A. Honeycutt, K.M. Venkat Narayan, Thomas J. Hoerger, Linda S. Geiss, Hong Chen, and Theodore J. Thompson. Projection of diabetes burden through 2050: Impact of changing demography and disease prevalence in the u.s. *Diabetes Care*, 24(11):1936–1940, 2001.

[22] J. Brand-Miller, S. Hayne, P. Petocz, and S. Colagiuri. Low-glycemic index diets in the management of diabetes: a meta-analysis of randomized controlled trials. *Diabetes Care*, 26(8):2261–7, 2003.

[23] Douglas Carroll and Paul Green. Psychometric methods in marketing research: Part i, conjoint analysis. *Journal of Marketing Research*, 32:385–391, 1995.

[24] Number of americans with diabetes projected to double or triple by 2050. `http://www.cdc.gov/media/pressrel/2010/r101022.html`, 2010. CDC Press Release.

[25] Mark E. Daly, Catherine Vale, Mark Walker, Alison Littlefield, K. George Alberti, and John C. Mathers. Acute effects on insulin sensitivity and diurnal metabolic profiles of a high-sucrose compared with a high-starch diet. *American Society for Clinical Nutrition*, 67:1186–1196, 1998.

[26] Michael L. Dansinger, Joi Augustin Gleason, John L. Griffith, Harry P. Selker, and Ernst J. Schaefer. Comparison of the atkins, ornish, weight watchers, and zone diets for weight loss and heart disease risk reduction. *JAMA*, 293(1):43–53, 2005.

[27] George B. Dantzig. The diet problem. *Interfaces*, 20(4):43–47, 1990.

[28] K. Deb and H. Gupta. Searching for robust pareto-optimal solutions in multi-objective optimization. *Proceedings of the Third Evolutionary Multi-Criterion Optimization (EMO-03)*, 2005.

[29] Alan M. Delamater. Improving patient adherence. *Clinical Diabetes*, 24(2):71–77, 2006.

[30] Diabetic recipes. `http://www.diabeticlivingonline.com/diabetic-recipes/`, 2013.

[31] Jon Doyle. Prospects for preferences. *Computational Intelligence*, pages 111–136, 2004.

[32] Pinky Dua, Francis J. Doyle III, and Efstratios N. Pistikopoulos. Multi-objective blood glucose control for type 1 diabetes. *Medical and Biological Engineering and Computing*, 47(3):343–352, 2009.

[33] James S. Dyer, Peter C. Fishburn, Ralph E. Steuer, Jyrki Wallenius, and Stanley Zionts. Multiple criteria decision making, multiattribute utility theory: The next ten years. *Management Science*, 38(5):645–654, 1992.

[34] Paul Embrechts, Claudia Kluppelberg, and Thomas Mikosch. *Modelling Extremal Events for Insurance and Finance*. Springer, New York, 1997.

[35] Vivek F Farias, Srikanth Jagabathula, and Devavrat Shah. A non-parametric approach to modeling choice with limited data. Management Science, forthcoming, 2012.

[36] Sylvia Franc, Dured Dardari, Caroline Peschard, Jean-Pierre Riveline, Magdalena Biedzinski, Beatrix Boucherie, Catherine Petit, Elisabeth Requeda, Frederic Mistretta, Michel Varroud-Vial, and Guillaume Charpentier. Can postprandial blood glucose excursion be predicted in type 2 diabetes? *Diabetes Care*, 33(9):1913–1918, 2010.

[37] Johannes Furnkranz and Eyke Hullermeier. *Preference Learning*. Springer, New York, 2011.

[38] Fabiola Demps Gains and Roniece Weaver. *The New Soul Food Cookbook For People With Diabetes, 2nd Edition*. American Diabetes Association, 2006.

[39] Susan Garner Garille and Saul I. Gass. Stigler's diet problem revisited. *Operations Research*, 49(1):1–13, 2001.

[40] R. Giacco, F. Brighenti, M. Parillo, M. Capuano, A. V. Ciardullo, A. Rivieccio, A. A. Rivellese, and G. Riccardi. Characteristics of some wheat-based foods of the italian diet in relation to their influence on postprandial glucose metabolism in patients with type 2 diabetes. *British Journal of Nutrition*, 85:33–40, 2001.

[41] Barry H. Ginsberg. Factors affecting blood glucose monitoring: Sources of errors in measurement. *J Diabetes Sci Technol.*, 3(4):903–913, 2009.

[42] Paul Green and V. Srinivasan. Conjoint analysis in consumer research: issues and outlook. *Consumer Research*, 5(2):103–123, 1978.

[43] Paul Green and V. Srinivasan. Conjoint analysis in marketing: new developments with implications for research and practice. *Journal of Marketing*, 54(4):3–19, 1990.

[44] Larry Hardesty. Improving recommendation systems. `http://web.mit.edu/newsoffice/2011/compare-recommendation-systems-0708.html`, 2011.

[45] Nancy S. Hughes. *The 4-ingredient Diabetes Cookbook: Simple, Quick and Delicious Recipes Using Just Four Ingredients or Less!* American Diabetes Association, 2007.

[46] Eyke Hullermeier, Johannes Furnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172:1897–1916, 2008.

[47] David J. A. Jenkins, Thomas M. S. Wolever, Rodney H. Taylor, Helen Barker, Hashmein Fielden, Janet M. Baldwin, Allen C. Bowling, Hillary C. Newman, Alexandra L. Jenkins, and David V. Goff. Glycemic index of foods: a physiological basis for carbohydrate exchange. *The American Journal of Clinical Nutrition*, 34:362–366, 1981.

[48] Joslin Diabetes Center: Know your food groups with diabetes. `http://www.joslin.org/info/know\_your\_food\_groups\_with\_diabetes.html`, 2013.

[49] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

[50] Adam Krzemienowski. Risk preference modeling with conditional average: an application to portfolio optimization. *Ann Oper Res*, 165:67–95, 2009.

[51] Jennifer Bucko Lamplough and Lara Rondinelli-Hamilton. *The Healthy Carb Diabetes Cookbook: Favorite Foods to Fit Your Meal Plan.* American Diabetes Association, 2008.

[52] Lilly M. Lancaster. The history of the application of mathematical programming to menu planning. *European Journal of Operational Research*, 57(3):339–347, 1992.

[53] Lean cuisine. `http://www.leancuisine.com`, 2013.

[54] C.S. Lee, M.H. Wang, and H. Hagras. A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation. *IEEE Transactions on Fuzzy Systems*, 18(2):374–395, 2010.

[55] László Lovász and Santosh Vempala. Hit-and-run from a corner. *SIAM Journal on Computing*, 35(4):985–1005, 2006.

[56] Diabetes meal plan recipes. `http://www.mayoclinic.com/health/diabetes-recipes/RE00091`, 2012.

[57] Kaisa Miettinen, Francisco Ruiz, and Andrzej P. Wierzbicki. Multiobjective optimization. chapter Introduction to Multiobjective Optimization: Interactive Approaches, pages 27–57. Springer-Verlag, Berlin, Heidelberg, 2008.

[58] Mark D. Mifflin, Sachiko T. St Jeor, Lisa A. Hill, Barbara J. Scott, Sandra A. Daugherty, and Young O. Koh. A new predictive equation for resting energy expenditure in healthy individuals. *The American Journal of Clinical Nutrition*, 51:241–247, 1990.

[59] John M. Mulvey, Robert J. Vanderbei, and Stavros A. Zenios. Robust optimization of large-scale systems. *Operations Research*, 43(2):264–281, 1995.

[60] My fit foods. `https://www.myfitfoods.com/`, 2013.

[61] Jackie Newgent. *The All-Natural Diabetes Cookbook: The Whole Food Approach to Great Taste and Healthy Eating*. American Diabetes Association, 2007.

[62] Wlodzimierz Ogryczak. Stochastic dominance relation and linear risk measures. In A.M.J. Skulimowski, editor, *Financial Modeling - Proceedings of the 23rd Meeting of the EURO Working Group on Financial Modeling*, pages 191–212. Progress and Business Publishers, 1998.

[63] Wlodzimierz Ogryczak. Tail mean and related robust solution concepts. *International Journal of Systems Science*, 2012. DOI:10.1080/00207721.2012.669869.

[64] Derek A. Paice. *Diabetes and Diet: A Type 2 Patient's Successful Efforts at Control*. Paice & Associates Inc., 1997.

[65] Karma L. Pearce, Manny Noakes, Jennifer Keogh, and Peter M. Clifton. Effect of carbohydrate distribution on postprandial glucose peaks with the use of continuous glucose monitoring in type 2 diabetes. *Am J Clin Nutr*, 87:638–644, 2008.

[66] T. Peña, P. Lara, and C. Castrodeza. Multiobjective stochastic programming for feed formulation. *Journal of the Operational Research Society*, 60(12):1738–1748, 2009.

[67] Salwa W. Rizkalla, Laika Taghrid, Muriel Laromiguiere, Dorothee Huet, Josette Boillot, Aude Rigoir, Fabienne Elgrably, and Gerard Slama. Improved plasma glucose control, whole-body glucose utilization, and lipid profile on a low-glycemic index diet in type 2 diabetic men. *Diabetes Care*, 27(8):1866–1872, 2004.

[68] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.

[69] B. Roy. A missing link in operational research decision aiding: Robustness analysis. *Foundations of Computing and Decision Sciences*, 23:141–160, 1998.

[70] M.B. Schulze, S. Liu, E. B. Rimm, J. E. Manson, W. C. Willett, and F. B. Hu. Glycemic index, glycemic load, and dietary fiber intake and incidence of type 2 diabetes in younger and middle-aged women. *Am J Clin Nutr*, 80(2):348–356, 2004.

[71] R. L. Smith. Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6):1296–1308, 1984.

[72] Sue Spitler, Linda Eugene, and Linda Yoakam, editors. *1,001 Delicious Recipes for People with Diabetes, Second Edition*. Surrey Books, Agate Publishing, Inc., 2007.

[73] G. Stigler. The cost of subsistence. *Journal of Farm Economics*, 27:303–314, 1945.

[74] Diabetic recipes. `http://www.tasteofhome.com/Recipes/Healthy-Eating/Diabetic-Recipes`, 2013.

[75] Sharon V. Thompson, Donna M. Winham, and Andrea M. Hutchins. Bean and rice meals reduce postprandial glycemic response in adults with type 2 diabetes: a cross-over study. *Nutrition Journal*, 11(23), 2012.

[76] Olivier Toubia, John R. Hauser, and Rosanna Garcia. Probabilistic polyhedral methods for adaptive choice-based conjoint analysis: Theory and application. *Marketing Science*, 26(5):596–610, 2007.

[77] Olivier Toubia, John R. Hauser, and Duncan I. Simester. Polyhedral methods for adaptive choice-based conjoint analysis. *Journal of Marketing Research*, 41(1):116–131, 2004.

[78] Olivier Toubia, John R. Hauser, Duncan I. Simester, and Ely Dahan. Fast polyhedral conjoint estimation. *Marketing Science*, 22(3):274–303, 2003.

[79] Amos Tversky and Daniel Kahneman. Loss aversion in riskless choice: A reference-dependent model. *Quarterly Journal of Economics*, 106(4):1039–1061, 1991.

[80] USDA National Nutrient Database for Standard Reference, Release 24, U.S. Department of Agriculture, Agricultural Research Service. Nutrient Data Laboratory Home Page, `http://www.ars.usda.gov/nutrientdata`, 2011.

[81] Santosh Vempala. Geometric random walks: A survey. *Computational Geometry*, 52:573–612, 2005.

[82] John von Neumann and Oscar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[83] Abraham Wald. Statistical decision functions which minimize the maximum risk. *The Annals of Mathematics*, 46:265–280, 1945.

[84] Jyrki Wallenius, James S. Dyer, Peter C. Fishburn, Ralph E. Steuer, Stanley Zionts, and Kalyanmoy Deb. Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead. *Management Science*, 54(7):1336–1349, 2008.

[85] J. G. Wang and S. Zionts. The aspiration level interactive method (aim) reconsidered: Robustness of solutions to multiple criteria problems. *EJOR*, 17(2):948–958, 2006.

[86] Robyn Webb. *Diabetic Meals in 30 minutes – or less!* American Diabetes Association, 2006.

[87] Robyn Webb. *The Diabetes Comfort Food Cookbook*. American Diabetes Association, 2011.

[88] Stanley Zionts and Jyrki Wallenius. An interactive programming method for solving the multiple criteria problem. *Management Science*, 22(6):652–663, 1976.