CORC Technical Report TR-2003-02

# Inverse conic programming and applications

G. Iyengar[*] W. Kang[†]

April 7th, 2003. Revised Dec. 23rd, 2003

**Abstract**

The past decade has seen a growing interest in inverse optimization. It has been shown that duality yields very efficient algorithms for solving inverse linear programming problems. In this paper, we consider a special class of conic programs that admits a similar duality and show that the corresponding inverse optimization problems are efficiently solvable. We discuss the applications of inverse conic programming in portfolio optimization, inverse quadratically constrained quadratic programming and utility function identification.

## 1 Introduction

Consider an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x};\boldsymbol{\theta}) \\ \text{subject to} & \mathbf{x}\in\mathcal{S}(\boldsymbol{\beta}), \end{array} \tag{1}$$

where $(\boldsymbol{\theta},\boldsymbol{\beta})$ are parameters and $\mathbf{x}$ is the decision variable. The goal of this problem, called the *forward problem* [25], is to compute an optimal decision corresponding to an estimate $(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\beta}})$ of the parameters. The *inverse optimization problem* [25] associated with (1) is the following: Given an observed decision $\widehat{\mathbf{x}}$, characterize the set $\boldsymbol{\Theta}(\widehat{\mathbf{x}})$ of parameters $(\boldsymbol{\theta},\boldsymbol{\beta})$ for which $\widehat{\mathbf{x}}$ is optimal; and if desired, solve

$$\begin{array}{ll} \text{minimize} & \|(\boldsymbol{\theta},\boldsymbol{\beta})-(\bar{\boldsymbol{\theta}},\bar{\boldsymbol{\beta}})\| \\ \text{subject to} & (\boldsymbol{\theta},\boldsymbol{\beta})\in\boldsymbol{\Theta}(\widehat{\mathbf{x}}), \end{array} \tag{2}$$

where $(\bar{\boldsymbol{\theta}},\bar{\boldsymbol{\beta}})$ are some nominal parameters and $\|\cdot\|$ is an appropriate norm.

Inverse optimization problems appear in at least three different contexts. The first is system identification; e.g., suppose seismic waves resulting from an earthquake are assumed to travel along shortest paths [10, 11, 23, 25]; then estimating terrain properties from the observed paths

[*]IEOR Department, Columbia University, Email: `garud@ieor.columbia.edu`. Research partially supported by NSF grants CCR-00-09972, DMS-01-04282 and ONR grant N000140310514.

[†]IEOR Department, Columbia University, Email: `wk2003@columbia.edu`

of seismic waves can be formulated as an inverse optimization problem. The second context is selecting parameters $(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\beta}})$ to force a desired response $\widehat{\mathbf{x}}$; e.g., suppose the traffic flow in a network is assumed to be the solution of an optimization problem with arc costs as parameters; then the problem of determining the minimal toll that ensures a prescribed flow is an instance of inverse optimization [12, 13]. A third context is optimization in uncertain environments. Let $\widehat{\mathbf{x}}$ denote the current decision and let $\boldsymbol{\Theta} = \{(\boldsymbol{\theta}, \boldsymbol{\beta}) : \|(\boldsymbol{\theta}, \boldsymbol{\beta}) - (\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\beta}})\| \leq \epsilon\}$ denote a confidence set for the unknown parameters. An inverse optimization based strategy, to manage the trade-off between the cost associated with modifying a decision and the higher value from a new decision, is as follows: solve (2); for optimal values of at most $\epsilon$, keep the current decision $\widehat{\mathbf{x}}$, otherwise, shift to the new solution corresponding to $(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\beta}})$. We discuss an application of such a policy in Section 3.

Inverse optimization is related to *robust* optimization [5, 6, 16]. However, there are important distinctions. The robust optimization problem associated with a set of parameters $\boldsymbol{\Theta}$ is given by

$$\begin{array}{ll} \text{maximize} & \gamma \\ \text{subject to} & (\mathbf{x}, \gamma) \in \mathcal{X} = \{(\mathbf{x}, \gamma) : \forall (\boldsymbol{\theta}, \boldsymbol{\beta}) \in \boldsymbol{\Theta}, f(\mathbf{x}, \boldsymbol{\theta}) \geq \gamma, \mathbf{x} \in \mathcal{S}(\boldsymbol{\beta})\}. \end{array}$$

Thus, in a robust problem, the decision $\mathbf{x}$ must be feasible for *all* parameter values $\boldsymbol{\beta}$; whereas, in an inverse problem, a given decision $\widehat{\mathbf{x}}$ must be optimal for *some* $(\boldsymbol{\theta}, \boldsymbol{\beta}) \in \boldsymbol{\Theta}$. Clearly, the robust methodology is not appropriate for the first two contexts discussed above. In the third context, one or both of these methodologies may be appropriate (e.g. [16] proposes a robust formulation for the portfolio selection problem discussed in Section 3).

Inverse optimization problems were first formulated in the context of shortest path problems [10, 11]. Subsequently, inverse optimization problems corresponding to several combinatorial optimization problems have been investigated [2, 14, 24, 26, 27]. See [17] for a recent survey. For this class of inverse optimization problems, the feasible set $S(\boldsymbol{\beta})$ is polyhedral and fixed (i.e $\boldsymbol{\beta}$ known and fixed) and the objective function $f(\mathbf{x}; \boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{x}$ (i.e. linear in the unknown $\boldsymbol{\theta}$). Such problems are called inverse linear programs (LPs). It follows from LP duality that inverse LPs can be reformulated as LPs when the norm $\|\cdot\|$ in (2) is the $\mathcal{L}_1$ or the $\mathcal{L}_\infty$ norm [2].

This paper is motivated by the fact that a more general class of problems called conic programs [7] has a duality theory very similar to LPs. It is shown that replacing LP duality with conic duality allows one to conclude that inverse conic programs, where only the objective function is uncertain and the gradient of the objective function is an affine function of the parameter $\boldsymbol{\theta}$, can be reformulated as conic programs when the norm in (2) is an $\mathcal{L}_q$-norm, $q \geq 1$, rational. The emphasis of this paper is not on mathematical innovation (the extension to inverse conic programs is fairly straightforward), but on modeling inverse problems. Since quadratic programs, quadratically constrained programs, and semidefinite programs can all be reformulated as conic programs [7] and these programs can be solved efficiently [22], our extension implies that a much wider class of inverse optimization problems can be efficiently solved in practice.

The organization of this paper is as follows. In Section 2, we formulate inverse conic programs and use standard results from convex analysis to establish that inverse conic programs can be solved efficiently. In Section 3, we discuss an inverse optimization based portfolio selection strategy. In

Section 4, we describe inverse quadratically constrained programs. In Section 5, we develop an inverse optimization based policy for identifying utility functions. In Section 6, we conclude with some comments and remarks.

## 2 Inverse conic programming

We assume that the forward problem (1) has the following form:

$$\begin{array}{rl} \text{minimize} & f(\mathbf{x};\boldsymbol{\theta}) \\ \text{subject to} & \mathbf{g}(\mathbf{x}) \succeq_{\mathcal{K}} \mathbf{0}, \\ & \mathbf{A}\mathbf{x} = \mathbf{b}, \end{array} \tag{3}$$

where $\mathbf{x} \in \mathbf{R}^n$ is the decision variable, $\boldsymbol{\theta} \in \mathbf{R}^p$ is a parameter, $\mathbf{A} \in \mathbf{R}^{l\times n}$ and $\mathbf{b} \in \mathbf{R}^l$. For fixed $\boldsymbol{\theta} \in \mathbf{R}^p$, the function $f(\mathbf{x};\boldsymbol{\theta}) : \mathbf{R}^n \mapsto \mathbf{R}$, is assumed to be convex and differentiable in $\mathbf{x}$; and, for fixed $\mathbf{x} \in \mathbf{R}^n$, the gradient $\nabla_x f(\mathbf{x};\boldsymbol{\theta})$ with respect to $\mathbf{x}$ is assumed to be an affine function of $\boldsymbol{\theta}$. Functions $f(\mathbf{x};\boldsymbol{\theta})$ that satisfy these conditions include:

$$\begin{array}{ll} f(\mathbf{x};\boldsymbol{\theta}) = \boldsymbol{\theta}^T\mathbf{x}, & \nabla_x f(\mathbf{x};\boldsymbol{\theta}) = \boldsymbol{\theta}, \\ f(\mathbf{x};\boldsymbol{\theta}) = \|\mathbf{A}\mathbf{x}+\boldsymbol{\theta}\|_2^2, & \nabla_x f(\mathbf{x};\boldsymbol{\theta}) = 2(\mathbf{A}^T\mathbf{A}\mathbf{x}+\mathbf{A}^T\boldsymbol{\theta}), \\ f(\mathbf{x};\boldsymbol{\theta}) = \mathbf{x}^T\mathbf{Q}\mathbf{x}+2\mathbf{q}^T\mathbf{x}+\gamma, & \nabla_x f(\mathbf{x};\boldsymbol{\theta}) = 2(\mathbf{Q}\mathbf{x}+\mathbf{q}), \text{ where } \boldsymbol{\theta} = (\mathbf{Q}_{11},\mathbf{Q}_{12},\cdots,\mathbf{Q}_{nn};\mathbf{q}). \end{array}$$

The notation $\succeq_{\mathcal{K}}$ (resp. $\succ_{\mathcal{K}}$) denotes the partial (resp. strict) order on $\mathbf{R}^m$ induced by a proper cone $\mathcal{K} \subset \mathbf{R}^m$, i.e. $\mathbf{x} \succeq_{\mathcal{K}} \mathbf{y}$ (resp. $\mathbf{x} \succ_{\mathcal{K}} \mathbf{y}$) iff $\mathbf{x}-\mathbf{y} \in \mathcal{K}$ (resp. $\mathbf{x}-\mathbf{y} \in \mathbf{int}(\mathcal{K})$). We assume that $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \mathcal{K}_k$, where each cone $\mathcal{K}_j$ belongs to one of the following three classes:

(i) Linear cone (LC): $\mathcal{K}_l = \left\{\mathbf{x} \in \mathbf{R}^r : x_i \geq 0, i = 1,\ldots,r\right\}$

(ii) Second-order cone (SOC): $\mathcal{K}_{so} = \left\{\mathbf{x} = (x_0;\bar{\mathbf{x}}) \in \mathbf{R}^{r+1} : x_0 \geq \sqrt{\bar{\mathbf{x}}^T\bar{\mathbf{x}}}\right\}$,

(iii) Semidefinite cone (SDC): $\mathcal{K}_{sd} = \left\{\mathbf{x} \in \mathbf{R}^{r^2} : \mathbf{mat}(\mathbf{x}) \succeq \mathbf{0}\right\}$, where $\mathbf{mat}(\mathbf{x}) \in \mathbf{R}^{r\times r}$ with $\mathbf{mat}(\mathbf{x})_{ij} = \mathbf{x}_{r(i-1)+j}$, and $\mathbf{A} \succeq \mathbf{0}$ denotes $\mathbf{A} = \mathbf{A}^T$ and positive semidefinite.

The dual cone $\mathcal{K}^*$ of the cone $\mathcal{K}$ is given by $\mathcal{K}^* = \{\mathbf{y} : \mathbf{y}^T\mathbf{x} \geq 0, \forall \mathbf{x} \in \mathcal{K}\}$. The cones $\mathcal{K}_l$, $\mathcal{K}_{so}$ and $\mathcal{K}_{sd}$ are self-dual [22]; therefore, $\mathcal{K}^* = (\mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \mathcal{K}_k)^* = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \mathcal{K}_k = \mathcal{K}$, i.e. $\mathcal{K}$ is self-dual.

The function, $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^m$, is assumed to be differentiable and concave with respect to the partial ordering $\succeq_{\mathcal{K}}$, i.e. for all $\mathbf{x}_1$, $\mathbf{x}_2 \in \mathbf{R}^n$, and $\lambda \in [0,1]$, $\mathbf{g}(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \succeq_{\mathcal{K}} \lambda\mathbf{g}(\mathbf{x}_1) + (1-\lambda)\mathbf{g}(\mathbf{x}_2)$. We will denote $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}),\ldots,g_m(\mathbf{x})]^T$. The function $\mathbf{g}$ does not depend on the parameter $\boldsymbol{\theta}$. Problems of the form (3) are called conic programs [7].

We make the following constraint qualification:

**Assumption 1** *There exists an* $\mathbf{x}_0 \in \mathbf{R}^n$ *with* $\mathbf{A}\mathbf{x}_0 = \mathbf{b}$ *and* $\mathbf{g}(\mathbf{x}_0) \succ_{\mathcal{K}} \mathbf{0}$.

**Lemma 1** *The set $\boldsymbol{\Theta}(\widehat{\mathbf{x}})$ of parameter values for which a feasible $\widehat{\mathbf{x}}$ is optimal for (3) is given by*

$$\boldsymbol{\Theta}(\widehat{\mathbf{x}}) = \Big\{\boldsymbol{\theta} \in \mathbf{R}^p : \exists \mathbf{u} \in \mathcal{K}, \mathbf{v} \in \mathbf{R}^l \text{ such that } \nabla_x f(\widehat{\mathbf{x}}; \boldsymbol{\theta}) - \sum_{i=1}^m u_i \nabla_x g_i(\widehat{\mathbf{x}}) - \mathbf{A}^T\mathbf{v} = \mathbf{0},\ \mathbf{u}^T\mathbf{g}(\widehat{\mathbf{x}}) = 0\Big\}. \tag{4}$$

*The associated inverse optimization problem*

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| \\ \text{subject to} & \boldsymbol{\theta} \in \boldsymbol{\Theta}(\widehat{\mathbf{x}}), \end{array} \tag{5}$$

*is a conic program, and can be solved efficiently provided the $\|\cdot\|$ is an $\mathcal{L}_q$ norm, $q \geq 1$, rational.*

**Proof:** To establish the first part, note that the Lagrangian $L(\mathbf{x}, \mathbf{u}, \mathbf{v})$ of (3) is given by $L(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}; \boldsymbol{\theta}) - \mathbf{u}^T\mathbf{g}(\mathbf{x}) - \mathbf{v}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$, where $\mathbf{u} \in \mathbf{R}^m$ and $\mathbf{v} \in \mathbf{R}^l$. From Assumption 1, it follows that a feasible $\widehat{\mathbf{x}}$ is optimal for (3) iff it satisfies the Karush-Kuhn-Tucker (KKT) conditions: $\exists\ \mathbf{u} \in \mathbf{R}^m$, $\mathbf{v} \in \mathbf{R}^l$ such that

$$\begin{aligned} \nabla_x L(\widehat{\mathbf{x}}, \mathbf{u}, \mathbf{v}) &= \nabla_x f(\widehat{\mathbf{x}}; \boldsymbol{\theta}) - \sum_{i=1}^m u_i \nabla_x g_i(\widehat{\mathbf{x}}) - \mathbf{A}^T\mathbf{v} = \mathbf{0}, && (6)\\ \mathbf{u} &\succeq_{\mathcal{K}^*} \mathbf{0}, && (7)\\ \mathbf{u}^T\mathbf{g}(\widehat{\mathbf{x}}) &= 0. && (8) \end{aligned}$$

Since $\nabla_x f(\widehat{\mathbf{x}}; \boldsymbol{\theta})$ is assumed to be affine in $\boldsymbol{\theta}$, the set $\boldsymbol{\Theta}(\widehat{\mathbf{x}})$ is convex; specifically, it is the intersection of an affine space with a cone. Consequently, (5) is a conic program [7]. Results in [7] imply that (5) can be solved efficiently for $\mathcal{L}_q$ norms, $q \geq 1$, rational. ■

Typically, the vectors $\mathbf{u}$ satisfying (8) belong to a subspace with dimension considerably smaller than $m$; therefore, formulating (5) in this subspace can yield significant computational benefits. To this end, partition $\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$ and $\mathbf{g}(\widehat{\mathbf{x}}) = [\mathbf{g}_1(\widehat{\mathbf{x}}), \ldots, \mathbf{g}_k(\widehat{\mathbf{x}})]$ such that $\mathbf{u}_j \in \mathcal{K}_j^* = \mathcal{K}_j$, $\mathbf{g}_j(\widehat{\mathbf{x}}) \in \mathcal{K}_j$, $j = 1, \ldots, k$. Since $\mathbf{u}_j^T\mathbf{g}_j(\widehat{\mathbf{x}}) \geq 0$, it follows that $\mathbf{u}^T\mathbf{g}(\widehat{\mathbf{x}}) = 0$ iff $\mathbf{u}_j^T\mathbf{g}_j(\widehat{\mathbf{x}}) = 0,\ j = 1, \ldots, k$. Lemma 2 simplifies the constraint $\mathbf{u}^T\mathbf{z} = 0$ for the cones $\mathcal{K}_l$, $\mathcal{K}_{so}$ and $\mathcal{K}_{sd}$.

**Lemma 2** *Fix $\mathbf{z} \in \mathcal{K}$. Let $\mathcal{U} \subset \mathcal{K}^*$ denote the set of vectors $\mathbf{u} \in \mathcal{K}^*$ such that $\mathbf{u}^T\mathbf{z} = 0$. Then for*

*(1) linear cone, $\mathcal{K} = \mathcal{K}_l \subset \mathbf{R}^r$ : $\mathcal{U} = \{\mathbf{u} \geq \mathbf{0} : u_j = 0, \forall j \text{ such that } z_j > 0\}$.*

*(2) second-order cone, $\mathcal{K} = \mathcal{K}_{so} \subset \mathbf{R}^{r+1}$ :*

$$\mathcal{U} = \begin{cases} \mathcal{K}_{so}, & \mathbf{z} = \mathbf{0}, \\ \big\{(u_0; \bar{\mathbf{u}}) : \bar{\mathbf{u}} = -\frac{u_0}{z_0}\bar{\mathbf{z}}, u_0 \geq 0\big\}, & \mathbf{z} \in \mathbf{bd}(\mathcal{K}_{so})\backslash\{\mathbf{0}\} \\ \mathbf{0}, & \mathbf{z} \in \mathbf{int}(\mathcal{K}_{so}). \end{cases},$$

*(3) semidefinite cone, $\mathcal{K} = \mathcal{K}_{sd} \subset \mathbf{R}^{r^2}$ : Let $r_z = \mathbf{rank}(\mathbf{mat}(\mathbf{z}))$ and $\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T$ denote the spectral decomposition of $\mathbf{mat}(\mathbf{z})$ where $\boldsymbol{\Lambda} = \mathbf{diag}(\lambda_1, \ldots, \lambda_{r_z}; \mathbf{0})$ and $\lambda_i$'s are eigenvalues of $\mathbf{mat}(\mathbf{z})$. Then*

$$\mathcal{U} = \left\{\mathbf{u} : \mathbf{mat}(\mathbf{u}) = \mathbf{Q}\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_{22}\end{pmatrix}\mathbf{Q}^T, \mathbf{U}_{22} \in \mathbf{R}^{(r-r_z)\times(r-r_z)}, \mathbf{U}_{22} \succeq \mathbf{0}\right\}.$$

**Proof:** The results follow from standard results for conic complementary slackness conditions (CSCs). For the second-order cone CSCs see [4] and for the semidefinite cone CSCs see [3]. For the completeness of this report we have included the proof below.

Part (1) follows from linear programming duality. See [2] for details.

Let $\mathbf{u} = (u_0; \bar{\mathbf{u}}), \mathbf{z} = (z_0; \bar{\mathbf{z}}) \in \mathcal{K}_{so} \subset \mathbf{R}^{r+1}$. Lemma 15 in [4] implies that

$$\mathbf{u}^T\mathbf{z} = 0 \quad \Leftrightarrow \quad \mathbf{u} \circ \mathbf{z} \equiv \begin{pmatrix} \mathbf{z}^T\mathbf{u} \\ u_0\bar{\mathbf{z}} + z_0\bar{\mathbf{u}} \end{pmatrix} = \mathbf{0}.$$

We have the following three mutually exclusive cases:

(i) $z_0 = 0$: In this case $\bar{\mathbf{z}} = 0$, therefore $\mathbf{u}^T\mathbf{z} = 0$ for all $\mathbf{u} \in \mathcal{K}_{so}$.

(ii) $\mathbf{z} \in \mathbf{int}(\mathcal{K}_{so})$: The second identity in $\mathbf{z} \circ \mathbf{u} = \mathbf{0}$ implies $\bar{\mathbf{u}} = -\left(\frac{u_0}{z_0}\right)\bar{\mathbf{z}}$. Therefore, $0 = \mathbf{u}^T\mathbf{z} = \bar{\mathbf{u}}^T\bar{\mathbf{z}} + u_0z_0 = u_0z_0\left(1 - \frac{\|\bar{\mathbf{z}}\|_2^2}{z_0^2}\right)$ . Since $\mathbf{z} \in \mathbf{int}(\mathcal{K}_{so})$, we have $z_0 > \sqrt{\bar{\mathbf{z}}^T\bar{\mathbf{z}}}$. Thus, $u_0 = 0$, and hence, $\mathbf{u} = \mathbf{0}$.

(iii) $\mathbf{z} \in \mathbf{bd}(\mathcal{K}_{so})$, $\mathbf{z} \neq \mathbf{0}$: In this case $\bar{\mathbf{u}} = -\left(\frac{u_0}{z_0}\right)\bar{\mathbf{z}}$.

This establishes part (2).

Let $\mathbf{U} = \mathbf{mat}(\mathbf{u})$ and $\mathbf{Z} = \mathbf{mat}(\mathbf{z})$. Then $\mathbf{u}^T\mathbf{z} = \mathbf{Tr}(\mathbf{UZ})$. Since $\mathbf{U}, \mathbf{Z} \succeq \mathbf{0}$, $\mathbf{Tr}(\mathbf{UZ}) = 0$ if and only if $\mathbf{UZ} = \mathbf{ZU} = \mathbf{0}$ [3]. Let

$$\mathbf{Q}^T\mathbf{U}\mathbf{Q} = \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ \mathbf{U}_{12}^T & \mathbf{U}_{22} \end{pmatrix}$$

where $\mathbf{U}_{11} = \mathbf{U}_{11}^T \in \mathbf{R}^{r_z \times r_z}$, $\mathbf{U}_{12} \in \mathbf{R}^{r_z \times (r - r_z)}$, and $\mathbf{U}_{22} = \mathbf{U}_{22} \in \mathbf{R}^{(r-r_z)\times(r-r_z)}$. Since $\mathbf{U} \succeq \mathbf{0}$, we have that $\mathbf{U}_{11} \succeq \mathbf{0}$ and $\mathbf{U}_{22} \succeq \mathbf{0}$. Next,

$$\mathbf{ZU} = \mathbf{0} \quad \Leftrightarrow \quad \begin{pmatrix} \mathbf{\Lambda} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}(\mathbf{Q}^T\mathbf{U}\mathbf{Q}) = \mathbf{0} \quad \Leftrightarrow \quad \begin{pmatrix} \mathbf{\Lambda}\mathbf{U}_{11} & \mathbf{\Lambda}\mathbf{U}_{12} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \mathbf{0}.$$

Thus, $\mathbf{U}_{11} = \mathbf{0}$ and $\mathbf{U}_{12} = \mathbf{0}$. This establishes part (3). ∎

As was noted in the introduction, the results in this section readily follow from known results in convex duality theory. Our goal is, in fact, to show that inverse conic programming is no harder than inverse LPs, and therefore, one should not hesitate to use it in modeling applications. In the following three sections, we describe three applications of inverse conic programming.

# 3 Portfolio selection

Consider a market with $n$ risky and 1 risk-free asset. Let $\mathbf{r} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$, $\mathbf{\Sigma} \succ \mathbf{0}$, denote the random *excess* return vector (i.e. return in excess of the risk-free rate) of the risky assets. Let $\boldsymbol{\phi}_s$ denote the optimal solution of

$$\max_{\{\boldsymbol{\phi}:\mathbf{A}\boldsymbol{\phi}\leq\mathbf{b},\mathbf{1}^T\boldsymbol{\phi}=1\}} \frac{\boldsymbol{\mu}^T\boldsymbol{\phi}}{\sqrt{\boldsymbol{\phi}^T\mathbf{\Sigma}\boldsymbol{\phi}}}, \tag{9}$$

where $\mathbf{A}\boldsymbol{\phi} \leq \mathbf{b}$ imposes side constraints, e.g., the no-short sales constraint corresponds to $(\mathbf{A}, \mathbf{b}) = (-\mathbf{I}, \mathbf{0})$. The portfolio $\boldsymbol{\phi}_s$ is called the Sharpe optimal portfolio and the corresponding optimal value $s(\boldsymbol{\mu})$ is called the *Sharpe ratio*. Suppose investors in this market have concave non-decreasing utility functions. Then, the 2-fund separation theorem [18] states that the optimal portfolio $\boldsymbol{\psi}^*$ for any investor is given by $\boldsymbol{\psi}^* = W(\beta e_f + (1-\beta)\boldsymbol{\phi}_s)$, where $e_f$ denotes the risk-free asset, $W$ is the investor's capital, the weight $\beta \in [0, 1]$ is dictated by the investor's utility function.

In a typical market the covariance matrix $\boldsymbol{\Sigma}$ is relatively stable [20]; however, the expected return vector $\boldsymbol{\mu}$ is not. Let $\widehat{\boldsymbol{\mu}}$ denote the current estimate of $\boldsymbol{\mu}$ and let $\widehat{\boldsymbol{\phi}}$ denote corresponding Sharpe optimal portfolio. Thus, $\boldsymbol{\psi}^* = W(\beta e_f + (1-\beta)\widehat{\boldsymbol{\phi}})$. Suppose new information (e.g. new observations of daily returns) becomes available; and the new estimate for $\boldsymbol{\mu}$ is $\bar{\boldsymbol{\mu}}(\neq \widehat{\boldsymbol{\mu}})$. Now, the new $\boldsymbol{\psi}^* = W(\beta e_f + (1-\beta)\bar{\boldsymbol{\phi}})$, where $\bar{\boldsymbol{\phi}}$ is Sharpe optimal for $\bar{\boldsymbol{\mu}}$. Markets typically have transaction costs; therefore, rebalancing to the new $\boldsymbol{\psi}^*$ would result in a loss. On the other hand, the new $\boldsymbol{\psi}^*$ is likely to have a higher risk-adjusted return. Thus, in order to maximize the overall return the investor needs to trade-off transaction cost with higher expected return.

Let $C_\alpha = \big\{\boldsymbol{\mu} \in \mathbf{R}^n : (\boldsymbol{\mu} - \bar{\boldsymbol{\mu}})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}) \leq c_\alpha^2\big\}$ denote the $\alpha$-confidence interval for $\boldsymbol{\mu}$ (we have implicitly used the fact that $\boldsymbol{\Sigma}$ is relatively stable). Since all $\boldsymbol{\mu} \in C_\alpha$ are statistically indistinguishable, an inverse optimization based strategy to manage the trade-off between costs and higher return is as follows: rebalance only if the current portfolio $\widehat{\boldsymbol{\phi}}$ is not optimal for any $\boldsymbol{\mu} \in C_\alpha$, i.e.

$$\begin{array}{rlll} c_\alpha & \leq & \text{minimize} & \sqrt{(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}})} \\ & & \text{subject to} & \boldsymbol{\mu} \in \boldsymbol{\Theta}(\widehat{\boldsymbol{\phi}}) \stackrel{\Delta}{=} \Big\{\boldsymbol{\mu} : \widehat{\boldsymbol{\phi}} \text{ is optimal for (9) and } s(\boldsymbol{\mu}) \geq 0\Big\}. \end{array} \tag{10}$$

**Remark 1** *Sharpe ratio $s(\boldsymbol{\mu}) \geq 0$ is a minimal requirement – no investor will consider investing in risky assets unless there is a portfolio of risky assets with positive excess returns.*

For this application one may also adopt a robust approach and use the minimax Sharpe optimal portfolio $\underline{\boldsymbol{\phi}}_s \in \text{argmax}_{\{\boldsymbol{\phi}:\mathbf{A}\boldsymbol{\phi}\leq\mathbf{b},\mathbf{1}^T\boldsymbol{\phi}=1\}} \min_{\boldsymbol{\mu}\in C_\alpha} \frac{\boldsymbol{\mu}^T\boldsymbol{\phi}}{\sqrt{\boldsymbol{\phi}^T\boldsymbol{\Sigma}\boldsymbol{\phi}}}$ instead of the Sharpe optimal portfolio $\bar{\boldsymbol{\phi}}$ corresponding to $\bar{\boldsymbol{\mu}}$. This approach is discussed in [16].

As a first step towards solving (10) we reformulate (9) as an optimization problem of the form (3).

**Claim 1** *Suppose the Sharpe ratio $s(\boldsymbol{\mu}) \geq 0$. Then (9) is equivalent to*

$$\begin{array}{lrcrcl} \textit{maximize} & \boldsymbol{\mu}^T\mathbf{x} & & & & \\ \textit{subject to} & \mathbf{1}^T\mathbf{x} & - & \xi & = & 0, \\ & -\mathbf{A}\mathbf{x} & + & \mathbf{b}\xi & \geq & \mathbf{0}, \\ & & & \xi & \geq & 0, \\ & [1; \mathbf{L}\mathbf{x}] & & & \succeq_{\mathcal{K}} & \mathbf{0}, \end{array} \tag{11}$$

*where $\xi$ is a homogenizing variable, $\boldsymbol{\Sigma} = \mathbf{L}^T\mathbf{L}$, and the cone $\mathcal{K} = \mathcal{K}_{so} \subset \mathbf{R}^{n+1}$.*

**Proof:** Let $\boldsymbol{\phi}^*$ be optimal for (9). Since $\boldsymbol{\Sigma} \succ \mathbf{0}$ and $\mathbf{1}^T\boldsymbol{\phi}^* = 1$, it follows that $\boldsymbol{\phi}^* \neq \mathbf{0}$, $\sqrt{(\boldsymbol{\phi}^*)^T\boldsymbol{\Sigma}\boldsymbol{\phi}^*} > 0$ and $(\mathbf{x}, \xi) = \frac{1}{\sqrt{(\boldsymbol{\phi}^*)^T\boldsymbol{\Sigma}\boldsymbol{\phi}^*}}(\boldsymbol{\phi}^*, 1)$ is feasible for (11). Also $\boldsymbol{\mu}^T\mathbf{x} = \frac{\boldsymbol{\mu}^T\boldsymbol{\phi}^*}{\sqrt{(\boldsymbol{\phi}^*)^T\boldsymbol{\Sigma}\boldsymbol{\phi}^*}}$, thus the optimal value of (11) is at least as large as the Sharpe ratio $s(\boldsymbol{\mu})$.

Since $\boldsymbol{\Sigma} \succ \mathbf{0}$ and the constraint $[1; \mathbf{L}\mathbf{x}] \succeq_{\mathcal{K}} \mathbf{0}$ is equivalent to $\sqrt{\mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x}} \leq 1$, $(\mathbf{x}, \xi)$ feasible for (11) are bounded. Let $(\mathbf{x}^*, \xi^*)$ denote any optimal solution of (11). Since $(\mathbf{0}, 0)$ is feasible for (11), $\boldsymbol{\mu}^T\mathbf{x}^* \geq 0$. Therefore, we only need to consider the following three cases:

(i) $\xi^* > 0$: Define $\boldsymbol{\phi} = \frac{1}{\xi^*}\mathbf{x}^*$. $\boldsymbol{\phi}$ is feasible for (9). Since $(\mathbf{x}^*)^T\boldsymbol{\Sigma}\mathbf{x}^* \leq 1$, $\frac{\boldsymbol{\mu}^T\boldsymbol{\phi}}{\sqrt{\boldsymbol{\phi}^T\boldsymbol{\Sigma}\boldsymbol{\phi}}} = \frac{\boldsymbol{\mu}^T\mathbf{x}^*}{\sqrt{(\mathbf{x}^*)^T\boldsymbol{\Sigma}\mathbf{x}^*}} \geq \boldsymbol{\mu}^T\mathbf{x}^*$.

(ii) $\xi^* = 0$ and $\boldsymbol{\mu}^T\mathbf{x}^* > 0$: Since $\mathbf{A}\mathbf{x}^* = 0$, $\mathbf{1}^T\mathbf{x}^* = 0$ and $\boldsymbol{\mu}^T\mathbf{x}^* > 0$, we must have $(\mathbf{x}^*)^T\boldsymbol{\Sigma}\mathbf{x}^* = 1$. Let $\boldsymbol{\phi}$ denote any feasible portfolio for (9). Define $\boldsymbol{\phi}_\gamma = \boldsymbol{\phi} + \gamma\mathbf{x}^*$, $\gamma > 0$. Then, for all $\gamma \geq 0$, $\boldsymbol{\phi}_\gamma$ is feasible. Moreover $\boldsymbol{\phi}_\gamma \neq \mathbf{0}$, therefore, $\boldsymbol{\phi}_\gamma^T\boldsymbol{\Sigma}\boldsymbol{\phi}_\gamma > 0$, and $\lim_{\gamma\to\infty}\left\{\frac{\boldsymbol{\mu}^T\boldsymbol{\phi}_\gamma}{\sqrt{\boldsymbol{\phi}_\gamma^T\boldsymbol{\Sigma}\boldsymbol{\phi}_\gamma}}\right\} = \frac{\boldsymbol{\mu}^T\mathbf{x}^*}{\sqrt{(\mathbf{x}^*)^T\boldsymbol{\Sigma}\mathbf{x}^*}} = \boldsymbol{\mu}^T\mathbf{x}^*$.

(iii) $\xi^* = 0$ and $\boldsymbol{\mu}^T\mathbf{x}^* = 0$: Since $\boldsymbol{\mu}^T\mathbf{x}^* \geq s(\boldsymbol{\mu})$, and, by assumption, $s(\boldsymbol{\mu}) \geq 0$, we have $s(\boldsymbol{\mu}) = 0$.

This establishes the result. ∎

Lemma 1 and Lemma 2 lead to the following simplification of the inverse problem (10).

**Claim 2** *Suppose there exists a portfolio $\boldsymbol{\phi}_0$ with $\mathbf{A}\boldsymbol{\phi}_0 < \mathbf{b}$ and $\mathbf{1}^T\boldsymbol{\phi}_0 = 1$. Let $\bar{\boldsymbol{\mu}}$ denote the new estimate of $\boldsymbol{\mu}$. Let $\widehat{\boldsymbol{\phi}}$ denote any feasible portfolio for (9), and let $(\widehat{\mathbf{x}}, \widehat{\xi}) \equiv (\widehat{\boldsymbol{\phi}}, 1)/\sqrt{\widehat{\boldsymbol{\phi}}^T\boldsymbol{\Sigma}\widehat{\boldsymbol{\phi}}}$. Then (10) is equivalent to*

$$\begin{array}{ll} \textit{minimize} & \big((\mathbf{A}_I^T - \mathbf{1}\mathbf{b}_I^T)\mathbf{u}_I + v_0\boldsymbol{\Sigma}\widehat{\mathbf{x}} - \bar{\boldsymbol{\mu}}\big)^T\boldsymbol{\Sigma}^{-1}\big((\mathbf{A}_I^T - \mathbf{1}\mathbf{b}_I^T)\mathbf{u}_I + v_0\boldsymbol{\Sigma}\widehat{\mathbf{x}} - \bar{\boldsymbol{\mu}}\big), \\ \textit{subject to} & \mathbf{u}_I \geq \mathbf{0}, \quad v_0 \geq 0, \end{array} \tag{12}$$

*where $I = \{i : \mathbf{a}_i^T\widehat{\mathbf{x}} = b_i\widehat{\xi}\}$, $\mathbf{A}_I$ and $\mathbf{b}_I$ are the respectively the submatrix of $\mathbf{A}$ and subvector of $\mathbf{b}$ corresponding to rows in $I$, and $\mathbf{u}_I \in \mathbf{R}^{|I|}$.*

**Proof:** The existence of $\boldsymbol{\phi}_0$ ensures that Assumption 1 holds.
Let $\boldsymbol{\Theta}(\widehat{\mathbf{x}}, \widehat{\xi}) = \big\{\boldsymbol{\mu} : (\widehat{\mathbf{x}}, \widehat{\xi}) \text{ is optimal for (11)}\big\}$. Then, Lemma 1 implies that $\boldsymbol{\mu} \in \boldsymbol{\Theta}(\widehat{\mathbf{x}}, \widehat{\xi})$ iff there exist $w \in \mathbf{R}$, $\mathbf{u} \geq \mathbf{0}$, $u_0 \geq 0$, and $(v_0; \mathbf{v}) \succeq_{\mathcal{K}_{so}} \mathbf{0}$ such that

$$-\begin{bmatrix}\boldsymbol{\mu} \\ 0\end{bmatrix} = \begin{bmatrix}-\mathbf{A}^T \\ \mathbf{b}^T\end{bmatrix}\mathbf{u} + \begin{bmatrix}\mathbf{0} \\ 1\end{bmatrix}u_0 + \begin{bmatrix}\mathbf{L}^T \\ 0\end{bmatrix}\mathbf{v} + \begin{bmatrix}\mathbf{1} \\ -1\end{bmatrix}w,$$

and

$$\mathbf{u}^T(\mathbf{A}\widehat{\mathbf{x}} - \mathbf{b}\widehat{\xi}) = \mathbf{0}, \quad u_0\widehat{\xi} = 0, \quad v_0 + \mathbf{v}^T\mathbf{L}\widehat{\mathbf{x}} = 0.$$

Since $\widehat{\xi} > 0$, $u_0 = 0$ and $w = \mathbf{b}^T\mathbf{u}$. Lemma 2 part (1) implies that $u_k = 0$ for all $k \notin I = \{i : \mathbf{a}_i^T\widehat{\mathbf{x}} - b_i\widehat{\xi} = 0\}$. Thus, we can restrict attention to the components $\mathbf{u}_I$ corresponding to the index set $I$. It is easy to see that we must have $\widehat{\mathbf{x}}^T\mathbf{\Sigma}\widehat{\mathbf{x}} = 1$, i.e. $\|\mathbf{L}\widehat{\mathbf{x}}\| = 1$. Hence, Lemma 2 part (2) implies that $\mathbf{v} = -v_0\mathbf{L}\widehat{\mathbf{x}}$, and

$$\mathbf{\Theta}(\widehat{\mathbf{x}}, \widehat{\xi}) = \left\{\boldsymbol{\mu} = (\mathbf{A}_I^T - \mathbf{1}\mathbf{b}_I^T)\mathbf{u}_I + v_0\mathbf{\Sigma}\widehat{\mathbf{x}} : \mathbf{u}_I \geq \mathbf{0}, v_0 \geq 0\right\}.$$

Since $(\mathbf{0}, 0)$ is always feasible for (11), $0 \leq \boldsymbol{\mu}^T\widehat{\mathbf{x}} \equiv (\boldsymbol{\mu}^T\widehat{\boldsymbol{\phi}})/(\widehat{\boldsymbol{\phi}}^T\mathbf{\Sigma}\widehat{\boldsymbol{\phi}})^{\frac{1}{2}} \leq s(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \mathbf{\Theta}(\widehat{\mathbf{x}}, \widehat{\xi})$. Therefore, by Claim 1,

$$\mathbf{\Theta}(\widehat{\mathbf{x}}, \widehat{\xi}) = \left\{\boldsymbol{\mu} : s(\boldsymbol{\mu}) \geq 0, \boldsymbol{\mu}^T\widehat{\mathbf{x}} \equiv (\boldsymbol{\mu}^T\widehat{\boldsymbol{\phi}})/(\widehat{\boldsymbol{\phi}}^T\mathbf{\Sigma}\widehat{\boldsymbol{\phi}})^{\frac{1}{2}} = s(\boldsymbol{\mu})\right\} = \mathbf{\Theta}(\widehat{\boldsymbol{\phi}}).$$

■

Two interesting special cases are as follows:

(a) No side constraints, i.e. $(\mathbf{A}, \mathbf{b}) = (\mathbf{0}, \mathbf{0})$: In this case (12) reduces to

$$\begin{array}{rl} \text{minimize} & \big(v_0\mathbf{\Sigma}\widehat{\mathbf{x}} - \bar{\boldsymbol{\mu}}\big)^T\mathbf{\Sigma}^{-1}\big(v_0\mathbf{\Sigma}\widehat{\mathbf{x}} - \bar{\boldsymbol{\mu}}\big), \\ \text{subject to} & v_0 \geq 0. \end{array} \tag{13}$$

(b) No short sales, i.e. $(\mathbf{A}, \mathbf{b}) = (-\mathbf{I}, \mathbf{0})$: In this case (12) reduces to

$$\begin{array}{rl} \text{minimize} & \big(-\mathbf{u}_I + v_0\mathbf{\Sigma}\widehat{\mathbf{x}} - \bar{\boldsymbol{\mu}}\big)^T\mathbf{\Sigma}^{-1}\big(-\mathbf{u}_I + v_0\mathbf{\Sigma}\widehat{\mathbf{x}} - \bar{\boldsymbol{\mu}}\big), \\ \text{subject to} & \mathbf{u}_I \geq \mathbf{0}, \quad v_0 \geq 0. \end{array} \tag{14}$$

Next, we report on our preliminary computational experiments with this inverse optimization based investment policy. We invested in stocks belonging to the SP500 index for the period Jan. 1st, 1994 to Mar. 26th, 2002 and no short sales were permitted. At the starting date $t_0$ the investment policy randomly selected $n = 50$ stocks and managed the portfolio as follows:

(1) The investment period $p$ was set equal to $p = 40$ trading dates. At beginning of each period, we computed an estimate $\bar{\boldsymbol{\mu}}$ for $\boldsymbol{\mu}$ using a trailing window of $3p$ trading dates and an estimate $\bar{\mathbf{\Sigma}}$ using the entire history. If particular stock was removed from the SP500 we assumed that our investment in that stock was lost entirely.

(2) For a chosen $\alpha \in [0, 1)$, we constructed the $\alpha$-confidence set $C_\alpha = \left\{\boldsymbol{\mu} \in \mathbf{R}^n : (\boldsymbol{\mu} - \bar{\boldsymbol{\mu}})^T\bar{\mathbf{\Sigma}}^{-1}(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}) \leq \chi_p^2(\alpha)/p\right\}$, where $\chi_p^2(\alpha)$ denotes the $\alpha$-critical value of a $\chi^2$-distribution with $p$ degrees of freedom.

(3) Next, we solved (14) with $\mathbf{\Sigma} = \bar{\mathbf{\Sigma}}$. For optimal values of at most $\chi_p^2(\alpha)/p$, we did not rebalance; otherwise, we rebalanced to the Sharpe optimal portfolio corresponding to the estimate $\bar{\boldsymbol{\mu}}$.

The goal of our experiments was to study the value added by step (3), whether choosing an $\alpha > 0$ in markets with transaction costs yields an investment policy that improves over the naive policy (i.e. $\alpha = 0$), and if so, what is the improvement ? The performance of the strategy was studied for the rate of proportional transaction cost $\lambda = 0.05\%, 1\%, 2.5\%, 5\%$ and the confidence parameter $\alpha = 0, 0.05, 0.5, 0.95, 0.99$. The performance measure was the final wealth returned by the policy over $15p$ periods for an initial investment of \$1 on the starting date $t_0$. The values reported are averaged over $N = 100$ independent runs. (Each independent run chose 50 stocks randomly.)

The results for $t_0 =$ Nov. 1st, 1999 are reported in Table 1. This table is labeled "flat/down" market because the SP500 index was flat or moderately down. For each $\lambda$, the difference in performance between any pair $(\alpha \neq 0, \alpha = 0)$ is significant at 5%, except for the pair $(\alpha = 0.05, \alpha = 0)$ for $\lambda = 0.05$. The performance of the strategy improves on increasing $\alpha$ for all levels of transaction costs, i.e. being conservative in reacting to new information leads to higher returns.

Table 2 reports the results for $t_0 =$ Nov. 1st, 1994. Since the SP500 index was sharply rising in this period we labeled this table "up market". Here, for each $\lambda$, the difference in performance between any pair $(\alpha \neq 0, \alpha = 0)$ is significant at 5%. Again, the performance initially improves on increasing $\alpha$; however, it appears to marginally decrease when $\alpha$ is increased from 0.95 to 0.99. Although the decrease is not significant, it suggests that if one becomes too conservative one might not able to take advantage of short term opportunities that are present in an up-market.

## 4 Inverse quadratically constrained quadratic programming

A generic convex quadratically constrained quadratic program (QCQP) is given by

$$\begin{array}{rl} \text{minimize} & \mathbf{x}^T\mathbf{Q}_0\mathbf{x} + 2\mathbf{q}_0^T\mathbf{x}, \\ \text{subject to} & \mathbf{x}^T\mathbf{Q}_i\mathbf{x} + 2\mathbf{q}_i^T\mathbf{x} + \gamma_i \leq 0, \quad i = 1, \ldots, m, \end{array} \tag{15}$$

where $\mathbf{x} \in \mathbf{R}^n$ and $\mathbf{Q}_i \succeq \mathbf{0}$, $i = 1, \ldots, m$. The special case of (15) where all the constraints are linear, i.e. $\mathbf{Q}_i = \mathbf{0}$, $i = 1, \ldots, m$, appears in the context of linear quadratic control.

Our interest in the inverse problem associated with (15) is motivated by possible applications in system identification. Traditional system identification is focused on systems that are modeled by systems of differential or difference equations. However, improvements in computing has led to the development of systems whose dynamics are determined by real-time optimization. Identification procedures for such systems would find applications in identifying receding horizon controllers and other control techniques that rely on online optimization. [15] recently proposed such a technique for identifying a controller that at every time step $t$ solves the LP

$$\begin{array}{rl} \text{minimize} & \mathbf{c}^T\mathbf{x}, \\ \text{subject to} & \mathbf{A}_t\mathbf{x} = \mathbf{b}_t, \\ & \mathbf{x} \geq \mathbf{0}, \end{array}$$

where the objective vector $\mathbf{c}$ is a constant but the constraints $(\mathbf{A}_t, \mathbf{b}_t)$ are time dependent. The problem of characterizing the set $\mathcal{C}$ of possible vectors $\mathbf{c}$ consistent with an observed sequence of constraints $\{(\mathbf{A}_t, \mathbf{b}_t) : t \geq 1\}$ and corresponding controller decisions $\{\widehat{\mathbf{x}}_t : t \geq 1\}$ reduces to solving a series of inverse LPs. If instead the controller were assumed to be a linear quadratic controller, i.e. it solved a QP to determine the controls, the identification problem reduces to solving a sequence of inverse QPs or more generally inverse QCQPs.

Let $\mathbf{Q}_i = \mathbf{V}_i^T\mathbf{V}_i$ for some $\mathbf{V}_i \in \mathbf{R}^{p_i\times n}$. Then

$$\mathbf{x}^T\mathbf{Q}_i\mathbf{x} + 2\mathbf{q}_i^T\mathbf{x} + \gamma_i \leq 0 \quad \Leftrightarrow \quad \begin{bmatrix} \frac{1}{2}(1 - 2\mathbf{q}_i^T\mathbf{x} - \gamma_i) \\ \frac{1}{2}(1 + 2\mathbf{q}_i^T\mathbf{x} + \gamma_i) \\ \mathbf{V}_i\mathbf{x} \end{bmatrix} \succeq_{\mathcal{K}_i} \mathbf{0},$$

where $\mathcal{K}_i \subset \mathbf{R}^{p_i+2}$ is an SOC. Thus, the forward problem is equivalent to the conic program

$$\begin{array}{ll} \text{minimize} & \mathbf{x}^T\mathbf{Q}_0\mathbf{x} + 2\mathbf{q}_0^T\mathbf{x}, \\ \text{subject to} & \begin{bmatrix} \frac{1}{2}(1 - 2\mathbf{q}_i^T\mathbf{x} - \gamma_i) \\ \frac{1}{2}(1 + 2\mathbf{q}_i^T\mathbf{x} + \gamma_i) \\ \mathbf{V}_i\mathbf{x} \end{bmatrix} \succeq_{K_i} \mathbf{0}, \quad i = 1, \ldots, m, \end{array} \tag{16}$$

In typical applications it is easy to establish that (15), or equivalently (16), has a strictly feasible solution, the gradient $\nabla f = 2(\mathbf{Q}_0\mathbf{x} + \mathbf{q}_0)$ of the objective $f$ in (16) is affine in the unknown parameters $\mathbf{c} = (\mathbf{Q}_0, \mathbf{q}_0)$, and $\mathbf{int}(\mathcal{K}_2) \neq \emptyset$. Thus, Assumption 1 holds. Therefore, Lemma 1 and Lemma 2 imply that the set $\mathcal{Q}$ of parameters $(\mathbf{q}_0, \mathbf{Q}_0)$ consistent with the observed optimal solution $\widehat{\mathbf{x}}$ is given by

$$\mathcal{Q} = \mathbf{\Theta}(\widehat{\mathbf{x}}) \cap \big\{(\mathbf{q}_0, \mathbf{Q}_0) : \mathbf{Q}_0 \succeq \mathbf{0}\big\}. \tag{17}$$

In the context of identifying a linear quadratic receding horizon controller, (17) implies that the set of admissible objective matrix $(\mathbf{Q}_0, \mathbf{q}_0)$ is given by a collection of SOC constraints, one for each decision epoch of the controller, and one semidefinite constraint.

## 5 Identification of utility functions

Consider a data network $\mathcal{N}$ with $n$ nodes and $m$ arcs. Let $\mathbf{c} \in \mathbf{R}_+^m$ denote the capacity of the $m$ arcs and let $\mathbf{A} \in \mathbf{R}^{n\times m}$ denote the node-arc incidence matrix of the network [1].

Consider a user of this network who wishes to send flow from a designated node $s$ to another node $t$ (this flow may be spread over a number of paths). Let $\mathbf{x} \in \mathbf{R}_+^m$ denote the flow on each arc, and let $y$ denote the total flow of $\mathbf{x}$ from $s$ to $t$. Then the set $\mathcal{F}$ of feasible pairs $(y, \mathbf{x})$ is given by

$$\mathcal{F} = \{(y, \mathbf{x}) : \mathbf{A}\mathbf{x} = y\mathbf{d}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{c}\},$$

where $\mathbf{d} \in \mathbf{R}^m$ with $d(s) = -1$, $d(t) = +1$ and $d(j) = 0$, $j \notin \{s, t\}$. Let $\bar{y} = \max\{y : (y, \mathbf{x}) \in \mathcal{F}\}$ denote the maximum $s$-$t$ flow in this network.

Consider a user of this network who wishes to send flow from $s$ to $t$. Let $r(y) = \sum_{k=1}^{l} \alpha_k^* y^k$ denote the reward accrued to the user from a total $s$–$t$ flow $y$. The function $r(y)$ is assumed to be non-decreasing and concave over the interval $[0, \bar{y}]$. Let $s : \mathbf{R}^m \mapsto \mathbf{R}_+$ denote the price charged by the network to route a certain arc flow vector. Then the utility $U(y, \mathbf{x})$ derived by the user from the flow $(y, \mathbf{x}) \in \mathcal{F}$ is given by $U(y, \mathbf{x}) = r(y) - s(\mathbf{x})$.

We assume that the network administrator knows the form of the reward function $r(\cdot)$, i.e. the maximum degree $l$ is known, and also that $r(\cdot)$ is non-decreasing and concave; however, the precise values of parameters $\boldsymbol{\alpha}^*$ are not known. In this section, we describe an inverse optimization based method for estimating $\boldsymbol{\alpha}^*$ by suitably controlling the price function $s(\cdot)$.

Although the problem of learning the objective function of an optimization problem by setting parameters is of significant practical importance, it has not received much attention. Biel and Wein [9] discuss an inverse optimization based multi-round procurement auction mechanism where the auctioneer uses a slightly different scoring rule in each round to learn the suppliers' cost functions. Eppstein [14] briefly mentions using inverse optimization to learn player strategies in a game setting.

Since $r(\cdot)$ is concave, it follows that $r'(y) \geq 0$, for all $y \in [0, \bar{y}]$, iff $r'(\bar{y}) \geq 0$. Thus, the initial estimate $\boldsymbol{\Theta}^{(0)}$ of the parameters $\boldsymbol{\alpha}^*$ is given by

$$\boldsymbol{\Theta}^{(0)} = \Big\{ \boldsymbol{\alpha} : r'(\bar{y}) = \sum_{k=1}^{l} k\alpha_k \bar{y}^{k-1} \geq 0, \text{ and } r''(y) = \sum_{k=2}^{l} k(k-1)\alpha_k y^{k-2} \leq 0, \forall y \in [0, \bar{y}] \Big\}.$$

It is well known [21, 8] that the polynomial $r''(y) = \sum_{k=0}^{l-2}(k+1)(k+2)\alpha_{k+2} y^k \leq 0$ for all $y \in [0, \bar{y}]$ iff there exists $\mathbf{U} = [u_{ij}]_{i,j=1,\ldots,l-2} \in \mathbf{R}^{(l-2)\times(l-2)}$, $\mathbf{U} \succeq \mathbf{0}$ such that

$$\begin{array}{rcll} 0 & = & \sum_{i,j:i+j=2k-1} u_{ij}, & k = 0, \ldots, l-2, \\ -\sum_{m=0}^{k} \begin{pmatrix} l-2-m \\ k-m \end{pmatrix} (m+1)(m+2)\alpha_{m+2}\bar{y}^m & = & \sum_{i,j:i+j=2k} u_{ij}, & k = 0, \ldots, l-2. \end{array} \tag{18}$$

Thus, $\boldsymbol{\Theta}^{(0)} = \Big\{ \boldsymbol{\alpha} \in \mathbf{R}^k : \sum_{k=1}^{l} k\alpha_k \bar{y}^{k-1} \geq 0, \exists \mathbf{U} \succeq \mathbf{0} \text{ satisfying } (18) \Big\}$.

Suppose the administrator sets a convex price function $s_1(\cdot)$ and observes a flow $\widehat{\mathbf{x}}$. Since the user sets the flow by solving the convex optimization problem

$$\begin{array}{ll} \text{maximize} & U(y, \mathbf{x}) = r(y) - s_1(\mathbf{x}) \\ \text{subject to} & (y, \mathbf{x}) \in \mathcal{F}, \end{array} \tag{19}$$

the KKT conditions imply that there exist $\boldsymbol{\lambda} \in \mathbf{R}^m_+$, $\boldsymbol{\gamma} \in \mathbf{R}^n$ such that

$$\begin{array}{rcl} \widehat{\mathbf{r}}^T \boldsymbol{\alpha}^* & = & \gamma_s - \gamma_t, \\ \nabla s(\widehat{\mathbf{x}})_i & \geq & \mathbf{A}_i^T \boldsymbol{\gamma} - \lambda_i, \quad i \in \mathcal{R}_0, \\ \nabla s(\widehat{\mathbf{x}})_i & = & \mathbf{A}_i^T \boldsymbol{\gamma} - \lambda_i, \quad i \notin \mathcal{R}_0, \end{array} \tag{20}$$

where $\widehat{\mathbf{r}} = [1, 2\widehat{y}, 3\widehat{y}^2, \ldots, l\widehat{y}^{(l-1)}]^T$, $(\widehat{y}, \widehat{\mathbf{x}}) \in \mathcal{F}$, $\mathcal{R}_0 = \{i : \widehat{x}_i = 0\}$ and $\mathbf{A}_i$ denotes the column of $\mathbf{A}$ corresponding to the arc $i$. Thus, the new estimate for $\boldsymbol{\alpha}^*$ is given by

$$\boldsymbol{\Theta}^{(1)} = \Big\{ \boldsymbol{\alpha} \in \boldsymbol{\Theta}^{(0)} : \exists \boldsymbol{\gamma} \in \mathbf{R}^n, \boldsymbol{\lambda} \in \mathbf{R}^m_+ \text{ satisfying } (20) \Big\}. \tag{21}$$

Repeating this process with different cost functions $s(\mathbf{x})$ leads to improved localization of the parameter $\boldsymbol{\alpha}^*$.

Next, we illustrate this technique on a simple example. The network $\mathcal{N}$ consists of $n = 2$ nodes and $m = 4$ parallel arcs. In this case the maximum flow $\bar{y} = \mathbf{1}^T\mathbf{c}$. The reward function $r(y)$ is assumed to be quadratic (i.e. $r(y) = \alpha_1^* y + \alpha_2^* y^2$ and $l = 2$). Thus, the initial estimate $\boldsymbol{\Theta}^{(0)}$ of the parameters $\boldsymbol{\alpha}^*$ is given by $\boldsymbol{\Theta}^{(0)} = \Big\{(\alpha_1, \alpha_2) : \alpha_2 \leq 0,\ \alpha_1 \geq -2\bar{y}\alpha_2\Big\}$.

In order to improve the estimate of $\boldsymbol{\alpha}^*$ the network administrator randomly sets the price vector $\widehat{\mathbf{s}} \in \mathbf{R}_+^m$ (i.e. the pricing function is $s(\mathbf{A}\mathbf{x}) = \widehat{\mathbf{s}}^T\mathbf{x}$). Next, the user sets the flows $\mathbf{x}$ by solving

$$\begin{array}{rl} \max & \alpha_1^* y + \alpha_2^* y^2 - \widehat{\mathbf{s}}^T\mathbf{x}, \\ \text{subject to} & y = \mathbf{1}^T\mathbf{x}, \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{c}. \end{array} \tag{22}$$

In this special case the update (21) can be simplified as follows:

Case (i) $\exists i$ such that $0 < \widehat{x}_i < c_i$: $\boldsymbol{\Theta}^{(1)} = \{\boldsymbol{\alpha} \in \boldsymbol{\Theta}^{(0)} : \widehat{\mathbf{r}}^T\boldsymbol{\alpha} = \widehat{s}_i\}$.

Case (ii) $\nexists i$ such that $0 < \widehat{x}_i < c_i$: $\boldsymbol{\Theta}^{(1)} = \big\{\boldsymbol{\alpha} \in \boldsymbol{\Theta}^{(0)} : \max_{\{i:\widehat{x}_i=0\}}\{\widehat{s}_i\} \leq \widehat{\mathbf{r}}^T\boldsymbol{\alpha} \leq \min_{\{i:\widehat{x}_i=c_i\}}\{\widehat{s}_i\}\big\}$.

This process is repeated until the set $\boldsymbol{\Theta}$ reduces to a point, i.e. the parameter $\boldsymbol{\alpha}^*$ is perfectly estimated, or when the administrator is satisfied with $\boldsymbol{\Theta}$.

Tables 3 and 4 display the progress of the estimation algorithm on two instances of the problem. For the instance displayed in Table 3, the capacity vector $\mathbf{c} = \mathbf{1}$, the true (unknown) parameter $\boldsymbol{\alpha}^* = (10, -1)$ and the random price vectors $\widehat{\mathbf{s}}$ were generated using the MATLAB function $\widehat{\mathbf{s}}$ `= unidrnd(50,4,1)/10`. The function `unidrnd(z,4,1)`, $z \in \mathbf{Z}$, generates vectors in $\mathbf{R}^4$ with each component uniformly chosen from the set $\{1, 2, \ldots, z\}$. In each iteration, the estimate $\boldsymbol{\Theta}$ was simplified as far as possible. For the instance in Table 3, the estimation algorithm correctly estimates the parameter in 3 iterations. For the instance displayed in Table 4, the capacity vector $\mathbf{c} = [2, 1, 2, 1]^T$, the true (unknown) parameter $\boldsymbol{\alpha}^* = (25, -2)$ and the random price vectors $\widehat{\mathbf{s}}$ were generated using the MATLAB function $\widehat{\mathbf{s}}$ `= unidrnd(190,4,1)/5`. For this instance, the estimation algorithm correctly estimates the parameter in 6 iterations.

## 6 Conclusion

In this paper we extend the inverse mathematical programming methodology to include inverse conic programs. Since a very wide class of convex optimization problems can be modeled as conic programs [4, 7], this extension significantly expands the class of efficiently solvable inverse problems. We present three applications of inverse conic programs. We would like to especially draw the reader's attention to the application in Section 5 where a network administrator attempts to learn a users' utility function by manipulating the price of arcs. We discuss the case where the price vector is randomly generated. Clearly, one can do better if one can choose a price vector $\widehat{\mathbf{s}}_k$ that "maximally" reduces the size of $\boldsymbol{\Theta}^{(k)}$. At this time we are not aware of any strategy that is guaranteed to do better than the random strategy.

We would also like to reiterate that we consider the case where only the objective function is uncertain. Inverse optimization in its full generality remains an interesting and open problem.

## References


[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows.* Prectice Hall, 1993.

[2] R. K. Ahuja and J. B. Orlin. Inverse optimization. *Oper. Res.*, 49:771–783, 2001.

[3] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5(1):13–51, 1995.

[4] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Math. Prog. Ser. B*, 2003.

[5] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Math. Oper. Res.*, 23(4):769–805, 1998.

[6] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Oper. Res. Lett.*, 25(1):1–13, 1999.

[7] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001. Analysis, algorithms, and engineering applications.

[8] D. Bertsimas and I. Popescu. Optimal inequalities in probability: A convex optimization approach. Submitted to *Math. Prog.*

[9] D. R. Biel and L. M. Wein. An inverse-optimization-based auction mechanism to support a multiattribute RFQ process. *Mgmt. Sc.*, 49:1529–1545, 2003.

[10] D. Burton and P-L Toint. On an instance of inverse shortest path problems. *Math. Prog.*, 53:45–61, 1992.

[11] D. Burton and P-L Toint. On the inverse shortest path algorithm for recovering linearly correlated costs. *Math. Prog.*, 55:1–22, 1994.

[12] R. Dial. Minimal revenue congestion pricing. Part I: A fast algorithm for the single-origin case. *Transportation Res. Ser. B*, 33:189–202, 1999.

[13] R. Dial. Minimal revenue congestion pricing. Part II: An efficient algorithm for the general case. *Transportation Res. Ser. B*, 34:645–665, 2000.

[14] D. Eppstein. Setting parameters by example. *SIAM J. Comput.*, 32(3):643–653, 2003.

[15] S. Gentry, V. Saligrama, and E. Feron. Dynamic inverse optimization. In *Proc. ACC*, pages 4722–7, 2001.

[16] D. Goldfarb and G. Iyengar. Robust portfolio selection problems. *Math. Oper. Res.*, 28(1):1–37, 2003.

[17] C. Heuberger. Inverse combinatorial optimization: A survey of problems, methods, and results. To appear in *J. of Comb. Opt.*, 2003.

[18] C-F Huang and R. H. Litzenberger. *Foundations for Financial Economics.* Prentice Hall, 1998.

[19] G. Iyengar and W. Kang. Inverse convex optimization. Technical Report TR-2003-02, IEOR Dept., Columbia University, 2003. Available at http://www.corc.ieor.columbia.edu/reports/techreports/.

[20] R. O. Michaud. *Efficient Asset Management: a practical guide to stock portfolio management and asset allocation.* Financial Management Association survey and synthesis series. HBS Press, 1998.

[21] Y. Nesterov. Structure of non-negative polynomials and optimization problems. Technical Report 9749, CORE, Universite Catholique de Louvain, 1997.

[22] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming.* SIAM, Philadelphia, 1993.

[23] G. Nolet. *Seismic Tomography.* Reidel, 1987.

[24] P. T. Sokkalingam, R. K. Ahuja, and J. B. Orlin. Solving inverse spanning tree problem through network flow techniques. *Oper. Res.*, 47(2):291–298, 1999.

[25] A. Tarantola. *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation.* Elsevier, Amsterdam, 1987.

[26] J. Zhang and M. Cai. Inverse problem of minimum cuts. *Math. Methods in OR*, 48:51–58, 1998.

[27] J. Zhang and Z. Liu. Caluculating some inverse programming problems. *J. Comp. Appl. Math.*, 72:261–273, 1996.

| | $\lambda = 0.05\%$ | $\lambda = 1\%$ | $\lambda = 2.5\%$ | $\lambda = 5\%$ |
|---|---|---|---|---|
| $\alpha = 0$ | 0.7719 | 0.7181 | 0.5769 | 0.3978 |
| $\alpha = 0.05$ | 0.7582 | 0.7536 | 0.7401 | 0.7184 |
| $\alpha = 0.50$ | 0.7872 | 0.7855 | 0.7805 | 0.7724 |
| $\alpha = 0.95$ | 0.8136 | 0.8128 | 0.8102 | 0.8058 |
| $\alpha = 0.99$ | 0.8157 | 0.8150 | 0.8129 | 0.8093 |

Table 1: Return on \$1 over 15 periods starting on $t_0 =$ Nov. 1st 1999 (flat/down market)

| | $\lambda = 0.05\%$ | $\lambda = 1\%$ | $\lambda = 2.5\%$ | $\lambda = 5\%$ |
|---|---|---|---|---|
| $\alpha = 0$ | 1.4005 | 1.3142 | 1.0842 | 0.7825 |
| $\alpha = 0.05$ | 1.7486 | 1.7462 | 1.7390 | 1.7270 |
| $\alpha = 0.5$ | 1.7534 | 1.7521 | 1.7481 | 1.7415 |
| $\alpha = 0.95$ | 1.7532 | 1.7520 | 1.7485 | 1.7426 |
| $\alpha = 0.99$ | 1.7509 | 1.7503 | 1.7482 | 1.7448 |

Table 2: Return on \$1 over 15 periods starting on $t_0 =$ Nov. 1st 1994 (up market)

| Iter $k$ | price vector | estimate $\mathbf{\Theta}^{(k)}$ |
|---|---|---|
| 0 | | $\left\{(\alpha_1, \alpha_2) : \alpha_2 \leq 0, \alpha_1 \geq -8\alpha_2\right\}$ |
| 1 | [4.2 1.6 1.9 0.1] | $\left\{(\alpha_1, \alpha_2) : \alpha_2 \leq 0, \alpha_1 \geq -8\alpha_2, 3.5 \leq \alpha_1 + 6\alpha_2 \leq 4.2\right\}$ |
| 2 | [1.0 4.4 3.5 1.0] | $\left\{(\alpha_1, \alpha_2) : -1.4 \leq \alpha_2 \leq 0, \alpha_1 + 6.5\alpha_2 = 3.5\right\}$ |
| 3 | [4.3 1.9 3.5 3.0] | $\left\{(\alpha_1, \alpha_2) : \alpha_1 + 6.5\alpha_2 = 3.5, \alpha_1 + 5.7\alpha_2 = 4.3\right\} = \left\{(10, -1)\right\}$ |

Table 3: Utility estimation $\big(m = 4,\ \mathbf{c} = \mathbf{1},\ \boldsymbol{\alpha}^* = (10, -1)\big)$

| Iter $k$ | price vector | estimate $\mathbf{\Theta}^{(k)}$ |
|---|---|---|
| 0 | | $\left\{(\alpha_1, \alpha_2) : \alpha_2 \leq 0, \alpha_1 \geq -8\alpha_2\right\}$ |
| 1 | [18.0 16.6 18.8 1.2] | $\left\{(\alpha_1, \alpha_2) : \alpha_2 \leq 0, \alpha_1 \geq -8\alpha_2, 9.2 \leq \alpha_1 + 6\alpha_2 \leq 15.4\right\}$ |
| 2 | [5.6 9.0 15.4 9.0] | $\left\{(\alpha_1, \alpha_2) : \alpha_2 \leq 0, \alpha_1 \geq -8\alpha_2, 12.4 \leq \alpha_1 + 6\alpha_2 \leq 15.4\right\}$ |
| 3 | [18.4 7.2 4.0 18.8] | $\left\{(\alpha_1, \alpha_2) : \begin{array}{l} \alpha_2 \leq 0, \alpha_1 \geq -8\alpha_2, \\ 8.4 \leq \alpha_1 + 4\alpha_2 \leq 18 \\ 12.4 \leq \alpha_1 + 6\alpha_2 \leq 15.4 \end{array}\right\}$ |
| 4 | [9.2 12.4 8.4 16.4] | $\left\{(\alpha_1, \alpha_2) : \begin{array}{l} \alpha_2 \leq 0, \alpha_1 \geq -8\alpha_2, \\ 8.4 \leq \alpha_1 + 4\alpha_2 \leq 18 \\ 12.4 \leq \alpha_1 + 6\alpha_2 \leq 15.4 \\ 7.2 \leq \alpha_1 + 8\alpha_2 \leq 16.4 \end{array}\right\}$ |
| 5 | [0.4 9.4 0.4 16.0] | $\left\{(\alpha_1, \alpha_2) : -2.1754 \leq \alpha_2 \leq 0, \alpha_1 + 9.7\alpha_2 = 5.6\right\}$ |
| 6 | [18.0 0.2 15.0 8.4] | $\left\{(\alpha_1, \alpha_2) : \alpha_1 + 9.7\alpha_2 = 5.6, \alpha_1 + 7.8\alpha_2 = 9.4\right\} = \left\{(25, -2)\right\}$ |

Table 4: Utility estimation $\left(m = 4,\ \mathbf{c} = [2, 1, 2, 1]^T,\ \boldsymbol{\alpha}^* = (25, -2)\right)$