A Tractable Approach To Inverse Optimization Under Euclidean Norm

by

Sara Ebrahimkhani

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The conventional optimization assumes that the problem and its parameters are known, and it utilizes this information to determine the optimal solution. Inverse optimization works in reverse by determining different parameters of an optimization model such that a given dataset of observed decisions from the past becomes optimal for the model. The parameters imputed through inverse optimization can be in the objective function and/or the constraints of the model. When inferring the constraint parameters, the choice of objective for the inverse optimization problem can result in different inverse optimal solutions. However, it is unclear which solution provides the best fit to the data. In this study, a goodness-of-fit measure is first introduced to evaluate the fit between the model and data and determine the quality of the inferred feasible region based on the distances of data points from its boundary. Next, employing this measure as the objective function, a multi-point inverse optimization problem under the Euclidean norm is proposed to infer the feasible region of a linear optimization model. Given the nonlinear nature of the Euclidean norm, a relaxation technique using the non-smooth L_1 penalty function is proposed for the inverse optimization problem. This reformulates the non-convex mixed-integer quadratically-constrained programming problem into a mixed-integer quadratic programming problem which is more tractable. Then, an exact heuristic method and a greedy heuristic method are introduced to alleviate the computational challenges of the problem. Finally, two application examples to illustrate the practicality and effectiveness of our proposed model and solution approach are presented. In the first application, our model determines the implicit criteria based on which a patient is identified as an outpatient without requiring hospital supervision. The second application focuses on improving the recommended diets by uncovering hidden preferences and suggesting meal plans based on individuals' past food choices.

Acknowledgements

I would like to express my special thanks to my supervisors, Professor Houra Mahmoudzadeh and Professor Hossein Abouee Mehrizi, for their invaluable guidance, support, and constant encouragement during my graduate studies.

I would like to thank my committee members, Professor Samir Elhedhli and Professor Saeed Ghadimi for taking the time to read my thesis and giving me valuable feedback to improve this work.

I would like to extend my thanks to my family for their unconditional love, support, and understanding, as well as to my friends who have kindly helped me in many different ways.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANOVA Analysis of variance 36

FO Forward optimization 1, 2, 4, 5, 7, 8, 12–14, 16, 18, 48, 50, 52

IO Inverse optimization 1–17, 21–23, 25, 27, 30, 32–34, 38, 39, 41–43, 46, 48, 52, 53

KKT Karush–Kuhn–Tucker 6, 7

LHS Left-hand side 2, 5, 8, 12

MIQCP Mixed-integer quadratically constrained programming 27, 52

MIQP Mixed-integer quadratic programming 28, 52

MSE Mean squared error 19

RHS Right-hand side 2, 5, 8, 12

Chapter 1

Introduction

In recent years, due to the availability of data, Inverse optimization (IO) has become a popular topic. Researchers have been able to explore and apply IO techniques in various domains, such as healthcare, transportation, power systems, and finances and economics (Chan et al., 2021b). This growing trend is due to the fact that IO can uncover valuable insights and improve decision-making processes by inferring the unknown parameters of an optimization problem. By analyzing data available from past decisions, IO allows researchers to build models that can be used to generate meaningful solutions for future decisions.

Unlike conventional optimization which determines the optimal decision of a perfectly-known optimization model, IO works reversely. It observes the decision(s) and imputes the objective or constraint parameters of the underlying optimization model such that each observed decision becomes optimal or near-optimal for the model. The conventional optimization problem is often called Forward optimization (FO) problem, and hence, IO determines the unknown parameters of the forward problem given a set of observed solutions

(Ahuja and Orlin, 2001). In the literature, certain parameters of the forward problem such as the objective coefficients, Left-hand side (LHS), and/or Right-hand side (RHS) vectors, are often inferred by solving an IO problem. In this sense, IO aims to fit an optimization model to some observed data that represent past solutions to the problem.

In the literature, for IO problems that impute the objective function, a goodness-of-fit measure is introduced to assess the fit between the inferred model and observed decisions. This metric measures the error in fit in terms of the projected distance of the decision points to the constraints of the model. However, such a measure cannot be used to evaluate the fit between the model and data when inferring the constraints of a FO problem. The literature offers certain linear measures of quality for problems that infer constraints, but these measures cannot be generally used to evaluate the fit between feasible regions.

This thesis makes the following specific contributions:

- First, we introduce a goodness-of-fit metric based on the Euclidean distance that holds three key properties: (1) it provides the feasible region with minimum Euclidean distance of observations from its boundary, (2) the metric improves when constraints are added to the feasible region, and (3) it yields a feasible region with a better Euclidean fit compared to other existing linear measures. The metric is also insensitive to redundant constraints and it changes monotonically as the number of constraints changes.

- Second, we present a multi-point IO model that employs the proposed goodness-of-fit measure as its objective function. Our aim is to determine the feasible region of a linear FO problem with maximum Euclidean fit.

- Third, we propose a tractable formulation for the inverse model to alleviate the complexity of the resulting non-linear model by implementing an equivalent simpler reformulation by L_1 penalty function method, a heuristic algorithm to find the exact solution, and a greedy heuristic algorithm to find a high-quality approximate solution efficiently.

Finally, we apply the proposed methodology to two applications in the healthcare sector, patient classification and diet planning problems. First, we infer the underlying criteria used for determining the severity of a patient's illness which affects the decision of whether the patient should be admitted (inpatient) or sent home shortly (outpatient). The proposed model finds linear criteria based on which outpatients are identified using laboratory report data from hospitalized patients. Second, by utilizing a dieter's past food choices, we use IO to find an individualized meal plan that not only meets their dietary needs but also takes into account their food preferences.

The structure of the remainder of this thesis is as follows. In Chapter 2, we present different variations of IO and review the relevant literature in each section. In Chapter 3, we propose the goodness-of-fit measure, develop the IO formulation, and present the solution methods to solve the problem. In Chapter 4, we perform different tests on the model and data to evaluate the performance of the inverse model and solution methods. This thesis concludes with highlights of findings and future research directions in Chapter 5.

Chapter 2

Literature Review

IO has recently gained significant attention and popularity in both theoretical research and practical applications (Chan et al., 2021b). There are many different variations of IO problems in the literature. These variations arise from the type of inferred parameters, the structure of the FO problem, data characteristics, and the objective function of the IO problem. In this chapter, each of these variations is further explained and a review of the relevant literature is provided.

2.1 Parameters inferred in inverse optimization

The parameters that are imputed by IO can be in the objective function and/or constraints. The first paper on IO goes back to the 1990s when Zhang et al. (1995) proposed an inverse shortest path problem to find the weight of arcs in the objective function such that the deviations from the estimated values are as small as possible. Zhang and Liu (1996) applied the optimality conditions on inverse minimum cost flow and assignment problems to perturb the cost associated with decision variables so as to make the observed solution optimal. Ahuja and Orlin (2001) developed a general IO framework to estimate the objective coefficients of a linear optimization model. In the literature, there have been studies that focused on imputing the objective function of a single-objective FO problem (Keshavarz et al., 2011; Tavaslıoğlu et al., 2018; Aswani et al., 2018; Ghate, 2020b; Shahmoradi and Lee, 2022b), as well as other studies that concentrated on determining the objective weights for a multi-objective FO problem (Sayre and Ruan, 2014; Chan et al., 2014; Boutilier et al., 2015; Goli et al., 2018; Chan and Lee, 2018; Gebken and Peitz, 2021; Babier et al., 2021).

While the literature on IO has mostly focused on objective inference, a few papers have focused on finding the RHS of the constraints alongside the objective. Dempe and Lohse (2006) estimated the objective function and RHS by developing a model that minimizes the distance between the observed point and the optimal solution. Chow and Recker (2012) estimated the objective coefficients and RHS of the household activity pattern problem such that the perturbation of the parameter from the initial belief is minimized under the L_1 norm. Černỳ and Hladík (2016) studied inverse linear programming with intervals in the RHS and/or the objective function. Saez-Gallego and Morales (2017) found the RHS and objective parameters of a linear forward problem given a time series of decisions.

Extending from inferring the RHS only, Chan and Kaw (2020) proposed two IO models in which the LHS parameters are imputed in addition to the objective coefficients based on a single observed optimal solution. Most closely related to our work, Ghobadi and Mahmoudzadeh (2021) and Mahmoudzadeh and Ghobadi (2022) proposed IO methods for inferring complete constraint parameters, including both RHS and LHS, in a linear FO problem with a known objective and multiple observations.

IO models for constraints inference are structurally and mathematically different from those that only infer the objective. When imputing the objective parameters, the inferred objective needs to maintain the optimality of the observed optimal point. To achieve this, the relevant inverse problem often enforces dual feasibility and strong duality constraints, ensuring that the forward problem has an optimal solution. Ahuja and Orlin (2001) employed dual feasibility and complementary slackness optimality conditions to develop the IO problem. Chan et al. (2019) addressed this problem using strong duality conditions instead of complementary slackness. On the other hand, when inferring constraint parameters, it is essential to consider primal feasibility in addition to dual feasibility and strong duality constraints. This is necessary to ensure that the inferred constraints maintain feasibility for all observations (Ghobadi and Mahmoudzadeh, 2021). As a result, the IO for constraints inference becomes nonlinear and therefore more complex, making it relatively less studied in the literature.

2.2 Structure of the forward problem

The structure of the forward problem is a key factor in the tractability of the IO problem. It impacts the complexity of the IO and the choice of mathematical techniques that can be used to solve the inverse problem (Chan et al., 2021b). The literature has considered the structure of the forward model to be linear, conic, convex, or discrete (Chan et al., 2021b). In a linear forward problem, both the objective and the constraints are considered to be linear. This type of problem has been addressed in many papers. Ahuja and Orlin (2001), Chan et al. (2019), Ghate (2020b), Babier et al. (2021), and Shahmoradi and Lee (2022b) solved the problem of finding the objective of a linear optimization problem. There are also studies in which some of the constraint parameters of the linear forward problem are imputed (Güler and Hamacher, 2010; Černỳ and Hladík, 2016; Saez-Gallego and Morales, 2017; Chan and Kaw, 2020; Ghobadi and Mahmoudzadeh, 2021).

When the forward problem is not linear, ensuring the optimality of the observations becomes more complex. When the model belongs to specific classes such as convex or conic optimization, the Karush–Kuhn–Tucker (KKT) conditions serve as sufficient optimality condition (Mahmoudzadeh and Ghobadi, 2022). In conic and convex models, any combination of linear, second-order, and quadratic objective being optimized over a linear, second-order, or semidefinite cone is possible. In this class, Iyengar and Kang (2005) presented the IO of a conic program. They used KKT conditions as sufficient conditions for the optimality of the given observation. Zhang and Zhang (2010) proposed an inverse conic programming problem with a quadratic objective and linear constraints. Their model minimizes the perturbation of the objective function such that the solution becomes optimal. Keshavarz et al. (2011) and Aswani et al. (2018) applied KKT conditions to find the objective coefficient of a convex optimization problem. Zhang et al. (2010) and Zhang et al. (2015) addressed the inverse quadratic and second-order cone quadratic programming problems, respectively.

The discrete models refer to (mixed) integer linear programs. The optimality of discrete FO problems cannot be easily represented by a small number of equations, such as the KKT conditions (Chan et al., 2021b). Using the strong duality and dual feasibility conditions, similar to the linear forward problem, will lead to an IO problem with a large number of variables and constraints (Schaefer, 2009). An alternative approach for this type of forward model involves iterative algorithms that employ cutting planes to enhance the accuracy of the approximate solution for the inverse model (Wang, 2009). Bulut and Ralphs (2021), Lamperski and Schaefer (2015), Moghaddass and Terekhov (2021), and Bodur et al. (2022) studied the discrete forward problem where the cost vector is unknown.

2.3 Data characteristics

The input of the IO problem can be one or multiple observations that are optimal, near-optimal feasible or infeasible. In single-point IO, one observed decision is available, while in multi-point IO, there is a finite number of observations Ghobadi and Mahmoudzadeh (2021). In objective inference, the single-point IO problems often include one feasible observed decision that can be optimal (Zhang and Liu, 1996, 1999; Iyengar and Kang, 2005; Zhang et al., 2010; Ghate, 2020b) or near-optimal (Aswani et al., 2018; Chan et al., 2019). Chan et al. (2014) and Naghavi et al. (2019) also considered a single-point problem, but with the recognition that the observation may be infeasible. The case of multiple near-optimal observed decisions has also gained attention in objective inference (Keshavarz et al., 2011; Bertsimas et al., 2015; Ajayi et al., 2022). In recent years, some studies considered the possibility of infeasible observations that are made near-optimal (Mohajerin Esfahani et al., 2018; Ahmadi et al., 2020a). Shahmoradi and Lee (2022b) considered sub-optimal or infeasible observations to find the cost function of a linear optimization model. Babier et al. (2021) proposed an IO model that infers the objective function with an ensemble of decisions that may be feasible or infeasible.

Dempe and Lohse (2006) and Chow and Recker (2012) addressed IO problems that impute cost vector and RHS based on a single feasible observation where the sub-optimality gap was taken into account. Chan and Kaw (2020) and Güler and Hamacher (2010) also considered the sub-optimality gap in inverse problems to infer cost vector and LHS, and RHS, respectively, given a single observed decision. In case of inferring the full constraints parameters, Ghobadi and Mahmoudzadeh (2021) considered multiple feasible observations. Mahmoudzadeh and Ghobadi (2022) proposed an IO method, which incorporates both feasible and infeasible observations.

2.4 Objective of the inverse optimization problem

In classical IO, a measure of the error in the fit between the model and data is minimized (Chan et al., 2019). When inferring the objective, the optimality condition is considered as a measure of the error. If the optimally conditions are satisfied, the error in the fit will become zero. In the linear FO problem, the model is a good fit to the decision points if all points are on the boundaries of the feasible region. On the other hand, if some of the observed decisions are interior points, then the error in the fit will get a positive value. Based on the choice of the objective function and the norm, the types of the error and subsequently, the structure of the IO problem can be different (Chan et al., 2019). In the literature on objective inference, there exist three distinct variations of the inverse models based on the objective function of the inverse model: (1) the p-norm error, that is measured in terms of the distance between the observed decision and the inferred optimal decision, (2) absolute duality gap, in which error is measured in terms of the objective function value of the observed decision and the inferred optimal decision, and (3) relative duality gap, in which the error is measured in terms of the competitive ratio of the objective function value of the observed decision and the inferred optimal decision. Güler and Hamacher (2010) investigated the minimum cost flow problem with the unknown arc capacities. Given a feasible flow, the capacities are estimated so that the solution becomes near-optimal under L_1 norm and L_∞ norm. Chan et al. (2014) suggested two inverse models that sought to minimize the sub-optimality measures, absolute and relative duality gaps. Moghaddass and Terekhov (2021) considered the L_1 norm and L_∞ norm as measures for evaluating the error in the fit.

While the optimality conditions can be utilized as an error metric for inverse models that infer the objective, the sub-optimality measures cannot be employed as the objective function when inferring the constraints. The rationale behind this is that, since the objective is known, the optimal point can be identified among the observations, and it is always possible to ensure that the optimality condition is satisfied by adjusting the constraints. This is because at least one of the inferred constraints, which is orthogonal to the cost vector, can pass through the optimal point, resulting in an optimality gap of zero for the best observed solution. Ghobadi and Mahmoudzadeh (2021) studied several loss functions for the IO problem that infers constraints. They presented two scenarios: one involving a prior belief regarding the constraint parameter, and another in which no prior belief is present. In the first scenario, the objective is to minimize the distance between the imputed parameters and a prior belief on what the constraints should look like. In the second case, the focus is on inferring the constraint parameters in a manner that minimizes different linear measures of distance between the observations and the constraints.

2.5 Applications of inverse optimization

IO has been employed in a wide range of domains, demonstrating its versatility and diverse applications. In every domain, a dataset of past decisions is observed, and the decision-making model that most accurately reflects those decisions is determined. Some examples of well-studied applications are listed below:

- **Healthcare**: IO has attracted interest within healthcare systems and treatment planning. Chan et al. (2021a) imputed the arc costs in a clinical pathway problem such that the reference pathways represent optimal solutions to an optimization problem formulated on this network. Considerable research has been undertaken to explore the application of IO in the field of radiation therapy treatment planning

(Chan et al., 2014; Boutilier et al., 2015; Babier et al., 2018; Chan and Lee, 2018; Ajayi et al., 2022; Ghate, 2020a; Mahmoudzadeh and Ghobadi, 2022). Diet recommendation represents another domain where IO finds application (Ghobadi et al., 2018; Ahmadi et al., 2020a; Shahmoradi and Lee, 2022b,a). In (Ghobadi and Mahmoudzadeh, 2021), the dietary choices are treated as observations, and through the utilization of IO, the implicit set of constraints that led to these choices is inferred.

- **Transportation**: Transportation systems utilize the observed decisions to deduce unknown parameters and improve the overall efficiency of the system. In this particular domain, the applications of IO include inferring parameters in transportation problem (Xu et al., 2018), vehicle routing problem (Chen et al., 2021), household activity pattern problem (Chow and Recker, 2012), air transportation and scheduling problem (Wei and Vaze, 2018), and freight transshipment assignment problem (Chow et al., 2014).

- **Power systems**: IO has been studied in the areas of the electricity market and power systems. Birge et al. (2017) applied IO on the electricity market to impute the energy price, as a constraint parameter. There are also several studies that presented an inverse model to estimate the price-response of electricity (Saez-Gallego and Morales, 2017; Lu et al., 2018; Fernández-Blanco et al., 2019, 2021).

- **Finance and economics**: In the field of finance, IO has been applied to improve decision-making processes. Wang et al. (2014) considered the portfolio rebalancing model and presented an inverse model to minimally perturb the current estimate of the expected return rate. Yu et al. (2023) has recently employed IO to derive investment risk preferences based on observed portfolios. Moreover, Iyengar and Kang (2005), Zhang and Zhang (2010), and Bertsimas et al. (2012) used IO to identify the parameters in portfolio optimization.

2.6 Contributions

The fundamental formulation for the IO problem proposed in this study is based on the work of Ghobadi and Mahmoudzadeh (2021), as we also employ IO techniques to infer the full constraint matrix of a linear problem using multiple input observations. However, our work has many additional novelties. Our key contributions are listed below:

- Introducing a goodness-of-fit measure utilizing Euclidean distance to identify the feasible region with the better fit.
- Incorporating the goodness-of-fit measure based on the Euclidean distance into the IO problem.
- Reformulating the model using L_1 penalty function and proposing exact and greedy heuristic algorithms to solve the problem.
- Demonstrating two example applications in patient classification and diet planning.

Chapter 3

Methodology

In this chapter, we present an IO method to infer the feasible region of a linear problem. It is assumed that the full constraint matrix is to be inferred, meaning both LHS and RHS of the FO problem can be unknown. Therefore, we consider having a set of known and unknown constraints. The objective coefficients are all known in our model. The model takes multiple observations as input and the goal is to find a set of linear constraints that renders all observations feasible and the resulting feasible region has the highest possible fit to data.

We first provide an overview of IO formulation in section 3.1. Then, we introduce the goodness-of-fit measure and incorporate it into the IO model in section 3.2. Finally, in section 3.3, we simplify the model formulation and propose both exact and heuristic solution methods to alleviate the complexity of the proposed model.

3.1 Inverse optimization background

In this section, we first define the FO problem. Then, we formulate the IO problem and present potential loss functions for the inverse problem that are proposed in the literature. Note that we focus on inverse optimization for inferring constraints throughout this thesis.

3.1.1 Forward optimization problem

Let $\mathbf{A}$ and $\mathbf{b}$ be the unknown constraint parameters in the FO problem that needs to be found by IO. Let $\mathbf{G}$ and $\mathbf{h}$ be the known constraints. Also, let $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m_1 \times n}$, $\mathbf{b} \in \mathbb{R}^{m_1}$,$\mathbf{G} \in \mathbb{R}^{m_2 \times n}$, and $\mathbf{h} \in \mathbb{R}^{m_2}$. Therefore, the FO problem can be written as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}'\mathbf{x} && \text{(3.1a)} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} && \text{(3.1b)} \\ & \mathbf{G}\mathbf{x} \geq \mathbf{h} && \text{(3.1c)} \\ & \mathbf{x} \in \mathbb{R}^n && \text{(3.1d)} \end{aligned}$$

The unknown and known constraints are indexed by the sets $\mathcal{I}_1 = \{1, \ldots, m_1\}$ and $\mathcal{I}_2 = \{1, \ldots, m_2\}$, respectively. Therefore, $\mathbf{a}_i$ represents the i^{th} row of the matrix $\mathbf{A}$. The data observations are indexed by the set $\mathcal{K} = \{1, \ldots, p\}$. The set $\mathcal{J} = \{1, \ldots, n\}$ describes the indices of the variable $\mathbf{x}$ in the forward problem.

3.1.2 Inverse optimization problem

Assume that there are some feasible or optimal observations available. The observed decision k is denoted by $\hat{\mathbf{x}}^k$. Since the objective function is known, we call the observation with the best value as the preferred point and denote it by $\hat{\mathbf{x}}^0$. If the problem had multiple optimal solutions, we choose one of the points arbitrarily as the preferred point.

$$\hat{\mathbf{x}}^0 \in arg \min_{\hat{\mathbf{x}}^k, k \in \mathcal{K}} \{\mathbf{c}'\hat{\mathbf{x}}^k\} \tag{3.2}$$

Consider $\mathbf{y}$ and $\mathbf{w}$ as dual variables associated with constraints $\mathbf{Ax} \geq \mathbf{b}$ and $\mathbf{Gx} \geq \mathbf{h}$, respectively. Following the convention of Ghobadi and Mahmoudzadeh (2021), the IO which infers the whole constraint parameters of the forward problem is as follows:

$$\begin{aligned}
\min_{\mathbf{A},\mathbf{b},\mathbf{y},\mathbf{w}} \quad & 0 && \text{(3.3a)}\\
\text{s.t.} \quad & \mathbf{A}\hat{\mathbf{x}}^k \geq \mathbf{b} \quad \forall k \in \mathcal{K} && \text{(3.3b)}\\
& \mathbf{c}'\hat{\mathbf{x}}^0 = \mathbf{b}'\mathbf{y} + \mathbf{h}'\mathbf{w} && \text{(3.3c)}\\
& \mathbf{A}'\mathbf{y} + \mathbf{G}'\mathbf{w} = \mathbf{c} && \text{(3.3d)}\\
& ||\mathbf{a}_i|| = 1 \quad \forall i \in \mathcal{I}_1 && \text{(3.3e)}\\
& \mathbf{y} \in \mathbb{R}^{m_1}, \quad \mathbf{w} \in \mathbb{R}^{m_2} && \text{(3.3f)}\\
& \mathbf{A} \in \mathbb{R}^{m_1 \times n}, \quad \mathbf{b} \in \mathbb{R}^{m_1} && \text{(3.3g)}
\end{aligned}$$

Equation (3.3b) enforces primal feasibility. Equations (3.3c) and (3.3d) are strong duality and dual feasibility constraints, respectively. Finally, equation (3.3e) is the normalization constraint on the i^{th} row of matrix $\mathbf{A}$ that prevents finding trivial solutions. Here, we assume that the IO model has no objective. However, in section 3.1.3, we will discuss possible loss functions.

The IO problem (3.3) is a nonlinear program due to the multiplication of variables in the strong duality and dual feasibility constraints. Ghobadi and Mahmoudzadeh (2021) presented a tractable reformulation of the IO which infers the feasible region of the FO problem. They proved that by adding the linear half-space $\mathcal{C} = \{\hat{\mathbf{x}} \in \mathbb{R}^n | \quad \mathbf{c}'\hat{\mathbf{x}} \geq \mathbf{c}'\hat{\mathbf{x}}^0\}$

to the set of known constraints, the strong duality and dual feasibility conditions will be guaranteed. Hence, the complexity of the IO problem will be reduced. The equivalent reformulation of the inverse problem (3.3) is shown below:

$$
\begin{aligned}
\min_{\mathbf{A},\mathbf{b}} \quad & 0 && (3.4a)\\
\text{s.t.} \quad & \mathbf{a}_i'\hat{\mathbf{x}}^k \geq b_i \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K} && (3.4b)\\
& ||\mathbf{a}_i|| = 1 \quad \forall i \in \mathcal{I}_1 && (3.4c)\\
& \mathbf{A} \in \mathbb{R}^{m_1 \times n}, \quad \mathbf{b} \in \mathbb{R}^{m_1} && (3.4d)
\end{aligned}
$$

3.1.3 Loss functions

The inclusion of a loss function $\mathcal{F}$ as the objective function in the model leads to multiple optimal solutions for the IO problem (3.4). Ghobadi and Mahmoudzadeh (2021) proposed four linear measures that can be employed as potential loss functions within this context.

- **Indifference measure**: When the loss function of the IO is set to zero, the problem simplifies to a feasibility problem. Model (3.4) illustrates the IO formulation utilizing the indifference measure as the loss function.

$$\mathcal{F} = 0 \tag{3.5}$$

- **Adjacency measure**: This measure minimizes the sum of the distances of each observation from all constraints. Let d_{ik} be the distance of each observation k from constraint i. The adjacency measure is:

$$\mathcal{F} = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}_1} d_{ik} \tag{3.6}$$

- **Fairness measure**: This measure tries to find constraints such that they are equally close to all observations. Let $d_k = \sum\limits_{i \in \mathcal{I}_1} d_{ik}$ be the total distance for all observations. The fairness measure is:

$$\mathcal{F} = \sum_{k \in \mathcal{K}} (d_k - \sum_{k \in \mathcal{K}} \frac{d_k}{|\mathcal{K}|}) \tag{3.7}$$

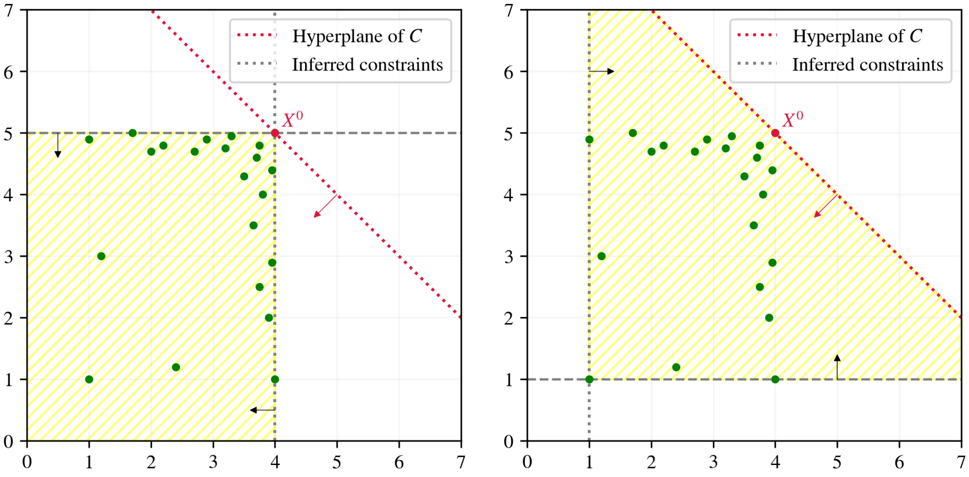
- **Compactness measure**: This measure tries to find constraints such that the minimum distance of each observation from all constraints is minimized. The compactness measure is:

$$\mathcal{F} = \sum_{k \in \mathcal{K}} \min_{i \in \mathcal{I}_1} d_{ik} \tag{3.8}$$

The utilization of different loss functions yields different optimal solutions. Although some loss functions are proposed in the literature, there is currently no method suggested to determine which solution is a better fit to data. Furthermore, so far, the literature has focused on linear measures. In the following section, we introduce a p-norm based goodness-of-fit measure to assess the fit between the model and data. Subsequently, we use it as the objective function for the IO problem.

3.2 Goodness of fit

The IO problem (3.4) focuses on finding the feasible region that contains all observations. However, it is important to acknowledge that multiple sets of constraints can lead to the feasibility of observations. Therefore, there is a need to assess and determine which set of constraints is more desirable. In other words, in addition to finding constraints for the FO model, it is important to consider additional criteria to evaluate and compare different



(a) Preferred inferred region (b) Possible inferred region

Figure 3.1: Motivational example

possible sets of constraints. For example, consider Figure (3.1), where the observed data points are depicted as green circles, the optimal point is denoted by the red circle, and the feasible region of the problem is represented by the shaded yellow area. Both figures (3.1a) and (3.1b) can potentially represent solutions to the IO problem (3.4) that infers 2 constraints. This is because the feasible region in both figures includes all data points in a manner that makes the point $\mathbf{x}^0$ optimal. However, we want to obtain Figure (3.1a) as the inferred feasible region, as it is a better fit to the data.

In this section, we define the concept of fit and propose a goodness-of-fit measure based on the Euclidean distance for constraint inference. Then, we incorporate the measure into the IO problem and provide a numerical example. We also compare the results of our proposed model with those in the literature using a numerical example.

3.2.1 Definition of fit

Before introducing the goodness-of-fit measure, we need to establish our definition of a good fit. We consider a feasible region to be a better fit to the data if the distances of observations from the nearest constraints are smaller. A smaller distance to the nearest constraint indicates that the observations are closer to the boundary of the feasible region. Hence, in an ideal situation, a perfect fit would involve a feasible region where all observations lie on the constraints. However, due to the presence of interior points, achieving this situation is not always attainable. Our goal is to introduce a goodness-of-fit measure that exhibits three main properties:

1. If a feasible region provides smaller Euclidean distances between observations and its boundary, the goodness-of-fit measure identifies it as a better fit.
2. By adding constraints, the goodness-of-fit measure can only improve and thereby the fit cannot get worsen.
3. With the same number of constraints, the goodness-of-fit measure provides a better Euclidean fit compared to other existing measures in the literature.

We next propose a measure that holds these properties and identifies the feasible region with the minimum distances between the boundary of the feasible region and all observations.

3.2.2 Goodness-of-fit measure

Having defined the concept of fit, we now introduce a goodness-of-fit measure to evaluate how well the feasible region of the FO problem fits the observations. Having multiple data points as observed decisions and different sets of constraints, the goal is to find the set of constraints that is a better fit to the data points. The proposed metric is inspired by the Mean squared error (MSE) in regression theory. Therefore, it determines the average minimum distance of each observation from all the constraints. Let ϵ_i^k be the p-norm distance of decision point k from constraint i. Recall that the p-norm is defined as $||\mathbf{x}||_p = (\sum\limits_{j\in\mathcal{J}} |x_j|^p)^{\frac{1}{p}}$ for $p \in [1, \infty)$ and $||\mathbf{x}||_\infty = \max\limits_j\{|x_j|\}$ for $p = \infty$. With this definition, the metric, denoted by ρ, is defined as follows:

$$\rho = \frac{\sum\limits_{k\in\mathcal{K}} \min\limits_{i\in\mathcal{I}_1} \ \epsilon_i^k}{|\mathcal{K}|} \tag{3.9}$$

Note that the lower the value of ρ, the better the fit will be. Intuitively, the distances between observations and the boundary of the feasible region indicate how well the model fits the data. To capture this notion, we have incorporated the minimum distance in the numerator of the metric function, allowing us to quantify the distance between observations and the nearest constraint. Additionally, to ensure the metric remains independent of the number of data points, we calculate the average of these minimum distances across all observations.

Proposition 1. *If we add a redundant constraint to the inferred feasible region, the proposed metric ρ does not change.*

Proof. Let's assume constraint $\mathbf{a}_l'\hat{\mathbf{x}}^k \geq b_l$ is redundant. Let's define $\mathcal{S}_l = \{\hat{\mathbf{x}}^k \in \mathbb{R}^n | \mathbf{a}_i'\hat{\mathbf{x}}^k \geq b_i \quad \forall i \in \mathcal{I}_1 - \{l\}\}$ and $\mathcal{S} = \{\hat{\mathbf{x}}^k \in \mathbb{R}^n | \mathbf{a}_i'\hat{\mathbf{x}}^k \geq b_i \quad \forall i \in \mathcal{I}_1\}$ as two feasible regions. Let ρ_l and ρ be the metrics associated with feasible regions $\mathcal{S}_l$ and $\mathcal{S}$, respectively. Now, suppose that $\rho_l \neq \rho$. Therefore, there exists at least a point $\hat{\mathbf{x}}^k$ for which the minimum distance corresponds to its distance from constraint $\mathbf{a}_l'\hat{\mathbf{x}}^k \geq b_l$. This means $\mathcal{S}_l \neq \mathcal{S}$, which contradicts the redundancy of constraint $\mathbf{a}_l'\hat{\mathbf{x}}^k \geq b_l$. □

Proposition 2. *As the number of unknown constraints increases, the metric decreases monotonically.*

Proof. We add constraint $m_1 + 1$ to the set of unknown constraints. Let's define $\mathcal{E}^k_{m_1} = \{\epsilon^k_1, \epsilon^k_2, ..., \epsilon^k_{m_1}\}$ and $\mathcal{E}^k_{m_1+1} = \{\epsilon^k_1, \epsilon^k_2, ..., \epsilon^k_{m_1}, \epsilon^k_{m_1+1}\}$ for all $k \in \mathcal{K}$. Observe that the number of decision points is fixed and only the number of unknown constraints changes. Hence, we need to prove that the following equation holds for all points $k \in \mathcal{K}$.

$$\min_{e^k \in \mathcal{E}^k_{m_1}} e^k \geq \min_{e^k \in \mathcal{E}^k_{m_1+1}} e^k \quad \forall k \in \mathcal{K} \tag{3.10}$$

We prove inequality (3.10) by contradiction. Suppose that $\min\limits_{e^k \in \mathcal{E}^k_{m_1}} e^k < \min\limits_{e^k \in \mathcal{E}^k_{m_1+1}} e^k$, $\forall k \in \mathcal{K}$. Then $\exists w^k \in \mathcal{E}^k_{m_1+1}$,$\forall k \in \mathcal{K}$ such that $\min\limits_{e^k \in \mathcal{E}^k_{m_1}} e^k < w^k$, $\forall k \in \mathcal{K}$. Consider the following two cases:

Case 1: If $w^k = \epsilon^k_{m_1+1}$, $\forall k \in \mathcal{K}$, then $\epsilon^k_{m_1+1} \leq \min\limits_{e^k \in \mathcal{E}^k_{m_1+1}} e^k$, $\forall k \in \mathcal{K}$. This means that $\epsilon^k_{m_1+1} \leq \epsilon^k_i$, $\forall k \in \mathcal{K}, i \in \mathcal{I}_1$. But ϵ^k_i, $\forall k \in \mathcal{K}, i \in \mathcal{I}_1$ are all in $\mathcal{E}^k_{m_1}$, $\forall k \in \mathcal{K}, i \in \mathcal{I}_1$. This contradicts the assumption on $\min\limits_{e^k \in \mathcal{E}^k_{m_1}} e^k < \min\limits_{e^k \in \mathcal{E}^k_{m_1+1}} e^k$, $\forall k \in \mathcal{K}$

Case 2: If $w^k = \epsilon^k_i$, $\forall k \in \mathcal{K}, i \in \mathcal{I}_1$, then w^k, $\forall k \in \mathcal{K}$ is also in $\mathcal{E}^k_{m_1}$, $\forall k \in \mathcal{K}$. Again, this is a contradiction.

Hence, inequality (3.10) must hold. □

3.2.3 The goodness-of-fit measure in inverse optimization

We use the proposed goodness-of-fit measure as a new loss function in the objective of the IO problem (3.4). In this research, we focus on the Euclidean distance of decision points from constraints and chose the L$_2$ norm for the normalization constraint. With that, the IO problem will be as follows:

$$\min_{\mathbf{A},\mathbf{b},\boldsymbol{\epsilon}} \quad \frac{\sum_{k\in\mathcal{K}} \min_i \ \epsilon_i^k}{|\mathcal{K}|} \tag{3.11a}$$

$$\text{s.t.} \quad \epsilon_i^k = \frac{|\sum_{j\in\mathcal{J}} a_{ij}\hat{x}_j^k - b_i|}{||\mathbf{a}_i||_2} \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K} \tag{3.11b}$$

$$\mathbf{a}_i'\hat{\mathbf{x}}^k \geq b_i \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K} \tag{3.11c}$$

$$||\mathbf{a}_i||_2 = 1 \quad \forall i \in \mathcal{I}_1 \tag{3.11d}$$

$$\mathbf{A} \in \mathbb{R}^{m_1\times n}, \quad \mathbf{b} \in \mathbb{R}^{m_1} \tag{3.11e}$$

$$\boldsymbol{\epsilon} \in \mathbb{R}^{p\times m_1} \tag{3.11f}$$

Note that any other p-norm distance in equation (3.11b) and any other p-norm in equation (3.11d) can be used instead of Euclidean (L$_2$) norm. We deliberately chose the L$_2$ norm for the normalization constraint to simplify the Euclidean distance constraint.

The IO problem (3.11) is nonlinear due to the loss function and the normalization constraint (3.11d). To simplify the model, we reformulate it in two steps. We first linearize the min-min by introducing a non-negative continuous variable m^k and a binary variable γ_i^k. The result is as follows:

$$\min_{\mathbf{A},\mathbf{b},\boldsymbol{\epsilon},\mathbf{m},\boldsymbol{\gamma}} \quad \frac{\sum_{k\in\mathcal{K}} m^k}{|\mathcal{K}|} \tag{3.12a}$$

$$\text{s.t.} \quad \epsilon_i^k = \frac{|\sum_{j\in\mathcal{J}} a_{ij}\hat{x}_j^k - b_i|}{||\mathbf{a}_i||_2} \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K} \tag{3.12b}$$

$$m^k \geq \epsilon_i^k - M\gamma_i^k \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K} \tag{3.12c}$$

$$\sum_{i \in \mathcal{I}_1} \gamma_i^k = |\mathcal{I}_1| - 1 \quad \forall k \in \mathcal{K} \tag{3.12d}$$

$$\mathbf{a}_i' \hat{\mathbf{x}}^k \geq b_i \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K} \tag{3.12e}$$

$$||\mathbf{a}_i||_2 = 1 \quad \forall i \in \mathcal{I}_1 \tag{3.12f}$$

$$\mathbf{A} \in \mathbb{R}^{m_1 \times n}, \quad \mathbf{b} \in \mathbb{R}^{m_1} \tag{3.12g}$$

$$\boldsymbol{\epsilon} \in \mathbb{R}^{p \times m_1}, \quad \mathbf{m} \in \mathbb{R}^p, \quad \gamma_i^k \in \{0,1\} \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K} \tag{3.12h}$$

Next, we linearize the absolute value that exists in the distance equation (3.12b). Recall that $||\mathbf{a}_i||_2 = \sqrt{\sum_{j \in \mathcal{J}} a_{ij}^2}$. Therefore, the normalization constraint (3.12f), $||\mathbf{a}_i||_2 = 1$, is equal to $\sum_{j \in \mathcal{J}} a_{ij}^2 = 1$. Hence, when using the Euclidean distance, the denominator of the equation can be set to one. With that, an equivalent simplified version of the IO problem can be written as follows:

$$\min_{\mathbf{A},\mathbf{b},\boldsymbol{\epsilon},\mathbf{m},\boldsymbol{\gamma}} \quad \frac{\sum_{k \in \mathcal{K}} m^k}{|\mathcal{K}|} \tag{3.13a}$$

$$\text{s.t.} \quad \epsilon_i^k \geq \sum_{j \in \mathcal{J}} a_{ij} \hat{x}_j^k - b_i \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K} \tag{3.13b}$$

$$\epsilon_i^k \geq -\sum_{j \in \mathcal{J}} a_{ij} \hat{x}_j^k + b_i \quad \forall i \in \mathcal{I}_1, k \in \mathcal{K} \tag{3.13c}$$

$$\sum_{j \in \mathcal{J}} a_{ij}^2 = 1 \quad \forall i \in \mathcal{I}_1 \tag{3.13d}$$

$$(3.12c) - (3.12e), (3.12g), (3.12h) \tag{3.13e}$$

Note that in the IO problem (3.13), all components are linear except for constraint (3.13d).

3.2.4 Numerical example

In this part, we illustrate the second property of the metric through a numerical example taken from Ghobadi and Mahmoudzadeh (2021). Suppose that there are 10 observations and 2 known constraints (including the half-space $\mathcal{C}$) as listed in Table 3.1. By solving the IO problem (3.13) 3 times, we infer 1, 2, and 3 constraints. The results are summarized in Figure 3.2. The observations are represented by green dots, with the preferred observation highlighted in red. The inferred constraints are depicted as dashed gray lines, while the known constraints are represented by dotted red lines. The feasible region of the problem is shaded in yellow.

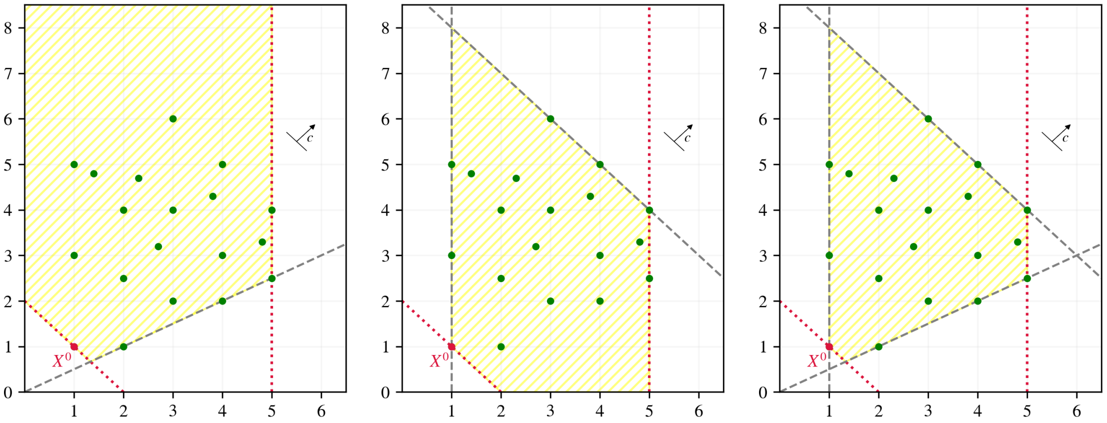
Starting with only one inferred constraint, we observe the poorest value for the goodness-of-fit measure ($\rho = 1.2281$). As we increase the number of inferred constraints up to 3, the goodness-of-fit measure consistently improves to $\rho = 0.6504$ and then to $\rho = 0.4709$. This occurs because the distances between the observations and the inferred constraints keep getting smaller as the feasible region becomes tighter. This example confirms the second property listed in section (3.2.1), meaning that when constraints are added, the goodness-of-fit measure improves and therefore the fit gets better.

3.2.5 Comparison between the goodness-of-fit measure and other loss functions

To evaluate the performance of our inverse model and the third property of the metric, we conduct a comparative analysis with the IO problems in the literature. The most recent study on inferring the feasible region is the work of Ghobadi and Mahmoudzadeh (2021). Their approach employs four distinct loss functions to address the inverse problem. They

Table 3.1: Numerical example

Description	Value(s)
Cost vector $\mathbf{c}$	(1,1)
Observations $\hat{\mathbf{x}}^0$;$\hat{\mathbf{x}}^k$	(1,1); (2,1), (4,2), (4,5), (3,6), (2,4), (3,4), (3,2), (4,3), (1,3), (2,2.5), (1,5), (5,2.5), (5,4), (2.7,3.2), (2.3,4.7), (1.4,4.8), (3.8,4.3), (4.8,3.3)
Known constraints	$0.5\hat{x}_1^k + 0.5\hat{x}_2^k \geq 1$ (half-space $\mathcal{C}$) -$\hat{x}_1^k \geq -5$



(a) One inferred constraint

$\rho = 1.2281$

(b) Two inferred constraints

$\rho = 0.6504$

(c) Three inferred constraints

$\rho = 0.4709$

Figure 3.2: Illustration of the numerical example

used the linear slack distance (i.e., $\epsilon_i^k = \mathbf{a}_i'\hat{\mathbf{x}}^k - b_i$) to calculate the distance of decision point $\hat{\mathbf{x}}^k$ from constraint $\mathbf{a}_i'\hat{\mathbf{x}}^k \geq b_i$. They also used an approximation of L_1 norm (i.e., $|\sum\limits_{j\in\mathcal{J}} a_{ij}| = 1$) for the normalization constraint. Together, these settings allow for their IO model to be linear. In contrast, our approach involves using the Euclidean distance and L_2 norm for these computations. By comparing our results with theirs across these four loss functions, we can gain insights into the performance of the models.

Consider the data in Table 3.1. In this case, we infer 6 unknown constraints first using indifference (3.5), adjacency (3.6), fairness (3.7), and compactness (3.8) measures and then using our proposed goodness-of-fit measure (3.9). The results are illustrated in Figure 3.3. The green dots depict the observations, and the red dot signifies the preferred observation. The dashed gray lines represent the constraints that are inferred, while the dotted red lines represent the known constraints. The feasible region of the problem is also shaded in yellow. The redundant constraints are included only once, and the number of repetitions is indicated next to the corresponding hyperplane. For example, (4×) next to a hyperplane indicates that out of the 6 inferred constraints, 4 of them are repeated. Two key observations can be inferred from the results in Figure 3.3:

1. Prior loss functions from the literature find redundant constraints. This means that these measures introduce additional constraints that do not contribute significantly to improving the fit of the model. On the other hand, our proposed measure successfully avoids such redundancy by selecting only the essential constraints for achieving a desirable fit.

2. Our model outperforms other models in terms of the value of ρ, indicating a superior fit between the model and the observed data. This improvement can be attributed to the utilization of our proposed goodness-of-fit measure, which leads to a tighter

feasible region.

Solving the IO problem (3.13) is challenging, particularly due to the presence of the nonlinear constraint (3.13d). In the following section, we propose solution methods to mitigate the complexity of the problem.

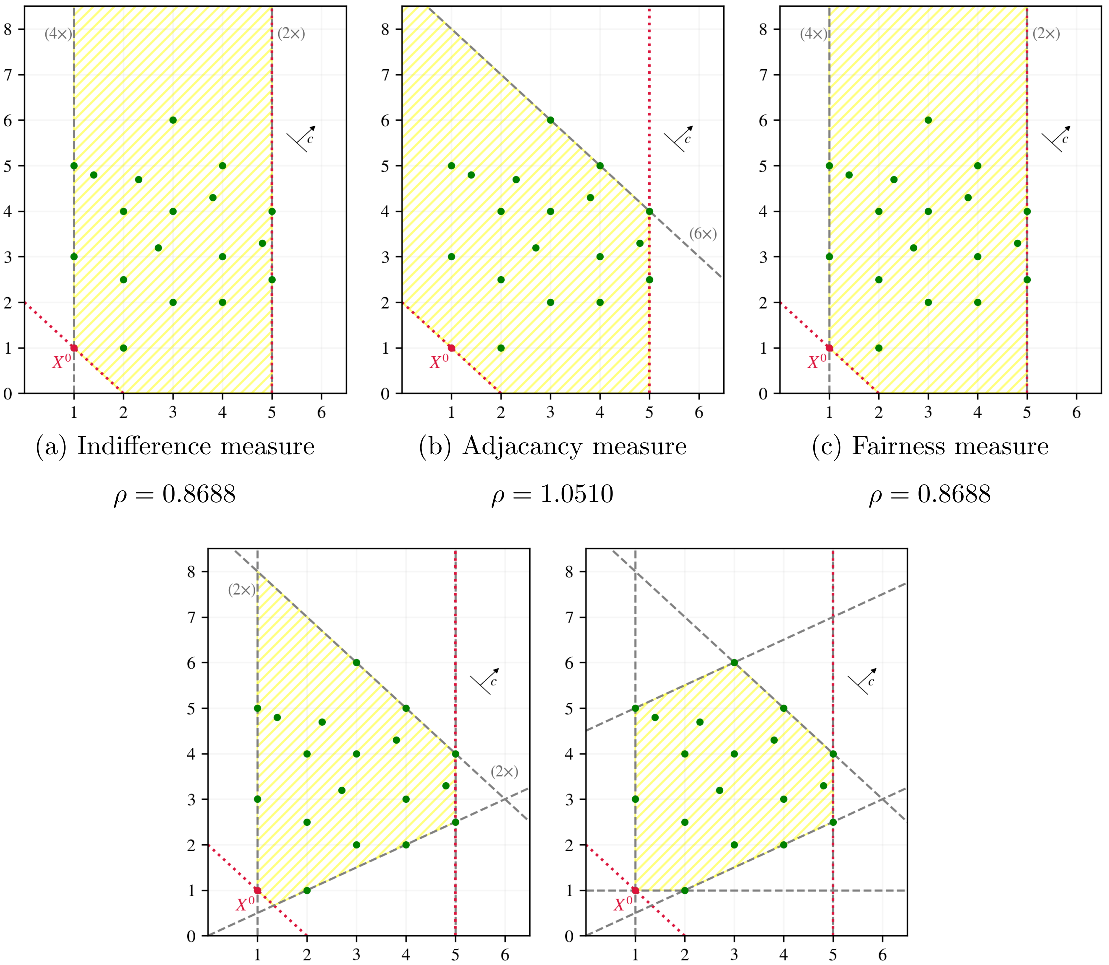
3.3 Solution approach

The final IO model is a non-convex Mixed-integer quadratically constrained programming (MIQCP). The complexity of the problem (3.13) can be reduced if we make the feasible region convex and linear. Looking at the constraints, the only nonlinear constraint is equation (3.13d), which causes a non-convex set. Therefore, we relax the equality constraint (3.13d) and move it to the objective function in order to deal with the difficulty of the problem. This way, the problem is minimized over a linear and convex set of constraints.

In this section, we first employ the non-smooth L_1 penalty function to provide a more tractable model for the IO problem. Then, we propose an exact heuristic method and a greedy heuristic method to solve the problem efficiently. Appendix A explains why we have chosen the L_1 penalty function instead of the lagrangian relaxation for this problem.

3.3.1 Relaxation using L_1 penalty function

The L_1 penalty function method adds a penalty term to the objective function that penalizes the violation of the L_1 norm of the quadratic constraint (3.13d). The absolute value will ensure that the constraint is satisfied in the original problem. Assuming that μ is a positive penalty, we are summing up some non-negative terms and the minimum occurs



(a) Indifference measure

$\rho = 0.8688$

(b) Adjacency measure

$\rho = 1.0510$

(c) Fairness measure

$\rho = 0.8688$

(d) Compactness measure

$\rho = 0.4709$

(e) Proposed goodness-of-fit metric

$\rho = 0.4450$

Figure 3.3: Comparing the loss functions in the literature with our goodness-of-fit metric

when each term is zero.

$$\min_{\mathbf{A},\mathbf{b},\boldsymbol{\epsilon},\mathbf{m},\boldsymbol{\gamma}} \quad \frac{\sum\limits_{k\in\mathcal{K}} m^k}{|\mathcal{K}|} + \mu \sum_{i\in\mathcal{I}_1} |\sum_{j\in\mathcal{J}} a_{ij}^2 - 1| \tag{3.14a}$$

$$\text{s.t.} \quad (3.13b), (3.13c), (3.13e) \tag{3.14b}$$

Let h be the measure of infeasibility. The algorithmic framework based on the L_1 penalty function to solve the model (3.14) is presented in Algorithm 1 (Wright et al., 1999).

Algorithm 1: Classical L_1 penalty method

```
Given penalty $\mu_0 > 0$, tolerance $\zeta > 0$
for $r = 0, 1, 2, ...$ do
    find an approximate solution for problem (3.14);
    if $h \leq \zeta$ then
        stop;
    end
    Choose new penalty parameter $\mu_{r+1} > \mu_r$;
end
```

If the current value of μ_r results in a minimizer that is not feasible within the tolerance ζ, the simplest scheme for updating the penalty parameter μ_r is to increase it by a constant multiple in the next iteration (Wright et al., 1999).

3.3.2 Exact heuristic

Model (3.14) is Mixed-integer quadratic programming (MIQP) and it is computationally expensive to be solved by mathematical optimization; specially when the size of the problem increases. To overcome this, we develop a heuristic algorithm which gives the exact solution of the inverse model. The proposed approach consists of four steps. Firstly, the set of hyperplanes that form the convex hull of the observations is identified. This can be done via the ConvexHull class in SciPy, a library in Python (Virtanen et al., 2020). In the second step, all possible combinations of m_1 hyperplanes out of all hyperplanes forming the convex hull (h) are generated using Equation (3.15). The metric associated with each combination is computed in the next step. Finally, the combination of hyperplanes that yields the minimum value of the metric is selected as the solution. The description of the heuristic is presented in Algorithm 2.

$$C_{m_1}(h) = \binom{h}{m_1} = \frac{h!}{m_1!(h-m_1)!} \tag{3.15}$$

Algorithm 2: Exact heuristic approach

```
Step 1: Identify the convex hull of the data points
Step 2: Generate all combinations of $m_1$ hyperplanes
Step 3: Calculate the value of the metric for all combinations
Step 4: Select the combination of $m_1$ hyperplanes with the minimum value of the
  metric
```

For example, suppose that given a number of observations, we want to infer $m_1 = 2$ constraints. In the first step, we find the convex hull of observations. Let's assume that the convex hull of observations consists of 10 hyperplanes. In the second step, we find all combinations of $m_1 = 2$ hyperplanes out of 10 hyperplanes that form the convex hull. In this case, there will be 45 possible combinations, derived from Equation (3.15). Note that each combination represents a potential solution for the problem. Next, in the third step, we calculate the value of the metric for all 45 combinations using Equation (3.9). Finally, in the fourth step, we compare the combinations (potential solutions) and select the one with the minimum value of the metric, as it represents the optimal solution to our problem.

Proposition 3. *The solution from the IO problem (i.e.* $\mathbf{Ax} \geq \mathbf{b}$ *) consists of all/some hyperplanes forming the convex hull of the observations. This means if the hyperplanes forming the conv*$(\hat{\mathbf{x}}^1, ..., \hat{\mathbf{x}}^p)$ *are indexed by the set* $\mathcal{H}$*, then* $\underset{i \in \mathcal{I}_1}{min}\ \epsilon_i^k = \epsilon_{i \in \mathcal{H}}^k, \forall k \in \mathcal{K}$*.*

Proof. Suppose that $\exists \hat{\mathbf{x}}^k, k \in \mathcal{K}$ for which $\min_{i \in \mathcal{I}_1} \epsilon_i^k = \epsilon_{i \notin \mathcal{H}}^k$. By definition, $conv(\hat{\mathbf{x}}^1, ..., \hat{\mathbf{x}}^p)$ is the smallest convex set containing all $\hat{\mathbf{x}}^k, k \in \mathcal{K}$. Therefore, $\exists \mathbf{a}_i' \hat{\mathbf{x}}^k - b_i = 0, i \in \mathcal{H}$ such that $\min_{i \in \mathcal{I}_1} \epsilon_i^k = \epsilon_{i \in \mathcal{H}}^k, \forall k \in \mathcal{K}$. This is a contradiction. □

3.3.3 Greedy heuristic

While the heuristic method discussed in the previous section offers improved computational efficiency compared to solving the mathematical optimization model, it still encounters challenges when dealing with larger-scale problems. The method finds the optimal solution by enumerating potential solutions and this enumeration process can become burdensome, particularly for larger-sized problems. Given the complexity of the proposed heuristic, we introduce a faster alternative in the form of a greedy heuristic. However, it is important to acknowledge that the trade-off for the reduced runtime in this greedy approach is a compromise in solution quality compared to exact methods. The initial step is identical to the first step of the heuristic algorithm. In the second step, we approach the problem by considering each hyperplane obtained in the first step individually and evaluating the Euclidean distances associated with a single hyperplane. The description of the greedy heuristic is presented in Algorithm 3.

```
Algorithm 3: Greedy heuristic approach
  Step 1: Identify the convex hull of the data points
  Step 2: Calculate the total Euclidean distance of each hyperplane forming the
    convex hull from all data points, independent of other hyperplanes
  Step 3: Select $m_1$ hyperplanes with the minimum total Euclidean distance
```

For example, suppose that given a number of observations, we want to infer $m_1 = 2$ constraints. In the first step, we find the convex hull of observations. Let's assume that the convex hull of observations consists of 10 hyperplanes. In the second step, we calculate the total Euclidean distance of each hyperplane forming the convex hull from all observed data points, independent from other hyperplanes. In the third step, we rank the hyperplanes based on their total Euclidean distance, arranging them in ascending order from the lowest to the largest distance, then we select the $m_1 = 2$ hyperplanes with the minimum distance as the solution to the problem.

We will compare the computational performance of L_1 penalty function, exact heuristic, and greedy heuristic methods in section 4.1.6.

Chapter 4

Results and Discussions

In this chapter, we investigate the application of the proposed methodology to two distinct problems. Through these numerical applications, we gain insights into the model's performance and conduct a comparative analysis of the proposed solution methods. First, in section 4.1, we consider the patient classification problem in hospital admission and evaluate the effectiveness of the IO model by examining multiple instances. Next, in section 4.2, we consider the diet recommendation problem and demonstrate how the IO method can be utilized to design more appealing diet plans based on the dieter's past food choices.

4.1 Patients classification problem

The patient classification problem at hospitals occurs when a triage process results in deciding whether a patient requires hospitalization (inpatient) or can be treated outside the hospital setting (outpatient). Although there are some guidelines in place to make such decisions, not all cases adhere strictly to these guidelines. For instance, we may see a patient who has characteristics contrary to guidelines but is released from the hospital. Hence, we aim to identify the implicit constraints that influenced the decision of whether a patient is considered inpatient or outpatient. Let's consider a scenario where we want to find out the factors influencing hospital discharge decisions. By utilizing past health data of outpatients (feasible observations), we can employ IO to identify a set of constraints (feasible region) that captures the process of identifying outpatients.

In this section, we first provide an overview of the dataset and the existing medical guidelines. Next, we perform feature selection to identify the important features. Following that, we present a two-dimensional illustration of our proposed methodology. Then, we evaluate the prediction accuracy of the model. Finally, we examine the computational performance of the solution methods using multiple instances.

4.1.1 Dataset

We employ an openly accessible dataset obtained from Alam et al. (2022) which is comprised of electronic health records of patients from a private hospital in Indonesia. It contains the patients' laboratory test results used to determine the next course of action for patients, including their health assessment and treatment plan. For each patient, a record of multiple blood component measures in the laboratory test result is reported. The description of features in the dataset is summarized in Table 4.1.

To provide further details, patients are categorized into two groups based on their test results. One group consists of inpatients who require hospitalization, meaning that they need ongoing medical attention and monitoring within the hospital environment. The second group includes outpatients who do not need to stay at the hospital overnight as their test results do not necessitate hospitalization. Note that we will narrow down the population under study by focusing only on the data of women.

4.1.2 Existing medical guidelines

Medical guidelines provide reference ranges for each blood component, which define the normal (healthy) values. A reference range consists of upper and lower limits that indicate the acceptable values for a laboratory test. Table 4.2 lists the reference ranges for the blood components of female patients in our dataset (Sadikin et al., 2021). In practice, the decision-making process regarding whether patients should be discharged from the hospital or stay requires consideration of various implicit criteria, which may be based on a combined effect of guidelines, so these decisions may not always adhere to a specific set of established guidelines. We hypothesize that there are certain trade-offs between different hidden factors that influence the decision-making process.

Our goal is to elucidate the decision-making process for hospital discharge by understanding the underlying criteria that determine the severity of a patient's illness. Using the laboratory report data of a set of past patients, we employ the IO method and derive a set of implicit criteria that result in the discharge of outpatients.

4.1.3 Feature selection

To reduce the complexity of the model, we employ feature selection techniques to identify the subset of features that exhibit the strongest relationship with the target variable. In our dataset, the target variable is represented in the last column, consisting of nominal values. If the patient is identified to be an outpatient, the value assigned is “out”, otherwise, it is “in”.

Table 4.1: Description of features

Feature name	Value sample	Brief description
HAEMATOCRIT	35.1	The ratio of the volume of red blood cells to the total volume of blood.
HAEMOGLOBINS	11.8	A protein in the red blood cells.
ERYTHROCYTE	4.65	Red blood cells that deliver oxygen to the tissues in the body.
LEUCOCYTE	6.3	White blood cells that are responsible for protecting the body from infection.
THROMBOCYTE	310	Platelets that control blood clotting.
MCH	25.4	The average amount of hemoglobin found in the red blood cells.
MCHC	33.6	The mean corpuscular hemoglobin concentration in the red blood cells.
MCV	75.5	The mean corpuscular volume of the red blood cells.
AGE	12	The age of the patient.
SEX	F	The gender of the patient.
SOURCE	out	The severity level of the patient's illness.

Table 4.2: Reference ranges for blood components

Blood component	Reference range
HAEMATOCRIT	34.9%-44.5%
HAEMOGLOBINS	12-15 g/dL
ERYTHROCYTE	4.2-5.4 $million/mm^3$
LEUCOCYTE	5-10 x $thousand/mm^3$
THROMBOCYTE	150-400 $thousand/mm^3$
MCH	27-33 $pg/cell$
MCHC	32%-37% $Hb/cell$
MCV	80-96 μm^3

Since the input data (i.e., all features except the target variable) is numerical and the output (i.e., the target variable) is a categorical variable, we employ Analysis of variance (ANOVA) that uses the F-test to select the useful features (Brownlee, 2020). In the F-test, the ratio of between and within group variances is calculated and this is called F-statistic (Kim, 2017). The feature selection is done by the SelectKBest class in scikit-learn, a library in Python, in which a score is given to each feature based on the strength of its relationship with the target variable (Pedregosa et al., 2011). The SelectKBest class scores the features using a function and since we are implementing ANOVA, we use f_classif function which computes F-statistic. The higher the score is, the more relevant the feature will be. Figure 4.1 illustrates the results after performing the feature selection process. The y-axis displays the F-statistic, indicated as scores. It is observed that the features HAEMATOCRIT, HAEMOGLOBINS, ERYTHROCYTE, and THROMBOCYTE exhibit higher scores compared to other features. Hence, we only consider these four features as the input data in the rest of our analysis.

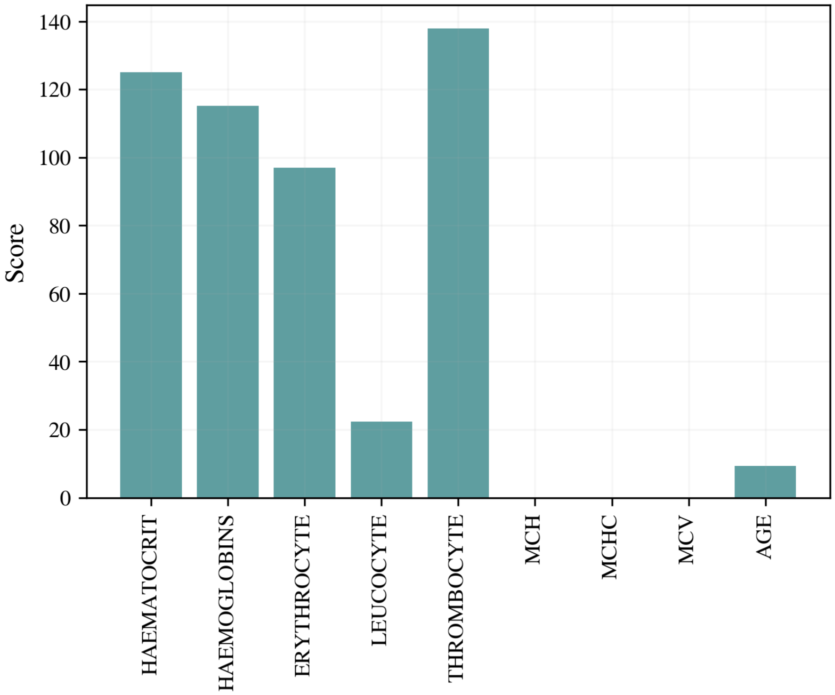
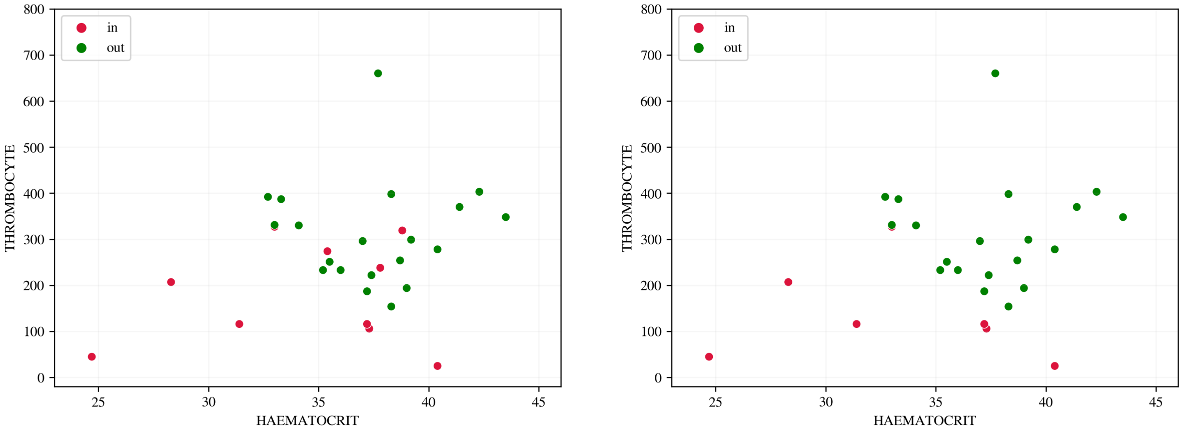


Figure 4.1: The ANOVA F-test feature importance

4.1.4 Illustrative results for a pair of features

To provide a clearer understanding of the proposed methodology, we first showcase a two-dimensional instance from the dataset before proceeding to test the model and solution methods. The example comprises 30 data points and includes 2 features only: HAEMATOCRIT and THROMBOCYTE. We perform a sensitivity analysis on the number of constraints by incrementally adding one constraint at a time, starting from 1 constraint and gradually increasing it to 6 constraints. The data points are displayed in Figure 4.2. The outpatient and inpatient data points are shown in green and red circles, respectively.



(a) Before cleaning the data

(b) After cleaning the data

Figure 4.2: Data points

To clean the data for the purpose of this numerical case, we exclude the inpatient data points that are situated within the convex hull of outpatients, because they are inconsistent with other decisions.

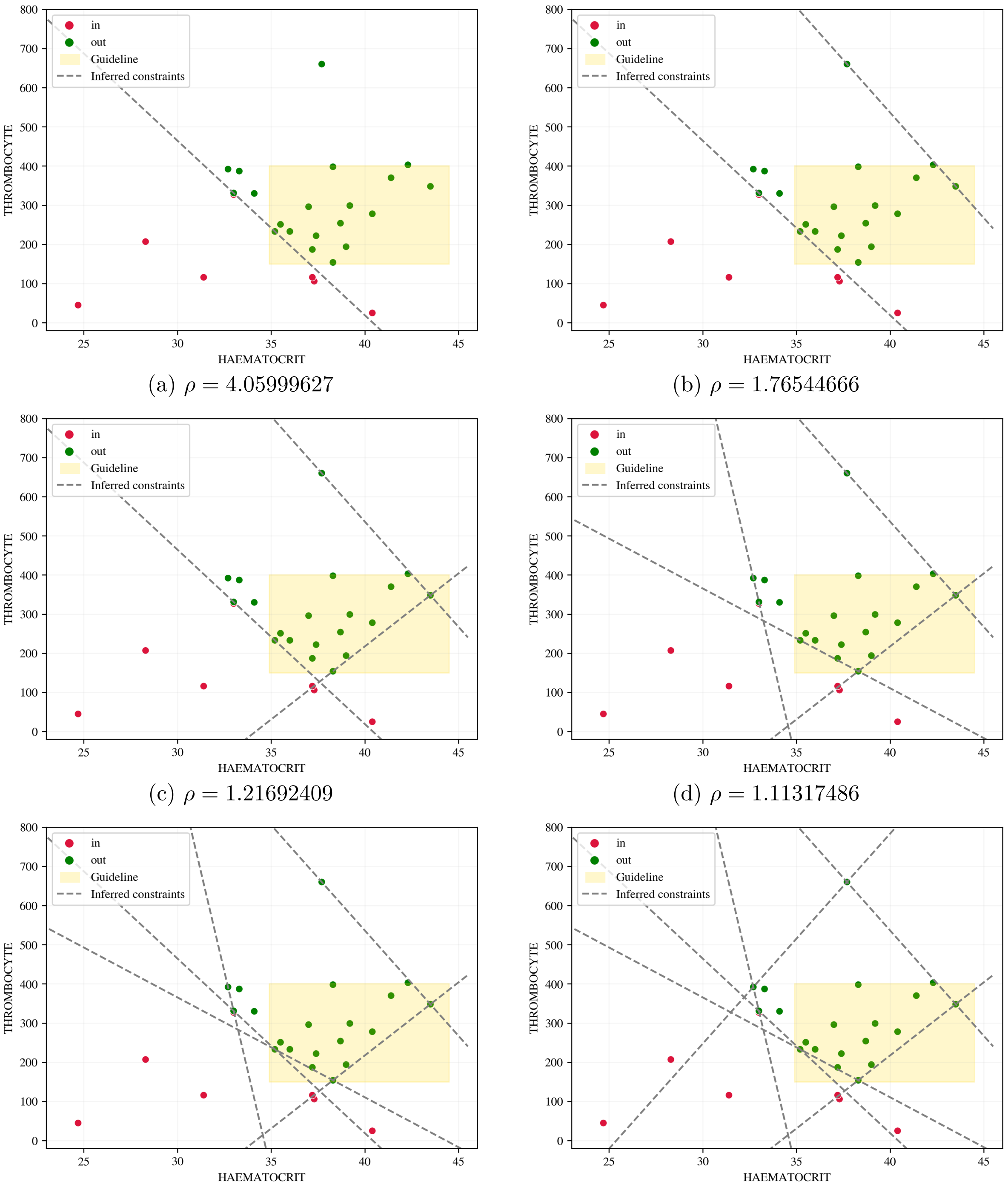
Figure 4.3 presents the outcomes of solving the IO problem (3.14) that aims to infer the feasible region of outpatient data points for different configurations, with a fixed number of data points and features while varying the number of constraints inferred. The outpatient and inpatient data points are shown in green and red circles, respectively. The inferred constraints are shown by dashed gray lines, while the guideline is displayed within a yellow-shaded box. The value of ρ for each instance is also provided. Initially, the first constraint is inferred to minimize the distance of all outpatient data points from it. Because of the min-min in the goodness-of-fit measure which tries to minimize the minimum distances for all data points, in the second setting, the second hyperplane is determined to pass through the points that are farthest from the first hyperplane. As we move on to inferring three constraints, the hyperplanes derived in the previous settings are included, along with a new hyperplane. This pattern continues until the feasible region is completed with the addition of the sixth hyperplane. It is important to note that further constraints will not result in an improvement of the metric.

The presented two-dimensional instance serves to illustrate the second property of the goodness-of-fit measure. As depicted in Figure 4.4, it is observed that the metric consistently decreases in a monotonic manner as the number of constraints increases. However, it is worth noting that the decrease in the metric becomes less significant as we increase the number of constraints. After 5 constraints, the graph levels off, indicating that additional constraints do not contribute to further improvement in the fit. This is because once the convex hull is formed, further improvements in the fit are not achievable, and thus the metric remains unchanged. This example also confirms Proposition 2, which states that the metric decreases monotonically as the number of inferred constraints increases.

4.1.5 Prediction accuracy

In this numerical case, the forward problem is a feasibility problem since we are not optimizing the problem over a certain objective function. Instead, our goal is to determine the feasible region of the problem. In particular, we aim to identify the implicit criteria that led to recognizing specific patients as outpatients. This understanding allows us to gain insights into the decision-making process for patient release and utilize this knowledge for future observations. Hence, evaluating the model's effectiveness in correctly identifying outpatients becomes an important aspect.

Given that the IO model is trained only on the data of outpatients, our evaluation focuses on the number of correctly predicted outpatients and the number of inpatients mistakenly classified as outpatients. Therefore, we assess the model's performance using



(a) $\rho = 4.05999627$

(b) $\rho = 1.76544666$

(c) $\rho = 1.21692409$

(d) $\rho = 1.11317486$

(e) $\rho = 1.06648537$

(f) $\rho = 1.06648532$

Figure 4.3: Sensitivity analysis on the number of the constraints

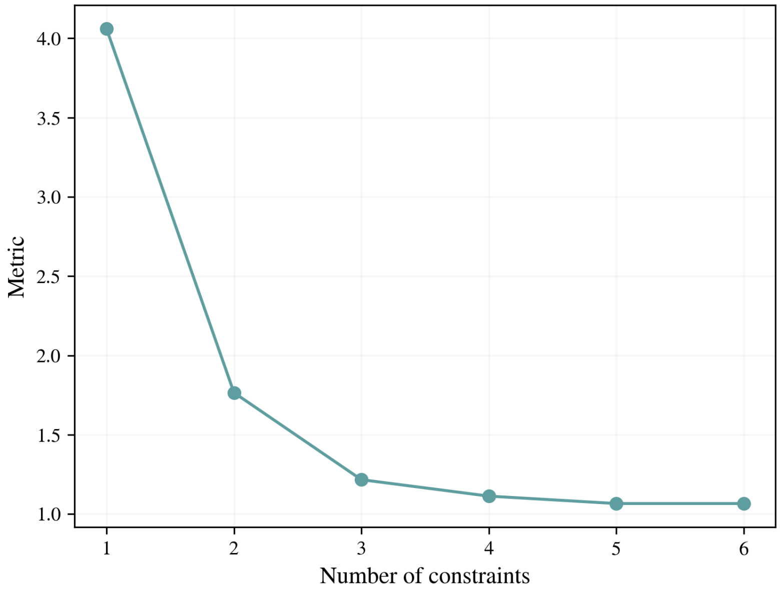


Figure 4.4: Monotonic trend in the metric value

the precision metric, which provides us with an understanding of how accurate the model is in identifying the outpatients.

We generate 5 random datasets, each consisting of 50 observations with the features HAEMATOCRIT and THROMBOCYTE. To evaluate the model's performance, we employ an 80/20 random split for training and testing. In the training set, we identify the outpatients and use this subset as input for the IO model to infer 5 constraints. Next, we apply the inferred constraints to the testing set and calculate the precision metric. We repeat this process 10 times for each dataset and compute the average precision within each dataset. Finally, we determine the overall average precision across the 5 datasets. In Figure 4.5, the ranges of precisions within each dataset are represented by error bars, and the average value is marked with red circles on them. The results demonstrate an average precision of 81.42% across all trials.

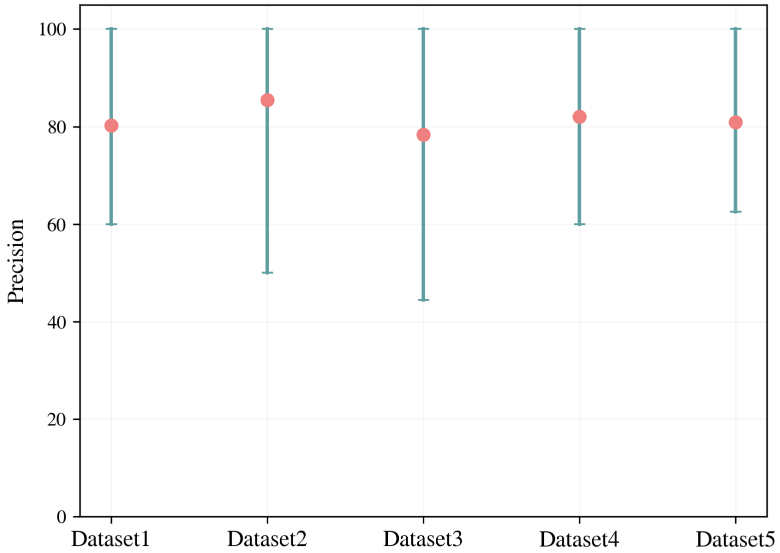


Figure 4.5: Precision of the inverse model

4.1.6 Computational performance of the solution methods

In this section, the computational performance of different solution methods presented in section 3.3 is assessed. This analysis allows us to identify the strengths and limitations of each method. For this purpose, we create 15 instances from the dataset, each varying in the number of observations and unknown constraints. The first set of instances includes 20 observations, the second set includes 50 observations, and the third set includes 100 observations. Within each set, we infer a varying number of constraints, ranging from 1 to 5. The results are summarized in Table 4.3. The IO problem (3.14) is labeled as "MIP". The exact and greedy heuristic methods are labeled as "Heu-E" and "Heu-G", respectively. For each solution method, two numbers are reported: the metric value (ρ) and the runtime. The optimization problems are solved by Gurobi Optimizer 10.0 in AMPL with an 8-core 3.00 GHz processor and 16 GB memory. The heuristic algorithms are coded in Python programming language using the Graham cluster located at the University of Waterloo with a 32-core and 125 GB memory. Also, all instances are run 5 times and the reported runtime is the average.

The results indicate the computational complexity associated with solving the "MIP" using the Gurobi solver. The solver takes a long time to search the solution space and it fails to converge to the optimal solution within 24 hours for several instances (the dash mark in Table 4.3 denotes this). To gain a better understanding of the level of difficulty involved in solving the problem by the exact mathematical method, Figure 4.6 is provided as an illustration of the growth rate observed in the number of binary variables and constraints as we progressively increase the number of observations and unknown constraints. The x-axis is divided into three sets of observations: 20, 50, and 100. The numbers on the x-axis indicate the number of inferred constraints within each set, ranging from 1 to 5. On the left y-axis, the blue graph represents the number of constraints, while on the right y-axis, the red graph represents the number of binary variables. As the size of the problem increases, we observe a linear increase in the number of constraints and binary variables. Notably, this growth speeds up when the number of observations is larger. According to the IO problem (3.14), the number of binary variables is $|\mathcal{I}_1| \times |\mathcal{K}|$, whereas the number of constraints is $4(|\mathcal{I}_1| \times |\mathcal{K}|) + |\mathcal{K}|$. For example, instance 7 which involves 50 observations and 2 unknown constraints, includes 100 binary variables and 450 constraints.

The exact heuristic method, on the other hand, performs well by not only delivering the optimal solution but also significantly improving the runtime compared to the previous mathematical approach. It is evident that as the size of the instances grows, the complexity and time required to solve instances using the exact heuristic method also increases. The bottleneck in the exact heuristic algorithm lies in the calculation of the metric for all combinations of $|\mathcal{I}_1|$ constraints out of the total constraints that form the convex hull. Figure

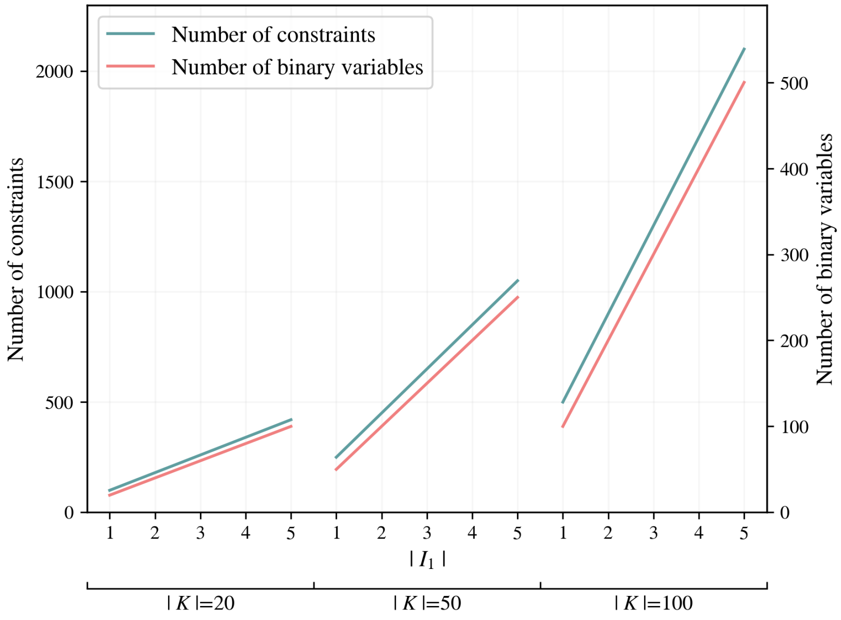


Figure 4.6: Complexity growth rate in the optimization problem

4.7 illustrates the exponential growth in the number of combinations of $|\mathcal{I}_1|$ constraints and runtime in each instance as the problem size increases. Note that the y-axis represents the values on a logarithmic scale. The x-axis is divided into three sets of observations and the numbers on it indicate the number of inferred constraints within each set, ranging from 1 to 5. On the left y-axis, the red graph represents the number of combinations, while on the right y-axis, the blue graph represents the runtime. To provide an example, let's consider instance 7 with 50 observations, where 2 constraints are inferred. In this case, the convex hull formed by these 50 observations consists of 73 hyperplanes. This value is computed using the ConvexHull class in SciPy, a library in Python (Virtanen et al., 2020). Then, using Equation (3.15), there are 2,628 possible combinations of 2 constraints out of the total 73 constraints. If we maintain the same number of observations and increase the problem size to infer 4 constraints, the number of possible combinations calculated using

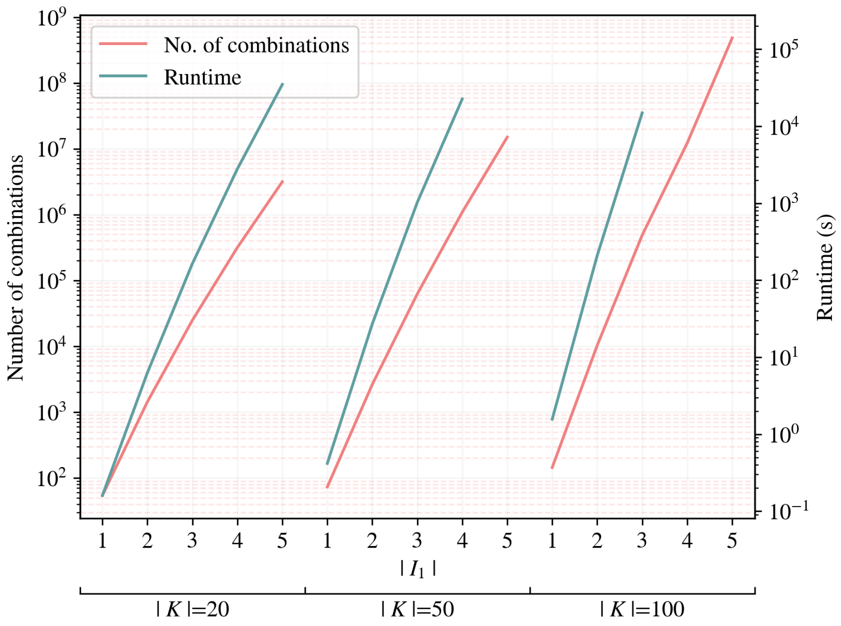


Figure 4.7: Complexity growth rate in the exact heuristic method (y-axis is in log scale)

Equation (3.15) increases exponentially to 1,088,430 cases. This exponential increase in combinations highlights the computational complexity of the problem as it scales up.

The runtimes for the greedy heuristic method are much shorter than the other two methods, with all instances being solved in less than 3 seconds. However, as expected, the trade-off is observed in the quality of the solution. To provide further details, when comparing the Metric column between the greedy heuristic and exact heuristic methods in each set, we can observe a more noticeable decrease in the exact heuristic method as the number of inferred constraints increases, indicating an improvement in the fit. Conversely, the decrease in the metric value is less substantial in the greedy heuristic method. Given that the metric represents the fit between the model and the data, it can be concluded that the solution quality is comparatively lower in the greedy heuristic method. For instance, let's consider instance 5 with 20 observations and 5 inferred constraints. The solution obtained through the exact heuristic method yields a fit of 0.0185, while the greedy heuristic method yields a fit of 0.2710. Recall that a lower metric value corresponds to a better fit.

4.2 Diet planning problem

The diet planning problem involves designing diet plans that effectively meet both the nutritional needs and food preferences of individuals. This (forward) optimization problem typically involves maximizing or minimizing specific nutrients or food components, while adhering to various constraints such as vitamin intake or calorie limits.

When developing a diet plan, the initial focus is the nutritional requirements. But it is also important to ensure that the plan aligns with the dieter's preferences as there may be some implicit criteria that influence the palatability of a diet plan for the dieter. Consider a dieter who prefers a vegetable-heavy diet with limited meat consumption, or another dieter who prefers to consume fewer dairy products. By understanding their food preferences, we can suggest diet plans that are more likely to be followed and enjoyed. To achieve this, we observe the dieter's past food choices (feasible observations) and use IO to impute a set of constraints (feasible region) that captures their dietary behavior. This allows us to provide food plans that are palatable.

As the second application to evaluate the effectiveness of the proposed methodology, we explore the problem of diet recommendation. We aim to uncover the underlying criteria that influence an individual's food choices through IO techniques. By analyzing the dieter's historical food preferences, our model identifies hidden linear constraints that shape their decision-making process.

In the rest of this section, we provide an explanation of the dataset. Following that,

Table 4.3: Computational performance results of the solution methods

Instance	$\lvert\mathcal{K}\rvert$	$\lvert\mathcal{I}_1\rvert$	MIP		Heu-E		Heu-G		
			Metric	Runtime (s)	Metric	Runtime (s)	Metric	Runtime (s)	Optimality gap
1	20	1	0.3039	1	0.3039	0.16	0.3039	0.185	0
2	20	2	0.1050	79	0.1050	6.29	0.3022	0.187	1.88
3	20	3	-	-	0.0540	164.59	0.3008	0.192	4.57
4	20	4	-	-	0.0288	2793	0.2987	0.228	9.37
5	20	5	-	-	0.0185	35000	0.2710	0.251	13.68
6	50	1	0.4677	1	0.4677	0.42	0.4677	0.596	0
7	50	2	-	-	0.3080	27.09	0.4579	0.620	0.49
8	50	3	-	-	0.2196	1026	0.4197	0.630	0.91
9	50	4	-	-	0.1891	22660	0.4191	0.688	1.22
10	50	5	-	-	-	-	0.3475	0.699	NA
11	100	1	0.5095	1	0.5095	1.57	0.5095	2.321	0
12	100	2	-	-	0.3555	209	0.4917	2.342	0.38
13	100	3	-	-	0.2771	14940	0.4896	2.355	0.77
14	100	4	-	-	-	-	0.4673	2.397	NA
15	100	5	-	-	-	-	0.4571	2.480	NA

we analyze the results obtained from applying IO to the diet recommendation problem.

4.2.1 Dataset

The dataset utilized in this study captures 30 observations of the daily food intake recorded as the number of servings consumed per food per day. The data is obtained from Ahmadi et al. (2020b). The original dataset consists of numerous food items, and to simplify the analysis, we categorize them into 5 main food groups: fruits, vegetables, grains, protein, and dairy.

For the purpose of this study, we assume that the objective function of the FO problem is to minimize the total fat. In addition, we impose a set of 3 known constraints related to key nutrition values: a lower bound on protein, an upper bound on calories, and an upper bound on carbohydrates. Notably, all 30 observations in the dataset satisfy these constraints, ensuring that the selected food choices meet the specified nutritional requirements. These constraints are based on the guidelines provided by the U.S. Department of Health and Human Services and U.S. Department of Agriculture (2020). Furthermore, to determine the nutrition values for one serving of each food group, we relied on available data from Government of Canada (2008). The serving size for each food group and a concise summary of the model specifications can be found in Tables 4.4 and 4.5, respectively. Recall that $\hat{x}_j^k$, $\forall k \in \mathcal{K}, j \in \mathcal{J}$, represent the observed data points, where observations are indexed by set $\mathcal{K} = \{1, ..., 30\}$ and the indices of variable $\mathbf{x}$ is denoted by set $\mathcal{J} = \{1, ..., 5\}$, as we have 30 observations and 5 food groups. With that, the detailed description of known constraints can be found in Table 4.6. Observe that, along with 3 known constraints on nutritional requirements, there is an additional constraint represented by the half-space $\mathcal{C}$ ($\mathbf{c}'\hat{\mathbf{x}}^k \geq \mathbf{c}'\hat{\mathbf{x}}^0, \forall k \in \mathcal{K}$). The nutrient coefficients in the objective and known constraints, as

Table 4.4: Food groups and their respective serving sizes

Food Category	Serving Size (Example)
Fruits	1 medium apple (about 150 grams)
Vegetables	1 cup (about 30 grams) of raw, leafy green vegetables
Grains	1 slice of whole-grain bread (about 36 grams)
Protein	1 ounce of cooked meat
Dairy	1 cup (240 ml) of milk

Table 4.5: Summary of the diet recommendation problem

Description	Value(s)
Cost vector $\mathbf{c}$	Minimize total fat intake
Observations	30 daily food intakes of 5 food groups
Known constraints	3 known constraints & half-space $\mathcal{C}$
Lower bound:	Protein
Upper bounds:	Calories and Carbohydrates
Unknown constraints	3 constraints

presented in Table 4.6, are determined by considering the serving size information provided in Table 4.4.

4.2.2 Results

In the FO problem, the objective function was set to minimize the total fat. We evaluate all the observations and identify the preferred observation $\hat{\mathbf{x}}^0$ with the best objective value of 39.409. Then, we proceed to infer 3 unknown constraints based on the 30 observations.

Table 4.6: The known constraints for diet recommendation problem

Description	Value(s)
Protein	$1.250\hat{x}_1^k + 0.653\hat{x}_2^k + 2.571\hat{x}_3^k + 8.651\hat{x}_4^k + 22.562\hat{x}_5^k \geq 50, \forall k \in \mathcal{K}$
Calories	$-103.851\hat{x}_1^k - 15.456\hat{x}_2^k - 115.467\hat{x}_3^k - 94.395\hat{x}_4^k - 437.166\hat{x}_5^k \geq -2400, \forall k \in \mathcal{K}$
Carbohydrate	$-26.132\hat{x}_1^k - 2.726\hat{x}_2^k - 19.455\hat{x}_3^k - 1.537\hat{x}_4^k - 28.257\hat{x}_5^k \geq -325, \forall k \in \mathcal{K}$
Half-space $\mathcal{C}$	$0.350\hat{x}_1^k + 0.294\hat{x}_2^k + 3.307\hat{x}_3^k + 5.898\hat{x}_4^k + 26.701\hat{x}_5^k \geq 39.409, \forall k \in \mathcal{K}$

Note: numbers are rounded to 3 decimal places.

We were unable to test the inference of more than 3 unknown constraints due to resource limitations. The detailed description of the inferred constraints can be found in Table 4.7. To further explore the impact of these constraints, we once solve the forward problem with only the known constraints, and then by considering both the known and the inferred constraints. The results are visualized in Figure 4.8. The red bars correspond to the recommended diet derived only based on the known constraint ("w/o IO"), while the blue bars represent the recommended diet that incorporates the additional constraints ("IO") as well. The ranges of observations for each food group are represented by error bars, and the preferred observation $\hat{\mathbf{x}}^0$ is marked with yellow circles on them. The results indicate that the diet without the inferred constraints contains higher amounts of fewer food groups. More than 5 and 65 servings of fruits and vegetables are recommended. However, the diet with both known and inferred constraints contains a reasonable amount of food groups, more in line with past observations. Moreover, the average L_1 norm distances of observations from "w/o IO" and "IO" recommended diets are 74.66 and 24.38, respectively. This further supports the notion that the "IO" recommended diet is more preferred and aligned with the dieter's taste. It is worth mentioning that, as expected, the objective value associated with the preferred observation remained unchanged after incorporating the inferred constraints

Table 4.7: The inferred constraints for diet recommendation problem

Description	Value(s)
First constraint	$0.033\hat{x}_1^k + 0.033\hat{x}_2^k + 0.147\hat{x}_3^k + 0.342\hat{x}_4^k + 0.928\hat{x}_5^k \geq 2.147, \forall k \in \mathcal{K}$
Second constraint	$0.454\hat{x}_1^k - 0.568\hat{x}_2^k - 0.250\hat{x}_3^k - 0.590\hat{x}_4^k + 0.250\hat{x}_5^k \geq -3.947, \forall k \in \mathcal{K}$
Third constraint	$-0.065\hat{x}_1^k + 0.595\hat{x}_2^k + 0.440\hat{x}_3^k - 0.181\hat{x}_4^k - 0.646\hat{x}_5^k \geq -0.904, \forall k \in \mathcal{K}$

Note: numbers are rounded to 3 decimal places.

into the FO problem and solving it. It means unknown constraints are inferred such that this solution becomes optimal.

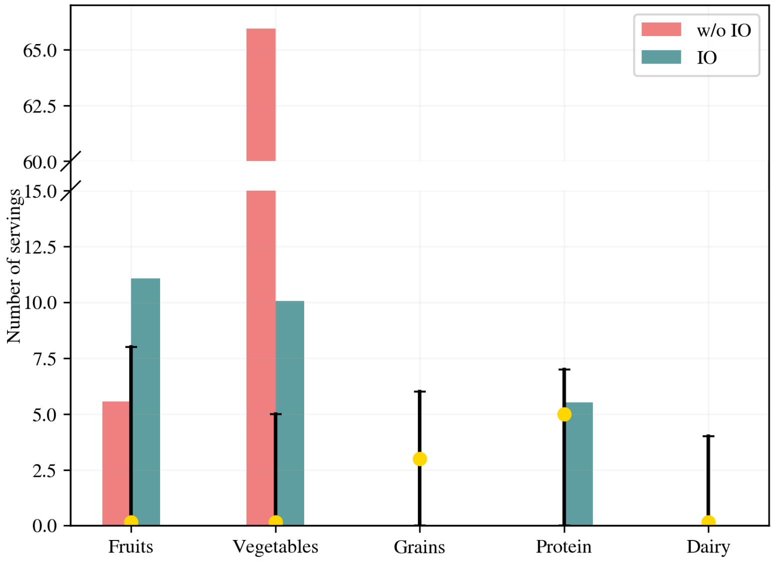


Figure 4.8: Comparing the diet recommendations with and without the inferred constraints

Chapter 5

Conclusions

In this study, we defined the fit between the model and data as the minimum Euclidean distances of the observations from the boundary of the feasible region. We introduced a goodness-of-fit measure based on the Euclidean distance, which aimed to capture the desirable properties of a good fit. Utilizing this goodness-of-fit measure as the objective function, we proposed a multi-point IO problem with maximum Euclidean fit to determine the feasible region of a linear FO problem. We also compared our model with existing inverse models that infer constraint parameters. Our findings indicated that our model outperformed the models in the literature in terms of fitting the data. We observed that the models in the literature were more likely to infer redundant constraints, whereas our model avoids incorporating redundant constraints unless there is no way to improve the fit further.

The nonlinearity introduced by the Euclidean distance made the resulting model a non-convex MIQCP problem. To enhance tractability, we applied relaxation techniques using the L_1 penalty function, which transformed the problem into a MIQP problem.

We proposed an exact heuristic method based on the identification of the convex hull to further mitigate the computational complexity. Additionally, a greedy heuristic method was proposed to improve runtime efficiency.

We applied our methodology to two example applications. The first example application consisted of laboratory report data from a hospital, which was utilized in the triage process to determine whether patients should be admitted as inpatients or discharged as outpatients. Through IO, we were able to uncover the hidden principles that influenced the decision to discharge patients from the hospital. Our model demonstrated a precision of 81.42% in accurately identifying the outpatients. Discovering such underlying principles provides a baseline for initial prediction and enhances the decision-making process for identifying outpatients in future observations. In our second application, we focused on addressing the diet recommendation problem. By analyzing the data on past food consumption, our model inferred the implicit criteria that influenced dieters' food choices. This allowed us to identify the factors contributing to their food preferences and develop diet recommendations that aimed to be not only nutritionally balanced but also more likely to be adhered to by dieters.

We also evaluated the computational performance of the solution methods using multiple instances from the laboratory report dataset. Our findings revealed that solving the IO problem resulted from the relaxation using the L_1 penalty function was highly complex. Conversely, the exact heuristic method delivered an exact optimal solution within a reasonable time frame. However, as the size of the instances increased, the complexity of the exact heuristic algorithm also grew. The greedy heuristic method, on the other hand, provided faster solutions compared to other methods, but it did not guarantee an optimal solution. Thus, there exists a trade-off between runtime and solution quality when selecting the appropriate solution method.

One important future direction is to implement this methodology on a real-world dataset in order to demonstrate the computational advantages of the proposed model and solution methods. Additionally, another potential direction for future research involves exploring the incorporation of infeasible observations into the forward problem.

References

Farzin Ahmadi, Fardin Ganjkhanloo, and Kimia Ghobadi. Inverse learning: A data-driven framework to infer optimizations models. *arXiv preprint arXiv:2011.03038*, 2020a.

Farzin Ahmadi, Fardin Ganjkhanloo, and Kimia Ghobadi. An open-source dataset on dietary behaviors and dash eating plan optimization constraints. *arXiv preprint arXiv:2010.07531*, 2020b.

Ravindra K Ahuja and James B Orlin. Inverse optimization. *Operations Research*, 49(5): 771–783, 2001.

Temitayo Ajayi, Taewoo Lee, and Andrew J Schaefer. Objective selection for cancer treatment: an inverse optimization approach. *Operations Research*, 70(3):1717–1738, 2022.

ANM Alam, Rimi Reza, Asir Abrar, Tanvir Ahmed, Salsabil Ahmed, Shihab Sharar, and Annajiat Alim Rasel. Patients' severity states classification based on electronic health record (ehr) data using multiple machine learning and deep learning approaches. *arXiv preprint arXiv:2209.14907*, 2022.

Anil Aswani, Zuo-Jun Shen, and Auyon Siddiq. Inverse optimization with noisy data. *Operations Research*, 66(3):870–892, 2018.

Aaron Babier, Justin J Boutilier, Michael B Sharpe, Andrea L McNiven, and Timothy CY Chan. Inverse optimization of objective function weights for treatment planning using clinical dose-volume histograms. *Physics in Medicine & Biology*, 63(10):105004, 2018.

Aaron Babier, Timothy CY Chan, Taewoo Lee, Rafid Mahmood, and Daria Terekhov. An ensemble learning framework for model fitting and evaluation in inverse linear optimization. *Informs Journal on Optimization*, 3(2):119–138, 2021.

Amir Beck. *First-order methods in optimization*. SIAM, 2017.

Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch Paschalidis. Inverse optimization: A new perspective on the black-litterman model. *Operations research*, 60(6):1389–1403, 2012.

Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch Paschalidis. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153:595–633, 2015.

John R Birge, Ali Hortaçsu, and J Michael Pavlin. Inverse optimization for the recovery of market structure from market outcomes: An application to the miso electricity market. *Operations Research*, 65(4):837–855, 2017.

Merve Bodur, Timothy CY Chan, and Ian Yihang Zhu. Inverse mixed integer optimization: Polyhedral insights and trust region methods. *INFORMS Journal on Computing*, 2022.

Justin J Boutilier, Taewoo Lee, Tim Craig, Michael B Sharpe, and Timothy CY Chan. Models for predicting objective function weights in prostate cancer imrt. *Medical physics*, 42(4):1586–1595, 2015.

Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005, 2003.

Jason Brownlee. How to choose a feature selection method for machine learning, 2020. URL https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/#:~:text=Feature%20selection%20is%20the%20process,the%20performance%20of%20the%20model.

Aykut Bulut and Ted K Ralphs. On the complexity of inverse mixed integer linear optimization. *SIAM Journal on Optimization*, 31(4):3014–3043, 2021.

Richard H Byrd, Jorge Nocedal, and Richard A Waltz. Steering exact penalty methods for nonlinear programming. *Optimization Methods and Software*, 23(2):197–213, 2008.

Michal Černỳ and Milan Hladík. Inverse optimization: towards the optimal parameter set of inverse lp with interval coefficients. *Central European Journal of Operations Research*, 24(3):747–762, 2016.

Timothy C. Y. Chan, Maria Eberg, Katharina Forster, Claire Holloway, Luciano Ieraci, Yusuf Shalaby, and Nasrin Yousefi. An inverse optimization approach to measuring clinical pathway concordance, 2021a.

Timothy CY Chan and Neal Kaw. Inverse optimization for the recovery of constraint parameters. *European Journal of Operational Research*, 282(2):415–427, 2020.

Timothy CY Chan and Taewoo Lee. Trade-off preservation in inverse multi-objective convex optimization. *European Journal of Operational Research*, 270(1):25–39, 2018.

Timothy CY Chan, Tim Craig, Taewoo Lee, and Michael B Sharpe. Generalized inverse multiobjective optimization with application to cancer therapy. *Operations Research*, 62 (3):680–695, 2014.

Timothy CY Chan, Taewoo Lee, and Daria Terekhov. Inverse optimization: Closed-form solutions, geometry, and goodness of fit. *Management Science*, 65(3):1115–1135, 2019.

Timothy CY Chan, Rafid Mahmood, and Ian Yihang Zhu. Inverse optimization: Theory and applications. *arXiv preprint arXiv:2109.03920*, 2021b.

Lu Chen, Yuyi Chen, and André Langevin. An inverse optimization approach for a capacitated vehicle routing problem. *European Journal of Operational Research*, 295(3): 1087–1098, 2021.

Joseph YJ Chow and Will W Recker. Inverse optimization with endogenous arrival time constraints to calibrate the household activity pattern problem. *Transportation Research Part B: Methodological*, 46(3):463–479, 2012.

Joseph YJ Chow, Stephen G Ritchie, and Kyungsoo Jeong. Nonlinear inverse optimization for parameter estimation of commodity-vehicle-decoupled freight assignment. *Transportation Research Part E: Logistics and Transportation Review*, 67:71–91, 2014.

Stephan Dempe and Sebastian Lohse. Inverse linear programming. In *Recent advances in optimization*, pages 19–28. Springer, 2006.

Chaosheng Dong and Bo Zeng. Inferring parameters through inverse multiobjective optimization. *arXiv preprint arXiv:1808.00935*, 2018.

Chaosheng Dong, Yiran Chen, and Bo Zeng. Generalized inverse optimization through online learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Ricardo Fernández-Blanco, Juan Miguel Morales, Salvador Pineda, and Alvaro Porras. Ev-fleet power forecasting via kernel-based inverse optimization. *arXiv preprint arXiv:1908.00399*, 2019.

Ricardo Fernández-Blanco, Juan Miguel Morales, and Salvador Pineda. Forecasting the price-response of a pool of buildings via homothetic inverse optimization. *Applied Energy*, 290:116791, 2021.

Bennet Gebken and Sebastian Peitz. Inverse multiobjective optimization: Inferring decision criteria from data. *Journal of Global Optimization*, 80:3–29, 2021.

Archis Ghate. Inverse optimization in countably infinite linear programs. *Operations Research Letters*, 43(3):231–235, 2015.

Archis Ghate. Imputing radiobiological parameters of the linear-quadratic dose-response model from a radiotherapy fractionation plan. *Physics in Medicine & Biology*, 65(22): 225009, 2020a.

Archis Ghate. Inverse optimization in semi-infinite linear programs. *Operations Research Letters*, 48(3):278–285, 2020b.

Kimia Ghobadi and Houra Mahmoudzadeh. Inferring linear feasible regions using inverse optimization. *European Journal of Operational Research*, 290(3):829–843, 2021.

Kimia Ghobadi, Taewoo Lee, Houra Mahmoudzadeh, and Daria Terekhov. Robust inverse optimization. *Operations Research Letters*, 46(3):339–344, 2018.

Ali Goli, Justin J Boutilier, Tim Craig, Michael B Sharpe, and Timothy CY Chan. A small number of objective function weight vectors is sufficient for automated treatment planning in prostate cancer. *Physics in Medicine & Biology*, 63(19):195004, 2018.

Government of Canada. Nutrient value of some common foods, 2008. URL https://www.canada.ca/en/health-canada/services/food-nutrition/healthy-eating/nutrient-data/nutrient-value-some-common-foods-2008.html.

Çiğdem Güler and Horst W Hamacher. Capacity inverse minimum cost flow problem. *Journal of Combinatorial Optimization*, 19(1):43–59, 2010.

Garud Iyengar and Wanmo Kang. Inverse conic programming with applications. *Operations Research Letters*, 33(3):319–330, 2005.

Arezou Keshavarz, Yang Wang, and Stephen Boyd. Imputing a convex objective function. In *2011 IEEE international symposium on intelligent control*, pages 613–619. IEEE, 2011.

Tae Kyun Kim. Understanding one-way anova using conceptual figures. *Korean journal of anesthesiology*, 70(1):22–26, 2017.

Jourdain B Lamperski and Andrew J Schaefer. A polyhedral characterization of the inverse-feasible region of a mixed-integer program. *Operations Research Letters*, 43(6):575–578, 2015.

Tianguang Lu, Zhaoyu Wang, Jianhui Wang, Qian Ai, and Chong Wang. A data-driven stackelberg market strategy for demand response-enabled distribution systems. *IEEE Transactions on Smart Grid*, 10(3):2345–2357, 2018.

Houra Mahmoudzadeh and Kimia Ghobadi. Learning from good and bad decisions: A data-driven inverse optimization approach. *arXiv preprint arXiv:2207.02894*, 2022.

Mahsa Moghaddass and Daria Terekhov. Inverse integer optimization with multiple observations. *Optimization Letters*, 15(4):1061–1079, 2021.

Peyman Mohajerin Esfahani, Soroosh Shafieezadeh-Abadeh, Grani A Hanasusanto, and Daniel Kuhn. Data-driven inverse optimization with imperfect information. *Mathematical Programming*, 167:191–234, 2018.

Mostafa Naghavi, Ali Asghar Foroughi, and Masoud Zarepisheh. Inverse optimization for multi-objective linear programming. *Optimization letters*, 13(2):281–294, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Mujiono Sadikin, Ida Nurhaida, and Ria Puspita Sari. Exploratory study of some machine learning techniques to classify the patient treatment. *International Journal of Advanced Computer Science and Applications*, 12(2), 2021.

Javier Saez-Gallego and Juan M Morales. Short-term forecasting of price-responsive loads using inverse optimization. *IEEE Transactions on Smart Grid*, 9(5):4805–4814, 2017.

GA Sayre and D Ruan. Automatic treatment planning with convex imputing. In *Journal of Physics: Conference Series*, volume 489, page 012058. IOP Publishing, 2014.

Andrew J Schaefer. Inverse integer programming. *Optimization Letters*, 3:483–489, 2009.

Zahed Shahmoradi and Taewoo Lee. Optimality-based clustering: An inverse optimization approach. *Operations Research Letters*, 50(2):205–212, 2022a.

Zahed Shahmoradi and Taewoo Lee. Quantile inverse optimization: Improving stability in inverse linear programming. *Operations research*, 70(4):2538–2562, 2022b.

Onur Tavaslıoğlu, Taewoo Lee, Silviya Valeva, and Andrew J Schaefer. On the structure of the inverse-feasible region of a linear program. *Operations Research Letters*, 46(1): 147–152, 2018.

U.S. Department of Health and Human Services and U.S. Department of Agriculture. 2020–2025 dietary guidelines for americans, 2020. URL https://www.dietaryguidelines.gov/sites/default/files/2021-03/Dietary_Guidelines_for_Americans-2020-2025.pdf.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Lizhi Wang. Cutting plane algorithms for the inverse mixed integer linear programming problem. *Operations research letters*, 37(2):114–116, 2009.

Meihua Wang, Fengmin Xu, and Guan Wang. Sparse portfolio rebalancing model based on inverse optimization. *Optimization Methods and Software*, 29(2):297–309, 2014.

Keji Wei and Vikrant Vaze. Modeling crew itineraries and delays in the national air transportation system. *Transportation Science*, 52(5):1276–1296, 2018.

Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 35(67-68): 7, 1999.

Susan Jia Xu, Mehdi Nourinejad, Xuebo Lai, and Joseph YJ Chow. Network learning via multiagent inverse transportation problems. *Transportation Science*, 52(6):1347–1364, 2018.

Shi Yu, Haoran Wang, and Chaosheng Dong. Learning risk preferences from investment portfolios using inverse optimization. *Research in International Business and Finance*, page 101879, 2023.

Jianzhong Zhang and Zhenhong Liu. Calculating some inverse linear programming problems. *Journal of Computational and Applied Mathematics*, 72(2):261–273, 1996.

Jianzhong Zhang and Zhenhong Liu. A further study on inverse linear programming problems. *Journal of Computational and Applied Mathematics*, 106(2):345–359, 1999.

Jianzhong Zhang and Chengxian Xu. Inverse optimization for linearly constrained convex separable programming problems. *European Journal of Operational Research*, 200(3): 671–679, 2010.

Jianzhong Zhang and Liwei Zhang. An augmented lagrangian method for a class of inverse quadratic programming problems. *Applied Mathematics and Optimization*, 61(1):57–83, 2010.

Jianzhong Zhang, Zhongfan Ma, and Chao Yang. A column generation method for inverse shortest path problems. *Zeitschrift für Operations Research*, 41(3):347–358, 1995.

Jianzhong Zhang, Liwei Zhang, and Xiantao Xiao. A perturbation approach for an inverse quadratic programming problem. *Mathematical Methods of Operations Research*, 72(3): 379–404, 2010.

Yi Zhang, Liwei Zhang, Jia Wu, and Jianzhong Zhang. A perturbation approach for an inverse quadratic programming problem over second-order cones. *Mathematics of Computation*, 84(291):209–236, 2015.

APPENDICES

Appendix A

Lagrangian function

A.1 Lagrangian function

In this section, we show that the lagrangian relaxation method does not work for our problem as it leads to zero solutions. Suppose that we apply lagrangian relaxation to the inverse optimization problem (3.13) to find a lower bound to the model. By relaxing constraint (3.13d), we get:

$$\min_{\mathbf{A},\mathbf{b},\mathbf{m}} \quad L = \frac{\sum_{k\in\mathcal{K}} m^k}{|\mathcal{K}|} + \sum_{i\in\mathcal{I}_1} \mu_i (\sum_{j\in\mathcal{J}} a_{ij}^2 - 1) \tag{A.1a}$$

$$\text{s.t.} \quad (3.13b), (3.13c), (3.13e) \tag{A.1b}$$

The lagrangian lower bound is derived by solving the following lagrangian dual problem:

$$Z_L = \max_{\boldsymbol{\mu}} \min_{\mathbf{A},\mathbf{b},\mathbf{m}} L \tag{A.2}$$

Proposition 4. *Problem (A.1) is unbounded if $\boldsymbol{\mu} < 0$.*

Proof. Observe that $\mathbf{A}, \mathbf{b} \in \mathbb{R} \cup \{-\infty, +\infty\}$. If $\boldsymbol{\mu} < 0$, $\exists \mathbf{A}, \mathbf{b}$ such that $\sum_{i \in \mathcal{I}_1} \mu_i (\sum_{j \in \mathcal{J}} a_{ij}^2 - 1) \to -\infty$ and $\epsilon_i^k < +\infty$, $\forall k \in \mathcal{K}, i \in \mathcal{I}_1$. Also, $\epsilon_i^k < +\infty$, $\forall k \in \mathcal{K}, i \in \mathcal{I}_1$ implies $\frac{\sum_{k \in \mathcal{K}} m^k}{|\mathcal{K}|} < +\infty$, $\forall k \in \mathcal{K}$. This means function $L \to -\infty$. □

Proposition 5. *The solution of problem (A.1) is zero if* $\boldsymbol{\mu} \geq 0$.

Proof. The optimization model (A.1) tries to minimize function L over some constraints. Observe that the first term of L, $\frac{\sum_{k \in \mathcal{K}} m^k}{|\mathcal{K}|}$, is always non-negative as $m^k \geq 0$. If $\boldsymbol{\mu} \geq 0$, $\exists \mathbf{A}, \mathbf{b} = 0$ such that $\epsilon_i^k = 0$, $\forall k \in \mathcal{K}, i \in \mathcal{I}_1$. Therefore, $\frac{\sum_{k \in \mathcal{K}} m^k}{|\mathcal{K}|} = 0$ and $\sum_{i \in \mathcal{I}_1} \mu_i (\sum_{j \in \mathcal{J}} a_{ij}^2 - 1) = -\sum_{i \in \mathcal{I}_1} \mu_i$.

□