Expert Learning through Generalized Inverse Multiobjective Optimization: Models, Insights, and Algorithms

Chaosheng Dong [1] **Bo Zeng** [1]

Abstract

We consider a new unsupervised learning task of inferring parameters of a multiobjective decision making model, based on a set of observed decisions from the human expert. This setting is important in applications (such as the task of portfolio management) where it may be difficult to obtain the human expert's intrinsic decision making model. We formulate such a learning problem as an inverse multiobjective optimization problem (IMOP) and propose its first sophisticated model with statistical guarantees. Then, we reveal several fundamental connections between IMOP, K-means clustering, and manifold learning. Leveraging these critical insights and connections, we propose two algorithms to solve IMOP through manifold learning and clustering. Numerical results confirm the effectiveness of our model and the computational efficacy of algorithms.

1. Introduction

Human decision makers are often confronted with multiple objectives when making decisions. For example, comfort and cost are two often conflicting criteria when customers make purchases (Greco et al., 2016). Actually, in economics, science, and engineering, the multiobjective decision making problem (DMP) is quite common and many decision making processes naturally involve multiple conflicting objectives (Hwang & Masud, 2012). These underlying multiobjective decision making schemes, once learned by artificial intelligence (AI) system, would presumably assist and accelerate the human expert's decision making process, such as supporting organizations in designing products or in providing services to customers.

Nevertheless, as in most scenarios, one can only observe the human decision maker's decisions while cannot directly access her underlying decision making schemes. Consider the recommender system used by retailers to support customers for online shopping. The customer's historical purchasing behaviors (Aggarwal et al.) are known to the system. To make personalized recommendations, it is especially crucial for the system to quickly learn the customer's needs and desires from these observed decisions.

In this paper, we investigate learning the parameters, e.g., objectives functions or constraints, from an expert by observing her decisions. Specifically, the learner observes a set of decisions $\{\mathbf{y}_i\}_{i\in[N]}$ and each $\mathbf{y}_i$ with $i \in [N]$ is an observation of the Pareto optimal solution of the multiobjective optimization problem (MOP):

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & \{f_1(\mathbf{x},\theta), f_2(\mathbf{x},\theta), \ldots, f_p(\mathbf{x},\theta)\} \\ s.t. & \mathbf{x} \in X(\theta), \end{array}$$

where θ is the true but unknown parameter for the expert's multiobjective decision making problem. Formally, we seek to answer the following fundamental question

how do we learn θ given $\{\mathbf{y}_i\}_{i\in[N]}$?

This question naturally occurs in many settings. For example, a portfolio manager typically uses the Markovitz mean-variance model to make investment decisions (Markowitz, 1952). One analyst might be interested in learning the key parameter of this model, e.g., the expected returns of the assets, by observing the portfolio manager's historical investment records. To learn the parameter θ, we formulate an inverse multiobjective optimization problem (IMOP), assuming that the human expert is rational or bounded rational. Given the fact that the learner often only has access to human expert's decisions and no other data, we show that IMOP essentially is an unsupervised learning task.

Subsequently, we reveal connections between IMOP and two seemingly unrelated unsupervised learning problems. The first one is the K-means clustering problem (MacQueen, 1967; Lloyd, 1982; Arthur & Vassilvitskii, 2007). Specifically, we prove that every K-means clustering problem can be transformed equivalently to an IMOP. As solving K-means clustering problem is NP-hard (Aloise et al.,

[1]Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, USA. Correspondence to: Chaosheng Dong and Bo Zeng <chaosheng@pitt.edu, bzeng@pitt.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

Table 1. Comparisons between IOP and IMOP from the machine learning point of view.

Task	# of obj	Signal	Supervised learning	Paper
IOP	single	yes	yes	(Keshavarz et al., 2011; Bertsimas et al., 2015; Aswani et al., 2018) (Esfahani et al., 2018; Bärmann et al., 2017; Dong et al., 2018)
IMOP	multiple	no	no	this paper

2009; Mahajan et al., 2012) even when $n = 2$, we thus show the NP-hardness of IMOP. Furthermore, we note that solving IMOP will automatically assign the observations to different clusters, while the centroids of these clusters are restricted to be Pareto optimal solutions of the estimated DMP. Hence, IMOP can be interpreted as a constrained K-means clustering problem (Wagstaff et al., 2001).

The second one is the manifold learning problem, which seeks to construct low-dimensional manifolds from data points embedded in high-dimensional spaces (Roweis & Saul, 2000; Tenenbaum et al., 2000). We note that the Pareto optimal set is a continuous manifold with an intrinsic dimension of $p - 1$ (p is the number of objective functions) under suitable smoothness conditions, regardless of the dimension of the decision space. Since the dimension of the decision space is usually much larger than p, the Pareto optimal set is a low-dimensional manifold embedded in the ambient decision space. Moreover, given that solving IMOP is to construct a DMP whose Pareto optimal set closely matches observations, IMOP can also be interpreted as a manifold learning problem.

Related works Inverse mutiobjective optimization is rather new and rarely investigated. Most existing works merely focus on the optimization perspective. (Roland et al., 2013) considers a binary integer DMP with a set of linear objective functions. They develop algorithms to find minimal adjustment of the objective functions such that a given set of feasible solutions becomes Pareto optimal. (Chan et al., 2014; Chan & Lee, 2018) study another situation where preferences or weights of several known (linear) criteria in the DMP will be inferred based on a single noiseless observation. Different from those studies, our study follows the data-driven approach and take a learning perspective to build an IMOP framework that directly considers many noisy observations to infer multiple objective functions or constraints of a convex DMP from an expert with a solid statistical significance.

Our work is most related to the inverse optimization problem (IOP), in which the decision making problem has only one objective function (Ahuja & Orlin, 2001). IOP has been extensively investigated (Ahuja & Orlin, 2001; Iyengar & Kang, 2005; Schaefer, 2009; Wang, 2009). Note that data for IOP typically consists of clear signal-response pairs (Keshavarz et al., 2011; Bertsimas et al., 2015; Aswani et al., 2018; Esfahani et al., 2018; Bärmann et al., 2017; Dong et al., 2018), where the signal is the input data and the response is the output. Consequently, IOP essentially can be seen as a supervised learning problem. In contrast, the IMOP we study is of unsupervised learning type because the data available to the learner is only the decision makers' decisions, not including any information on their preferences used to generate these decisions. In other words, the weight information associated with the decision is a "hidden variable". This fundamental difference highlights the main reason why IMOP is much more complex than IOP. In this paper, an unconventional framework for IMOP is developed to address this challenge. We summarize the comparisons between IOP and IMOP in Table 1.

Another relevant line of research is the inverse reinforcement learning (IRL) which seeks to extract the reward function given observed optimal behavior (Ng & Russell, 2000; Abbeel & Ng, 2004). In this case, the underlying decision making problem is a Markov decision process instead of a multiobjective optimization making problem. There have been many exciting contributions to the IRL literature since the seminar work (Ng & Russell, 2000). Some standouts are (Ratliff et al., 2006; Ramachandran & Amir; Ziebart et al., 2008; Hadfield-Menell et al., 2016; Pirotta & Restelli, 2016; Finn et al., 2016; Amin et al., 2017; Metelli et al., 2017), which are developed to handle its limitations. Others are (Abbeel & Ng, 2004; Syed et al., 2008; Ziebart et al., 2008; Ratliff et al., 2009; Ho et al., 2016) which are applied to more complex and realistic situations in designing AI systems and modeling nature learning.

Our contributions Our contributions for expert learning through IMOP are threefold: models, insights, and algorithms. To the best of the authors' knowledge, we propose the first general model for the unsupervised learning task of inferring parameters of the multiobjective decision making model, based on a set of observed decisions from the human expert. This model can learn general convex objective functions or constraints of the expert's multiobjective decision making problem from observed decisions. Moreover, we provide extensive statistical analysis of our model, and also show its generalization to the unseen data. Furthermore, we reveal hidden connections between IMOP, K-means clustering, and manifold learning. Leveraging these connections and insights, we propose two algorithms to solve IMOP through manifold learning and clustering. Numerical results on both synthetic and real datasets confirm the effectiveness of our model and the computational efficacy of algorithms to attack large-scale IMOP.

2. Problem Setting

2.1. Multiobjective Decision Making Problem

Consider the following parameterized decision making problem with $p\ (\geq 2)$ objective functions,

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & \{f_1(\mathbf{x},\theta), f_2(\mathbf{x},\theta),\ldots,f_p(\mathbf{x},\theta)\} \\ s.t. & \mathbf{x}\in X(\theta), \end{array} \qquad \text{DMP}$$

where θ is the parameter for the expert's multiobjective decision making problem. For easy exposition, we use $\mathbf{f}(\mathbf{x},\theta)$ to denote the vector valued function $(f_1(\mathbf{x},\theta), f_2(\mathbf{x},\theta),\ldots,f_p(\mathbf{x},\theta))^T$. Also, the feasible set $X(\theta)$ is characterized as $X(\theta)=\{\mathbf{x}\in\mathbb{R}^n : \mathbf{g}(\mathbf{x},\theta)\leq \mathbf{0}\}$, where $\mathbf{g}(\mathbf{x},\theta)=(g_1(\mathbf{x},\theta),\ldots,g_q(\mathbf{x},\theta))^T$ is another vector-valued function.

In this paper, we focus on a convex DMP where all objective functions and constraints are convex in $\mathbf{x}$ for each θ.

Definition 2.1 (Pareto optimality)**.** For a fixed θ, a decision $\mathbf{x}^*\in X(\theta)$ is said to be Pareto optimal if there exists no other decision $\mathbf{x}\in X(\theta)$ such that $f_i(\mathbf{x},\theta)\leq f_i(\mathbf{x}^*,\theta)$ for all $i\in[p]$, and $f_j(\mathbf{x},\theta)< f_j(\mathbf{x}^*,\theta)$ for at least one $j\in[p]$.

In the study of multiobjective optimization, the set of all Pareto optimal solutions is denoted by $X_P(\theta)$ and called the Pareto optimal set. One common way to derive a Pareto optimal solution is to solve a problem with a single objective function constructed by the weighted sum of original objective functions (Gass & Saaty, 1955), i.e.,

$$\begin{array}{ll} \min & w^T\mathbf{f}(\mathbf{x},\theta) \\ s.t. & \mathbf{x}\in X(\theta), \end{array} \qquad \text{WP}$$

where $w=(w^1,\ldots,w^p)^T$ is the nonnegative weight vector in the $(p-1)$-simplex $\mathscr{W}_p\equiv\{w\in\mathbb{R}^p_+ :\ \mathbf{1}^Tw=1\}$. When all weight components are required to be positive, such set is denoted by $\mathscr{W}_p^+$. Denote $S(w,\theta)$ the set of optimal solutions of WP, i.e.,

$$S(w,\theta)=\arg\min_{\mathbf{x}}\left\{w^T\mathbf{f}(\mathbf{x},\theta) : \mathbf{x}\in X(\theta)\right\}.$$

Then, we have the next theoretical results of WP following from (Miettinen, 2012).

Proposition 2.1. For convex DMP, $\bigcup\limits_{w\in\mathscr{W}_p^+} S(w,\theta)\subseteq X_P(\theta)\subseteq \bigcup\limits_{w\in\mathscr{W}_p} S(w,\theta)$.

Remark 2.1. Results in Proposition 2.1 provide us a theoretical basis to make use of the weighted sum method to derive all Pareto optimal solutions. Actually, when DMP is convex and the objective functions are strictly convex, we have $X_P(\theta)=\bigcup\limits_{w\in\mathscr{W}_p} S(w,\theta)$.

2.2. Inverse Multiobjective Optimization

In this section, given a set of observed decisions that are noisy Pareto optimal solutions collected from an expert, we construct an inverse multiobjective optimization model to infer the parameter θ in DMP.

Let $\mathbf{y}$ denote one observed decision from an expert that is distributed according to an unknown distribution $\mathbb{P}_\mathbf{y}$ and supported on $\mathcal{Y}$. We first construct our loss function with respect to a hypothesis θ. Note that if weights over objective functions, i.e., the weight vector w, are known, IMOP degenerates into IOP, and the conventional loss function in (Aswani et al., 2018; Dong et al., 2018) can be directly applied with respect to $\mathbf{y}$ and $S(w,\theta)$. Nevertheless, w is often missing and the Pareto optimal set should be adopted instead as follows

$$l(\mathbf{y},\theta)=\min_{\mathbf{x}\in X_P(\theta)}\|\mathbf{y}-\mathbf{x}\|_2^2, \tag{1}$$

where $X_P(\theta)$ denotes the Pareto optimal set of DMP for a given θ. Then, our inverse multiobjective optimization problem can be formulated with its distribution $\mathbb{P}_\mathbf{y}$ as

$$\min_{\theta\in\Theta}\quad M(\theta)\equiv\mathbb{E}_{\mathbf{y}\sim\mathbb{P}_\mathbf{y}}\Big(l(\mathbf{y},\theta)\Big), \qquad \text{IMOP}$$

where $M(\theta)$ is also called the risk of the loss $l(\mathbf{y},\theta)$ for the hypothesis θ.

Nevertheless, one challenge of using (1) is that there is no general approach to comprehensively and explicitly characterize Pareto optimal set $X_P(\theta)$. According to Proposition 2.1, we adopt a sampling approach to generate a set of $\{w_k\}_{k\in[K]}\in\mathscr{W}_p$ and approximate $X_P(\theta)$ as the union of $\{S(w_k,\theta)\}_{k\in[K]}$. That is,

$$\bigcup_{k\in[K]} S(w_k,\theta)\approx X_P(\theta).$$

Then, we get the *surrogate loss function* in the following:

$$\begin{aligned} l_K(\mathbf{y},\theta)= & \min_{\mathbf{x}_k, z_k\in\{0,1\}} \|\mathbf{y}-\sum_{k\in[K]} z_k\mathbf{x}_k\|_2^2 \\ & \text{s.t.} \sum_{k\in[K]} z_k=1,\ \mathbf{x}_k\in S(w_k,\theta). \end{aligned} \tag{2}$$

Remark 2.2. (i) Constraint $\sum\limits_{k\in[K]} z_k=1$ ensures that only one of Pareto optimal solutions will be selected to measure the distance to the expert decision $\mathbf{y}$. Hence, solving this problem identifies some w_k with $k\in[K]$ such that one Pareto optimal solution from $S(w_k,\theta)$ is closest to $\mathbf{y}$. (ii) As it is practically infeasible to sample all weight vectors in $\mathscr{W}_p$, we can control K to achieve a desired tradeoff between approximation accuracy and computational efficacy. In practice, for general convex DMP, we evenly sample

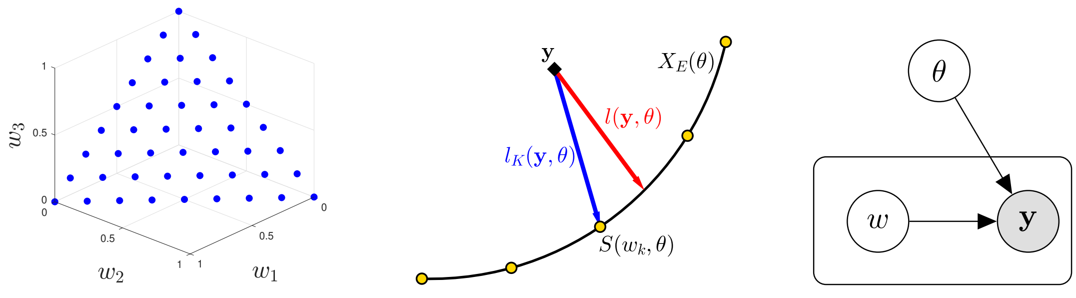


Figure 1. (a) Blue dots indicate the evenly sampled weights from the 3-dimensional simplex $\mathscr{W}_3$. Here, $w_1 + w_2 + w_3 = 1$. (b) Yellow dots are the Pareto optimal solutions sampled from $X_P(\theta)$. Red and blue arrows indicate $l(\mathbf{y}, \theta)$ and $l_K(\mathbf{y}, \theta)$, respectively. (c) Graphical plate model of IMOP. The unshaded nodes are hidden variables and the shaded nodes represent observed variables.

$\{w_k\}_{k\in[K]}$ from $\mathscr{W}_p^+$ to ensure that $S(w_k, \theta) \in X_P(\theta)$. If $\mathbf{f}(\mathbf{x}, \theta)$ is known to be strictly convex, we can evenly sample $\{w_k\}_{k\in[K]}$ from $\mathscr{W}_p$ as $S(w_k, \theta) \in X_P(\theta)$ by Proposition 2.1. As an example, we show the evenly sampled weights when $p = 3$ in Figure 1a.

Moreover, as $\mathbb{P}_\mathbf{y}$ is not known a priori, we next provide an empirical formulation of IMOP. Specifically, given observations $\{\mathbf{y}_i\}_{i\in[N]}$ from an expert, θ will instead be inferred through solving the following empirical risk minimizing problem with weight samples $\{w_k\}_{k\in[K]}$ (i.e., using (2)):

$$
\begin{array}{lll}
\min\limits_{\theta\in\Theta} & M_K^N(\theta) \equiv \frac{1}{N}\sum\limits_{i\in[N]} \|\mathbf{y}_i - \sum\limits_{k\in[K]} z_{ik}\mathbf{x}_k\|_2^2, & \\
\text{s.t.} & \mathbf{x}_k \in S(w_k, \theta), & \forall k \in [K], \\
& \sum\limits_{k\in[K]} z_{ik} = 1, & \forall i \in [N], \\
& z_{ik} \in \{0, 1\}, & \forall i \in [N],\ k \in [K],
\end{array}
\tag{3}
$$

where $S(w_k, \theta)$ can be replaced by optimality conditions, e.g., KKT conditions, in computation.

Remark 2.3. Differences between (1) and (2) are illustrated in Figure 1b. The convergence rate of $l_K(\mathbf{y}, \theta)$ to $l(\mathbf{y}, \theta)$ is of $\mathcal{O}(1/K^{\frac{1}{p-1}})$. Details are provided in the supplementary material. As p increases, we might require (approximately) exponentially more weight samples $\{w_K\}_{k\in[K]}$ to achieve a certain approximation accuracy. In fact, this phenomenon is a reflection of *curse of dimensionality* (Hastie et al., 2001), a principle that estimation of the parameter becomes exponentially harder as the number of dimension increases.

Remark 2.4. We highlight that the learner only has access to the expert's decisions without any weight information. In fact, this is often the case in real world applications. For example, it is impractical for the analyst to have access to the portfolio manager's preference on each asset when learning the expected returns of the assets. Therefore, IMOP is an unsupervised learning task and the goal is to recover the structure of the Pareto optimal set from which these decisions are generated. However, this does not mean that the weight w disappears in our setting. In contrast, w appears in IMOP as a hidden variable and generates the decision $\mathbf{y}$ together with θ as shown in Figure 1c. Just like any other machine learning tasks involving hidden variables (Dempster et al., 1977), we also need to learn w in order to infer the model parameter θ.

3. Consistency and generalization bound

3.1. Consistency of (3)

We show that the estimator obtained by solving (3) asymptotically predict as well as the best possible result that this type of model can achieve. Specifically, we demonstrate that the estimator is risk consistent.

Before proving the consistency of the estimator, we first need to prove the uniform convergence of the empirical risk. Next, a few assumptions are typically adopted to define a friendly structure.

Assumption 3.1. **(i)** The parameter set Θ is convex and compact. **(ii)** For each $\theta \in \Theta$, $X(\theta)$ is compact, and has a nonempty relatively interior. $X(\theta)$ is also uniformly bounded, that is, there exists $B > 0$ such that $\|\mathbf{x}\|_2 \leq B$ for all $\mathbf{x} \in X(\theta)$ and $\theta \in \Theta$. **(iii)** Functions $\mathbf{f}(\mathbf{x}, \theta)$ and $\mathbf{g}(\mathbf{x}, \theta)$ are continuous on $\mathbb{R}^n \times \Theta$, and convex in $\mathbf{x}$ for each $\theta \in \Theta$. **(iv)** The support $\mathcal{Y}$ of the decisions $\mathbf{y}$ is contained within a ball of radius R almost surely, where $R < \infty$.

Those assumptions are practically mild and often adopted in IOP literature, e.g., (Aswani et al., 2018; Dong et al., 2018). We note that (ii) and (iii) are important for the continuity of $X_E(\theta)$. Also, (iv), which is ensured once variance of the noise is finite, is fundamental to apply *the uniform law of large numbers* (ULLN) (Jennrich, 1969), the most common tools in performing consistency analysis.

We next introduce one definition to support the convergence proof with respect to both N and K.

Definition 3.1 (Double index convergence)**.** Let $\{X_{mn}\}$ be an array of double-index random variables. Let X be a random variable. If $\forall \delta > 0, \forall \epsilon > 0, \exists N$, s.t. $\forall m, n \geq N$, $\mathbb{P}(|X_{mn} - X| > \epsilon) < \delta$. Then X_{mn} is said to converge in probability to X (denoted by $X_{mn} \xrightarrow{P} X$).

Proposition 3.1. Suppose Assumption 3.1 holds. If $\mathbf{f}(\mathbf{x}, \theta)$ is strongly convex in $\mathbf{x}$ for each $\theta \in \Theta$, then $M_K^N(\theta)$ uniformly converges to $M(\theta)$ in N and K for $\theta \in \Theta$. That is,

$$\sup_{\theta \in \Theta} |M_K^N(\theta) - M(\theta)| \xrightarrow{P} 0.$$

Denote Θ^* the set of parameters that minimize the risk and refer to it as the optimal set, i.e., $\Theta^* = \{\theta^* \in \Theta : M(\theta^*) = \min_{\theta \in \Theta} M(\theta)\}$. Denote $\hat{\theta}_K^N$ the optimal solution for (3). We are now ready to state the result of risk consistency.

Theorem 3.2 (Consistency of (3))**.** Under the same assumptions as Proposition 3.1, $M(\hat{\theta}_K^N) \xrightarrow{P} M(\theta^*)$.

The above result implies that $\hat{\theta}_K^N$ will asymptotically predict as well as the best possible result that this type of model can achieve.

3.2. Generalization bound for (3)

Let $M_K(\theta)$ be the risk for the hypothesis θ using (2). For fixed weight samples $\{w_k\}_{k \in [K]}$, we really want to estimate the risk $M_K(\hat{\theta}_K^N)$ as it quantifies how well the performance of our estimator $\hat{\theta}_K^N$ generalizes to the unseen data. However, this quantity cannot be obtained since the distribution $\mathbb{P}_{\mathbf{y}}$ is unknown, and thus is a random variable. One way to make a statement about this quantity is to say how it relates to an estimate such as the empirical risk $M_K^N(\hat{\theta}_K^N)$.

Theorem 3.3. Suppose Assumption 3.1 holds. For any $0 < \delta < 1$, with probability at least $1 - \delta$, for each K

$$\begin{aligned} & M_K(\hat{\theta}_K^N) \leq M_K^N(\hat{\theta}_K^N) \\ & + \frac{1}{\sqrt{N}} \left(2K(B^2 + 2BR) + (B + R)^2 \sqrt{\log(1/\delta)/2} \right). \end{aligned}$$

The key step in proving the theorem is to bound the Rademacher complexity of (2). We emphasize that the proof needs more subtle analyses in IMOP than that of one similar generalization bound in IOP (Bertsimas et al., 2015). Due to the introduction of K weight samples, one needs to first establish the relationship between Rademacher complexity of the space of minimizing K functions and Rademacher complexity of the space of a single function, which is non-trivial. Essentially, this theorem indicates that the true risk for the estimator $\hat{\theta}_K^N$, which can be seen as the test error for fixed weight samples $\{w_k\}_{k \in [K]}$, is no worse than the empirical risk, which can be seen as the training error, by an additional term that is of $\mathcal{O}(1/\sqrt{N})$.

4. Connections between IMOP, clustering, and manifold Learning

In section 2.2, we show that IMOP in essence is an unsupervised learning task. Subsequently, we study connections between IMOP and two unsupervised learning tasks. The first one is the K-means clustering problem (MacQueen, 1967; Lloyd, 1982). The second one is the manifold learning problem, which seeks to construct low-dimensional manifolds from data points embedded in high-dimensional spaces (Roweis & Saul, 2000; Tenenbaum et al., 2000).

4.1. Connection bewteen IMOP and clustering

K-means clustering aims to partition the observations into K clusters such that the average squared distance between each observation and its closest cluster centroid is minimized. Given observations $\{\mathbf{y}_i\}_{i \in [N]}$, a mathematical formulation of K-means clustering is presented in the following (Bagirov, 2008; Aloise & Hansen, 2009):

$$\begin{array}{lll} \min\limits_{\mathbf{x}_k, z_{ik}} & \frac{1}{N} \sum\limits_{i \in [N]} \|\mathbf{y}_i - \sum\limits_{k \in [K]} z_{ik} \mathbf{x}_k\|_2^2 & \\ \text{s.t.} & \mathbf{x}_k \in \mathbb{R}^n, & \forall k \in [K], \\ & \sum\limits_{k \in [K]} z_{ik} = 1, & \forall i \in [N], \\ & z_{ik} \in \{0, 1\}, & \forall i \in [N],\ k \in [K], \end{array} \tag{4}$$

where K is the number of clusters, and $\{\mathbf{x}_k\}_{k \in [K]}$ are the centroids of the clusters.

Proposition 4.1. Given any (4), we can construct an instance of (3), such that solving (4) is equivalent to solving the corresponding (3).

The key step for the proof in Proposition 4.1 is to construct a DMP whose objective functions are quadratic and feasible region is a ball. Details of the proof are given in the supplementary material.

Theorem 4.2. (3) is NP-hard to solve.

Note that K-means clustering problem is NP-hard even for instances in the plane (Mahajan et al., 2012), or with two clusters in the general dimension (Aloise et al., 2009). This suggests that (3) is also difficult to solve even for instances in the plane, or $K = 2$ in general dimension.

Remark 4.1. As (3) belongs to the bi-level optimization problem, it is not unexpected that solving (3) is NP-hard as the bi-level optimization problem is NP-hard in general. However, one still needs to construct a polynomial reduction from one NP-hard problem to (3) in order to show the NP-hardness.

Now, we explain why we can interpret (3) as a Constrained K-means clustering problem. Here, the meaning of *Constrained* in our paper is slightly different from that of

(Wagstaff et al., 2001). While both emphasize the incorporation of background knowledge into the clustering process, their *Constrained* means more about which instances should or should not be grouped together. One can observe that (3) has one more type of constraints than (4), i.e., the constraints $\mathbf{x}_k \in S(w_k, \theta), \forall k \in [K]$. All the other components are the same. These constraints require that the centroids of the clusters are restricted to be Pareto optimal solutions of the estimated DMP.

4.2. Connection between IMOP and manifold learning

Given a set of high-dimensional observations $\{\mathbf{y}_i\}_{i\in[N]}$ in $\mathbb{R}^n$, manifold learning attempts to find an embedding set $\{\mathbf{x}_i\}_{i\in[N]}$ in a low-dimensional space $\mathbb{R}^d$ $(d < n)$, and the local manifold structure formed by $\{\mathbf{y}_i\}_{i\in[N]}$ is preserved in the embedded space (Tenenbaum et al., 2000; Roweis & Saul, 2000; Saul & Roweis, 2003; Smith et al., 2009).

Formally, given a set of data points $\{\mathbf{y}_i\}_{i\in[N]}$, we are required to find a mapping $f : \mathbb{R}^d \to \mathbb{R}^n$ and another set of points $\{\mathbf{x}_i\}_{i\in[N]}$ in $\mathbb{R}^d$ such that $\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon_i, i \in [N]$, where ϵ_i represents random noise.

In this paper, we assume that the dimension of the decision space is larger than the number of objectives: $n \geq p$.

Theorem 4.3 (Pareto manifold)**.** Under the same assumptions as Theorem 3.2, if $\forall w_1, w_2 \in \mathscr{W}_p$, $w_1 \neq w_2$ implies $S(w_1, \theta) \neq S(w_2, \theta)$ for each $\theta \in \Theta$, we have that the Pareto optimal set of DMP is a $(p-1)$-dimensional manifold.

Here, we would like to highlight that the manifold mentioned in Theorem 4.3 is a topology manifold with boundary (Milnor & Weaver, 1997). Therefore, the Pareto optimal set of a DMP with two objectives is a piecewise continuous curve, and the Pareto optimal set of a DMP with three objectives is a piecewise continuous surface, etc.

Theorem 4.4. Suppose that both $\mathbf{f}(\mathbf{x}, \theta)$ and $\mathbf{g}(\mathbf{x}, \theta)$ are linear functions in $\mathbf{x}$ for all $\theta \in \Theta$. Then, $X_P(\theta)$ is a piecewise linear manifold that has dimension not exceeding $p-1$ for all $\theta \in \Theta$.

Note that the feasible set for a multiobjective linear program is a polyhedron. Thus, one way to interpret Theorem 4.4 is that the Pareto optimal set of such a program consists of Pareto optimal faces of the polyhedron that are arc-wise connected. Therefore, the Pareto optimal set naturally has a piecewise linear structure and forms a manifold. Note that each piece might have different dimensions. In this case, the Pareto optimal set of a linear program is a special manifold that is the disjoint union of topological manifolds with different dimensions.

Now, we explain why we can interpret IMOP as a manifold learning problem. One can show that $\{\mathbf{x}_k\}_{k\in[K]}$ in (3) are Pareto optimal points on the $X_P(\theta)$ to be estimated. In addition, we note that (3) is solved by minimizing the average distance between $\{\mathbf{y}_i\}_{i\in[N]}$ and $\{\mathbf{x}_k\}_{k\in[K]}$. Therefore, IMOP essentially seeks to find the DMP whose Pareto optimal set matches best the true Pareto optimal set where $\{\mathbf{y}_i\}_{i\in[N]}$ are sampled from.

Remark 4.2. Manifold learning methods typically returns a set of points in the dimension-reduced space (Roweis & Saul, 2000; Tenenbaum et al., 2000). By solving (3), however, we obtain a set of representative points $\{\mathbf{x}_k\}_{k\in[K]}$ of a manifold in the decision space, instead of the $(p-1)$-dimensional space. Thus, the manifold recovered by solving IMOP is more like the Principal manifold introduced by (Hastie & Stuetzle, 1989) as lines or surfaces passing through“the middle” of the data distribution.

5. Algorithms for solving IMOP

Leveraging the connections between IMOP, clustering, and manifold learning, we propose two algorithms to solve IMOP through manifold learning and clustering.

5.1. A Clustering-type Algorithm

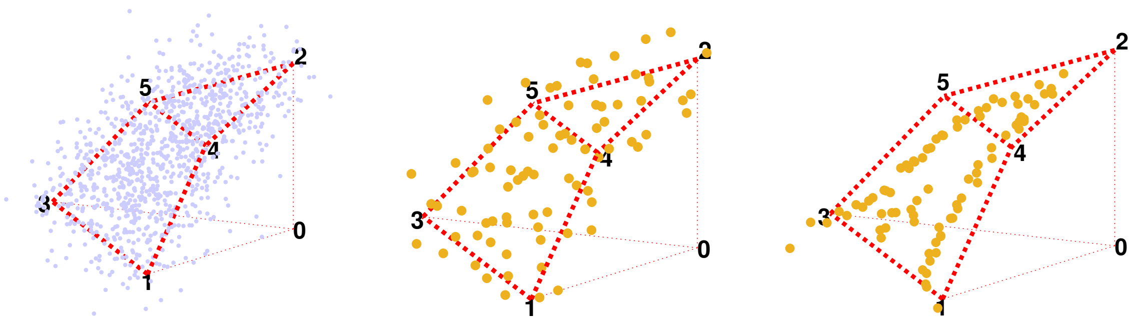
For each $k \in [K]$, we denote C_k the set of noisy decisions with $z_{ik} = 1$ after solving (3) to optimal. Consequently, we partition $\{\mathbf{y}_i\}_{i\in[N]}$ into K clusters $\{C_k\}_{k\in[K]}$. Let $\bar{\mathbf{y}}_k = \frac{1}{|C_k|}\sum_{\mathbf{y}_i \in C_k} \mathbf{y}_i$ be the centroid of cluster C_k, and denote $Var(C_k)$ the variance of C_k. Algebraically, we get

$$\begin{aligned} M_K^N(\theta) &= \frac{1}{N}\sum_{i\in[N]} \|\mathbf{y}_i - \sum_{k\in[K]} z_{ik}\mathbf{x}_k\|_2^2 \\ &= \frac{1}{N}\sum_{k\in[K]} |C_k|\Big(\|\bar{\mathbf{y}}_k - \mathbf{x}_k\|_2^2 + Var(C_k)\Big). \end{aligned}$$

Note that $\{Var(C_k)\}_{k\in[K]}$ is fixed when clusters $\{C_k\}_{k\in[K]}$ are given. However, similar to K-means clustering, $\{C_k\}_{k\in[K]}$ are not known a priori. In K-means clustering algorithm (Lloyd, 1982), this problem is solved by initializing the clusters, and then iteratively updating the clusters until convergence. Similarly, we propose a procedure that alternately clusters the noisy decisions (assignment step) and find θ and $\{\mathbf{x}_k\}_{k\in[K]}$ (update step) until convergence. Specifically, the update step can be established by solving the following problem

$$\begin{array}{lll} \min\limits_{\theta, \mathbf{x}_{k'}} & \frac{1}{N}\sum\limits_{k\in[K]} |C_k|\|\bar{\mathbf{y}}_k - \sum\limits_{k'\in[K]} z_{kk'}\mathbf{x}_{k'}\|_2^2 & \\ \text{s.t.} & \mathbf{x}_{k'} \in S(w_{k'}, \theta), & \forall k' \in [K], \\ & \sum\limits_{k'\in[K]} z_{kk'} = 1, & \forall k \in [K], \\ & z_{kk'} \in \{0, 1\}, & \forall k \in [K],\ k' \in [K]. \end{array} \tag{5}$$

The expectation-maximization (EM)-style algorithm is formally presented in Algorithm 1.



(a) Data (b) Algorithm 1 (c) Algorithm 2

Figure 2. Tri-objective linear program. (a) The light blue dots indicate the $N = 10000$ observations in the data set. (b) Orange dots indicate the centroids using K-means clustering. (c) Orange dots indicate the centroids after using Kernel PCA and K-means clustering.

Algorithm 1 Solving IMOP through clustering

1: **Input:** Noisy decisions $\{\mathbf{y}_i\}_{i\in[N]}$, evenly sampled weights $\{w_k\}_{k\in[K]}$.
2: **Initialization**: Group $\{\mathbf{y}_i\}_{i\in[N]}$ into K clusters through K-means clustering.
Find the clusters $\{C_k\}_{k\in[K]}$ and centroids $\{\overline{\mathbf{y}}_k\}_{k\in[K]}$.
Solve (5) and get $\hat{\theta}$ and $\{\mathbf{x}_k\}_{k\in[K]}$.
3: **while** Stopping criterion is not satisfied **do**
4: **Assignment step**: Assign each $\mathbf{y}_i$ to the closest $\mathbf{x}_k$ and get $\{C_k\}_{k\in[K]}$. Calculate centroids $\{\overline{\mathbf{y}}_k\}_{k\in[K]}$.
5: **Update step**: Update θ and $\{\mathbf{x}_k\}_{k\in[K]}$ by (5).
6: **end while**

Algorithm 2 Initialization with manifold learning

1: **Input:** Noisy decision $\{\mathbf{y}_i\}_{i\in[N]}$, evenly sampled weights $\{w_k\}_{k\in[K]}$.
2: Apply any nonlinear manifold learning algorithm: $\mathbf{y}_i \in \mathbb{R}^n \rightarrow \mathbf{x}_i \in \mathbb{R}^{p-1}, \forall i \in [N]$.
3: Group $\{\mathbf{x}_i\}_{i\in[N]}$ into K clusters by K-means clustering.
Denote I_K the set of labels of $\{\mathbf{x}_i\}_{i\in[N]}$.
Find the clusters $\{C_k\}_{k\in[K]}$ and centroids $\{\overline{\mathbf{y}}_k\}_{k\in[K]}$ of $\{\mathbf{y}_i\}_{i\in[N]}$ according to I_K.
4: Solve (5) and get $\hat{\theta}$ and $\{\mathbf{x}_k\}_{k\in[K]}$.
5: Run Step 3 - 6 in Algorithm 1.

Although (3) is non-convex, Algorithm 1 converges in finite steps as shown below.

Theorem 5.1. Suppose there is an oracle to solve (5). Algorithm 1 converges to a (local) optimal solution of (3) in a finite number of iterations.

In practice, Algorithm 1 converges pretty fast, typically within several iterations. The main reason is that the **Initialization** step often provides a good estimation of the true parameter, since the K centroids returned by K-means clustering represent the observations quite well in general.

5.2. An Enhanced Algorithm with Manifold Learning

We provide another algorithm leveraging the connection that the Pareto optimal set is a continuous manifold with intrinsic dimension of $p-1$, where p is the number of objectives, regardless of the dimension of the decision space.

Linear manifold learning methods, such as PCA and Linear Discriminant Analysis, perform well when there exists a linear structure in the data. However, applying them in step 2 might not be appropriate since our data has a non-linear structure by Theorem 4.3, even for the simplest linear case stated in Corollary 4.4.

Theorem 5.2. Suppose there is an oracle to solve (5). Algorithm 2 converges to a (local) optimal solution of (3) in a finite number of iterations.

6. Experiments

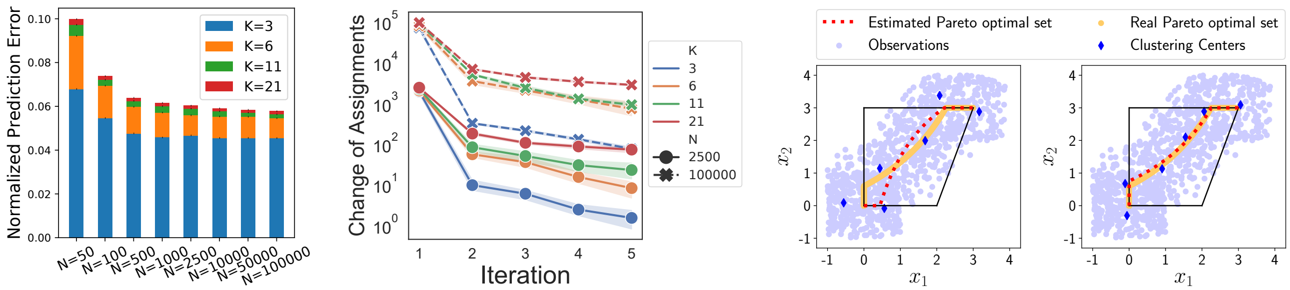
We provide preliminary results to illustrate the performance of the proposed Algorithms 1 and 2 on both synthetic and real datasets. Advanced optimization techniques, e.g., **ADMM**, are applied to enhance the efficiency of our algorithms. Details of the experiments and the associated techniques can be seen in the supplementary material.

6.1. Learning the Objective Functions of a LP

Consider the following tri-objective linear program

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^3_+} & \{\mathbf{c}_1^T\mathbf{x}, \mathbf{c}_2^T\mathbf{x}, \mathbf{c}_3^T\mathbf{x}\} \\ s.t. & A\mathbf{x} \leq \mathbf{b}. \end{array}$$

In this example, there are two Pareto optimal faces, one is the triangle defined by vertices $(2, 4, 5)$, the other one is the tetragon defined by vertices $(1, 3, 5, 4)$. As is shown by Figure 2a, the dimension of the Pareto optimal set is 2,



(a)

(b)

(c) Algorithm 1 vs Algorithm 2

Figure 3. Learning objective functions of an MQP. (a) Normalized prediction error for different N and K. (b) Change of the assignments in 5 iterations over 10 repetitions for different N and K. (c) Estimation results with $N = 1000, K = 6$. **Left**: Algorithm 1. **Right**: Algorithm 2 using tSNE for manifold learning.

which equals to $p - 1$. Here, $p = 3$ since there are 3 objective functions. We seek to learn the objective functions, i.e. $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\}$, given the Pareto optimal solutions corrupted by noise.

Our results show that both Algorithm 1 and Algorithm 2 are able to learn objective functions that recover the true Pareto optimal set even with the initial estimation of the parameter in the Initialization step. Thus, we won't run the later steps in Algorithm 1. However, as illustrated in Figure 2b - 2c, the initialization step with manifold learning almost projects the noisy observations onto the true Pareto optimal set and thus accelerates the learning process.

6.2. Learning the Objective Functions of an MQP

We consider the following multiobjective quadratic programming problem (MQP)

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^2_+} & \begin{pmatrix} f_1(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} + \mathbf{c}_1^T \mathbf{x} \\ f_2(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_2 \mathbf{x} + \mathbf{c}_2^T \mathbf{x} \end{pmatrix} \\ s.t. & A\mathbf{x} \geq \mathbf{b}. \end{array}$$

We seek to learn $\mathbf{c}_1$ and $\mathbf{c}_2$ in this experiment. Estimation results are illustrated in Figure 3.

6.3. Experiments with Real Data: Learning the Expected Returns

We consider various decisions arising from different investors in a stock market. Specifically, we consider a portfolio selection problem. The classical Markovitz mean-variance portfolio selection (Markowitz, 1952) is

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \begin{pmatrix} f_1(\mathbf{x}) & = -\mathbf{r}^T \mathbf{x} \\ f_2(\mathbf{x}) & = \mathbf{x}^T Q \mathbf{x} \end{pmatrix} \\ s.t. & 0 \leq x_i \leq b_i, \forall i \in [n], \\ & \sum\limits_{i=1}^{n} x_i = 1, \end{array}$$

where $\mathbf{r} \in \mathbb{R}^n_+$ is a vector of individual security expected returns, $Q \in \mathbb{R}^{n\times n}$ is the covariance matrix of securities returns, $\mathbf{x}$ is a portfolio specifying the proportions of capital to be invested in the different securities, and b_i is an upper bound put on the proportion of security $i \in [n]$.

Dataset: Stock price data is scraped from S&P 500 Index. Quarterly portfolio data is scraped from the mutual fund VHCAX (Vanguard Capital Opportunity Fund Admiral Shares) and SWPPX (Schwab S&P 500 Stock Index) from March 2010 to December 2019. The assets are grouped into 11 sectors as listed in Table 2.

We first learn the average quarterly returns $\mathbf{r}$ for the 11 sectors from the portfolio data. We treat the learned returns as the market equilibrium returns. Note that the Black-Litterman model is an asset allocation approach that allows investment analysts to incorporate subjective views (based on investment analyst estimates) into market equilibrium returns. By blending analyst views and equilibrium returns instead of relying only on historical asset returns, the Black-Litterman model provides a systematic way to estimate the mean of asset returns. Our market views of the 11 sectors: Energy, Healthcare, Technology and Utilities have higher returns than the equilibrium returns. We therefore add an additional return that follows the uniform distribution $U(0, 0.1)$ to each of the 4 sectors to reflect our market views. We then seek to learn the blended expected return for each of the 4 sectors.

As we can see from Figure 4, the estimated efficient frontier becomes closer and closer to the true efficient frontier as we sample more and more weights, which shows the consistency of our model. Moreover, this also suggests that the capability of our model for IMOP has the potential to be scaled up with more computational resources.

7. Conclusion

We consider in this paper the problem of learning parameters of an expert's multiobjective decision making model, based on a set of observed decisions. Specifically, we for-

Table 2. 11 sectors for the assets

Materials	Communication	Consumer Cyclical	Consumer Defensive	Energy	Financial Services
Healthcare	Industrial	Real Estate	Technology	Utilities	

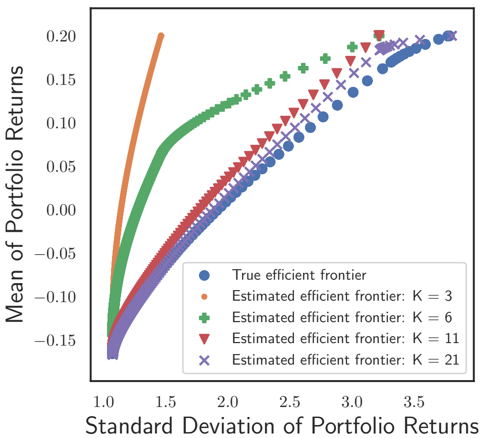


Figure 4. The efficient frontier and estimated efficient frontier.

mulate such a learning task as an inverse multiobjective optimization problem, and provide a deep analysis to establish the statistical significance of the inference results from the presented model. We also reveal several fundamental connections between IMOP, K-means clustering, and manifold learning. We show the effectiveness of our model and the computational efficacy of algorithms to solve large-scale IMOP by extensive numerical experiments.

Acknowledgements

The authors would like to thank Shi Yu for providing the Portfolio data used in this work and the discussion in designing the experiment in Section 6.3.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004.

Aggarwal, C. C. et al. *Recommender systems*, volume 1. Springer.

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

Aloise, D. and Hansen, P. A branch-and-cut sdp-based algorithm for minimum sum-of-squares clustering. *Pesquisa Operacional*, 29(3):503–516, 2009.

Aloise, D., Deshpande, A., Hansen, P., and Popat, P. Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.

Amin, K., Jiang, N., and Singh, S. Repeated inverse reinforcement learning. In *NIPS*. 2017.

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Aswani, A., Shen, Z.-J., and Siddiq, A. Inverse optimization with noisy data. *Operations Research*, 2018.

Bagirov, A. M. Modified global k-means algorithm for minimum sum-of-squares clustering problems. *Pattern Recognition*, 41(10):3192–3199, 2008.

Bärmann, A., Pokutta, S., and Schneider, O. Emulating the expert: Inverse optimization through online learning. In *ICML*, 2017.

Bertsimas, D., Gupta, V., and Paschalidis, I. C. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153(2):595–633, 2015.

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

Chan, T. C., Craig, T., Lee, T., and Sharpe, M. B. Generalized inverse multiobjective optimization with application to cancer therapy. *Operations Research*, 62(3):680–695, 2014.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Dong, C., Chen, Y., and Zeng, B. Generalized inverse optimization through online learning. In *NeurIPS*, 2018.

Esfahani, P. M., Shafieezadeh-Abadeh, S., Hanasusanto, G. A., and Kuhn, D. Data-driven inverse optimization with imperfect information. *Mathematical Programming*, 167(1):191–234, 2018.

Finn, C., Levine, S., and Abbeel, P. Guided cost learning: Deep inverse optimal control via policy optimization. In *ICML*, 2016.

Gass, S. and Saaty, T. The computational algorithm for the parametric objective function. *Naval Research Logistics*, 2(1-2):39–45, 1955.

Greco, S., Figueira, J., and Ehrgott, M. *Multiple Criteria Decision Analysis*. Springer, 2016.

Hadfield-Menell, D., Russell, S. J., Abbeel, P., and Dragan, A. Cooperative inverse reinforcement learning. In *NIPS*. 2016.

Hastie, T. and Stuetzle, W. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer, 2001.

Ho, J., Gupta, J., and Ermon, S. Model-free imitation learning with policy optimization. In *ICML*, 2016.

Hwang, C.-L. and Masud, A. S. M. *Multiple Objective Decision Making—methods and Applications: A State-of-the-art Survey*, volume 164. Springer Science & Business Media, 2012.

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

Jennrich, R. I. Asymptotic properties of non-linear least squares estimators. *The Annals of Mathematical Statistics*, 40(2):633–643, 1969.

Keshavarz, A., Wang, Y., and Boyd, S. Imputing a convex objective function. In *Intelligent Control (ISIC), 2011 IEEE International Symposium on*, pp. 613–619. IEEE, 2011.

Lloyd, S. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056489.

MacQueen, J. B. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 281–297, 1967.

Mahajan, M., Nimbhorkar, P., and Varadarajan, K. The planar k-means problem is np-hard. *Theoretical Computer Science*, 442:13–21, 2012.

Markowitz, H. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

Metelli, A. M., Pirotta, M., and Restelli, M. Compatible reward inverse reinforcement learning. In *NIPS*. 2017.

Miettinen, K. *Nonlinear Multiobjective Optimization*, volume 12. Springer Science & Business Media, 2012.

Milnor, J. and Weaver, D. W. *Topology from the differentiable viewpoint*. Princeton university press, 1997.

Ng, A. Y. and Russell, S. J. Algorithms for inverse reinforcement learning. In *ICML*, 2000.

Pirotta, M. and Restelli, M. Inverse reinforcement learning through policy gradient minimization. In *AAAI*, 2016.

Ramachandran, D. and Amir, E. Bayesian inverse reinforcement learning. *Urbana*, 51(61801):1–4.

Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A. Maximum margin planning. In *ICML*, 2006.

Ratliff, N. D., Silver, D., and Bagnell, J. A. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27(1):25–53, 2009.

Roland, J., De Smet, Y., and Figueira, J. R. Inverse multi-objective combinatorial optimization. *Discrete Applied Mathematics*, 161(16):2764–2771, 2013.

Roweis, S. T. and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290 (5500):2323–2326, 2000.

Saul, L. K. and Roweis, S. T. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of machine learning research*, 4(Jun):119–155, 2003.

Schaefer, A. J. Inverse integer programming. *Optimization Letters*, 3(4):483–489, 2009.

Smith, A., Zha, H., and Wu, X.-m. Convergence and rate of convergence of a manifold-based dimension reduction algorithm. In *NIPS*, 2009.

Syed, U., Bowling, M., and Schapire, R. E. Apprenticeship learning using linear programming. In *ICML*, 2008.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Wagstaff, K., Cardie, C., Rogers, S., and Schroedl, S. Constrained k-means clustering with background knowledge. In *ICML*, 2001.

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

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.