

Design and implementation of a homogeneous interior-point method for conic programming involving exponential cone constraints

Yuan Gao¹

Supervised by

Prof. Kim-Chuan Toh¹

Prof. Melvyn Sim²

¹ Department of Mathematics, National University of Singapore

² Department of Decision Sciences, National University of Singapore

Honour's Project Introductory Talk
Oct 2016

Contents

- ▶ Introduction, motivation and preliminaries.
- ▶ Current implementation.
- ▶ Examples of application.
 - ▶ Optimizing the satisficing measure.*
 - ▶ Convex approximation of chance constrained problems.
 - ▶ Robust choice model.
- ▶ The algorithm.
 - ▶ The standard conic form.
 - ▶ A homogeneous self-dual model.
 - ▶ The central path and search directions.
 - ▶ Methods for solving the linear systems.*
 - ▶ Termination conditions.
- ▶ Plans for the next step.

* Optional contents. Will be briefly covered if time permits.

Introduction

- ▶ Many interesting real-world problems can be modeled as *convex optimization* problems.
 - ▶ More specifically, LP, SOCP and SDP problems.
- ▶ *In theory*, these problems can be *efficiently* solved by a class of algorithms known as *interior-point methods* (IPM).
 - ▶ Kamarkar (1984) first proposed and studied IPM for LP.
- ▶ Powerful solvers have been developed since then.
 - ▶ LP: Excel, Matlab, AMPL.
 - ▶ SOCP/MISOCP: Cplex, Gurobi.
 - ▶ SDP: SDPT3 (Toh, Todd & Tütüncü), SeDuMi (Sturm, Romanko & Pólik).

Motivation

- ▶ Recently, researchers in OR, Econometrics and EECS have been considering (convex) optimization models that cannot be formulated as LP, SOCP or SDP.
 - ▶ Only small and “nice” instances can be solved (through LP/SOCP approximation and calling the respective solvers).
- ▶ Models with log and exp functions in their objectives and constraints can be formulated as conic programming problems involving *exponential cone constraints*.
 - ▶ They can potentially be solved efficiently by IPM.

Motivation

- ▶ There have been a few works on both theoretical and computational aspects of IPM for *non-symmetric conic programming*.
 - ▶ Nesterov (2006).
 - ▶ Charles and Glineur (2009).
 - ▶ Ye and Skajaa (2015).
- ▶ We need a solver/program that efficiently solves problems involving exponential cone constraints.
 - ▶ An important class of non-symmetric conic programming problems.

Preliminaries

- ▶ Notations.

- ▶ $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}$.

- ▶ $\mathcal{Q}^p = \left\{ (x_1, \dots, x_p)^T \in \mathbb{R}^p \mid x_1 \geq \sqrt{x_2^2 + \dots + x_p^2} \right\}$.

- ▶ $\mathcal{K}_{\text{exp}}^0 = \left\{ (x, y, z)^T \in \mathbb{R}^3 \mid z > 0, \exp\left(\frac{x}{z}\right) \leq \frac{y}{z} \right\}$.

- ▶ $\mathcal{K}_{\text{exp}} = \text{closure}(\mathcal{K}_{\text{exp}}^0)$.

- ▶ It can be shown that

$$\mathcal{K}_{\text{exp}} = \mathcal{K}_{\text{exp}}^0 \cup (-\mathbb{R}_+) \times \mathbb{R}_+ \times \{0\}.$$

- ▶ More notations and definitions are needed when discussing the algorithm.

Current implementation

- ▶ We developed a program that *solves* problems coded in the following form

$$\begin{aligned} \min_{x_1, \dots, x_N} \quad & \sum_{i=1}^N c_i^T x_i \\ \text{s.t.} \quad & \sum_{i=1}^N A_i x_i = b, \quad x_i \in K_i, \forall i \end{aligned} \tag{P*}$$

where $K_i \subset \mathbb{R}^{n_i}$ is one of the following.

- ▶ nonnegative orthant $\mathbb{R}_+^{n_i}$.
 - ▶ product of second-order cones $\mathcal{Q}^{q_1} \times \dots \times \mathcal{Q}^{q_{k_i}}, \sum_{j=1}^{k_i} q_j = n_i$.
 - ▶ product of the exponential cone $(\mathcal{K}_{\text{exp}})^{k_i}, 3k_i = n_i$.
 - ▶ unrestricted space \mathbb{R}^{n_i} .
- ▶ To solve a general convex optimization model involving log and exp functions, it has to be converted into (P*).

Current implementation

- ▶ For $x_i \in \mathbb{R}^{n_i}$, let $x_i = x_i^+ - x_i^-$, where $x_i^+, x_i^- \in \mathbb{R}_+^{n_i}$.
- ▶ Therefore (P*) can be converted into the following *standard conic form*

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \quad x \in \mathcal{K} \end{aligned} \tag{P}$$

where $\mathcal{K} = \mathbb{R}^{n_l} \times (Q^{q_1} \times \dots \times Q^{q_m}) \times (\mathcal{K}_{\text{exp}})^h$,
 $n_q = q_1 + \dots + q_m$, $n_e = 3h$, $n = n_l + n_q + n_e$, $A \in \mathbb{R}^{m \times n}$,
 $x \in \mathbb{R}^n$.

Example: Asset allocation via satisficing measure

- ▶ Brown, Giorgi and Sim (2009) consider the following entropic prospective satisficing measure (EPSM) of a random variable \mathbf{X} ,

$$\rho(\mathbf{X}) = \sup \left\{ t \in \mathbb{R} \setminus \{0\} \mid \frac{1}{t} \log \mathbb{E} [\exp(-t\mathbf{X})] \leq 0 \right\}.$$

Example: Asset allocation via satisficing measure

- ▶ Suppose there are n assets with independent random returns $V_i \in \{v_i^l, v_i^h\}$, $v_i^l \leq v_i^h$, $\mathbb{P}(V_i = v_i^j) = p_i^q$, $q = h, l$, $p_i^h + p_i^l = 1$, $i = 1, \dots, n$.
- ▶ Consider the following asset allocation problem, with a target return $\tau < \max_{1 \leq j \leq n} \mathbb{E}(V_j)$,

$$\begin{aligned} \mathcal{O}^{EPSM} = & \\ & \sup_{w_1, \dots, w_n} \rho \left(\sum_{i=1}^n w_i V_i - \tau \right) \\ & \text{s.t. } \sum_{i=1}^n w_i = 1, \quad w_i \geq 0, \forall i. \end{aligned} \tag{1}$$

- ▶ This example is for illustration purpose.

Example: Asset allocation via satisficing measure

- Problem (1) can be formulated into (P^*) as follows.

$$\begin{aligned} \mathcal{O}' &= \min \quad a_1^l \\ \text{s.t.} \quad & \sum_{j=1}^n z_j = -\tau, \quad \sum_{j=1}^n w_j = 1, \\ & p_j^l s_j^l + p_j^h s_j^h - a_j = 0, \quad j = 1, \dots, n, \\ & u_j^q + z_j + v_j^q w_j = 0, \quad q = l, h, \quad j = 1, \dots, n, \\ & a_1^l - a_1^h = 0, \quad a_1^q - a_j^q = 0, \quad q = l, h, \quad j = 2, \dots, n, \end{aligned} \tag{2}$$

with decision variables

$$\begin{aligned} (u_j^q, s_j^q, a_j^q) &\in \mathcal{K}_{\text{exp}}, \quad q = l, h, \quad j = 1, \dots, n, \\ (w_1, \dots, w_n)^T &\in \mathbb{R}_+^n, \quad (z_1, \dots, z_n)^T \in \mathbb{R}^n. \end{aligned}$$

Example: Asset allocation via satisficing measure

- ▶ Note that $\mathcal{O}^{EPSM} = 1/\mathcal{O}' > 0$.
- ▶ To solve a feasible instance with $n = 1000$, our solver takes less than 3 minutes (on a modest laptop) while SOCP approximation becomes impractical.
 - ▶ The resulting SOCP instances are extremely large.
 - ▶ The approximate solution may not converge.

Example: Convex approximation of chance constrained problems

- ▶ Chance constrained problems arise naturally in OR models.
 - ▶ Chance constraints look like $\mathbb{P}\left(F\left(\mathbf{x}, \tilde{\xi}\right) \leq 0\right) \geq 1-\alpha$, where \mathbf{x} is the decision vector, $\tilde{\xi}$ is a random vector and α is a given “confidence level.”
- ▶ In general, they are very difficult to solve/approximate.

Example: Convex approximation of chance constrained problems

- ▶ The scenario approach (based on Monte Carlo simulation) can be used to approximate problems with “nice” structures (Nemirovski & Shapiro, 2005).
 - ▶ In general, the solution yielded is not feasible under the original chance constraints.
 - ▶ It is difficult to establish *risk bounds*.
 - ▶ A reasonably reliable solution requires a huge number of realizations of $\tilde{\xi}$.

Example: Convex approximation of chance constrained problems

- ▶ Nemirovski and Shapiro (2006) developed a *conservative* approximation scheme named the *Bernstein approach*, which has several advantages compared to the scenario approach.
 - ▶ Given a “nice” chance constrained problem (P1) with information about the random variables, define an associated convex optimization problem (P2).
 - ▶ (P2) can be solved efficiently (at least in theory).
 - ▶ The solution to (P2) is always feasible to (P1).
 - ▶ The optimal objective value of (P2) is (usually) suboptimal to (P1).

Example: Convex approximation of chance constrained problems

- Consider the following asset allocation problem

$$\begin{aligned} \mathcal{O}^{chance} = & \max_{\tau \in \mathbb{R}} (\tau - 1) \\ & x_0, x_1, \dots, x_n \geq 0 \\ \text{s.t. } & \mathbb{P} \left(\tau > \sum_{j=0}^n \tilde{r}_j x_j \right) \leq \alpha, \quad \sum_{j=0}^n x_j \leq 1 \end{aligned} \tag{3}$$

where \tilde{r}_j is the random return of asset j , $j = 0, \dots, n$ and $\alpha \in [0, 1]$ denotes the confidence level.

Example: Convex approximation of chance constrained problems

- ▶ We characterize \tilde{r}_j , $j = 0, \dots, n$ as follows.
 1. The returns satisfy $\tilde{r}_0 = r_0 = 1$, $\mathbb{E}(\tilde{r}_j) = 1 + \rho_j$, $j = 1, \dots, n$.
 2. For $j = 1, \dots, n$, $l = 1, \dots, q$, let $\tilde{r}_j = \tilde{\eta}_j + \sum_{l=1}^q \gamma_{jl} \tilde{\zeta}_l$ where $\tilde{\eta}_j \sim \mathcal{LN}(\mu_j, \sigma_j^2)$, $\tilde{\zeta}_l \sim \mathcal{LN}(\nu_l, \theta_l^2)$ (\mathcal{LN} denotes lognormal distribution).
 3. All $\tilde{\eta}_j$ and $\tilde{\zeta}_l$ are mutually independent.
 4. The parameters ρ_j , μ_j , σ_j , ν_j , θ_j , γ_{jl} satisfy

$$\mu_j, \nu_j, \gamma_{jl} \geq 0, \quad 0 \leq \rho_1 \leq \dots \leq \rho_n,$$

$$\mathbb{E} \left[\sum_{l=1}^q \gamma_{jl} \tilde{\zeta}_l \right] = \sum_{l=1}^q \gamma_{jl} \exp \left(\nu_l + \frac{\theta_l^2}{2} \right) = \frac{\rho_j}{2},$$

$$\mathbb{E} [\tilde{\eta}_j] = \exp \left(\mu_j + \frac{\sigma_j^2}{2} \right) = 1 + \frac{\rho_j}{2}, \quad j = 1, \dots, n.$$

Example: Convex approximation of chance constrained problems

- ▶ To apply the approximation scheme, the \mathcal{LN} random variables $\tilde{\eta}_j$, $\tilde{\gamma}_l$ need to be discretized (rounded from below) so that their moment generating functions (MGF) are well defined.
- ▶ Assume this has been done without change of notation.

Example: Convex approximation of chance constrained problems

- ▶ Let $q = n + d$, $\bar{\mathbf{x}} = (x_0, x_1, \dots, x_n)$,

$$g_0(\bar{\mathbf{x}}) = \tau - x_0,$$

$$\tilde{\xi}_j = \tilde{\eta}_j, \quad g_j(\bar{\mathbf{x}}) = -x_j, \quad j = 1, \dots, n,$$

$$\tilde{\xi}_{n+l} = \tilde{\zeta}_l, \quad g_{n+l}(\bar{\mathbf{x}}) = -\sum_{j=1}^n \gamma_{jl} x_j, \quad l = 1, \dots, q.$$

- ▶ For each j , $k = 1, \dots, N_j$, let $\tilde{\xi}_j \in \{v_k^j \mid k = 1, \dots, N_j\}$ and $\mathbb{P}(\tilde{\xi}_j = v_k^j) = p_k^j$. Let $M_j : z \rightarrow \sum_{k=1}^{N_j} p_k^j \exp(v_k^j z)$ be the MGF of $\tilde{\xi}_j$ and denote $\Lambda_j(\cdot) = \log M_j(\cdot)$.

- ▶ Values of v_k^j , p_k^j are determined by the parameters of the \mathcal{LN} distributions and the discretization scheme.

Example: Convex approximation of chance constrained problems

- The Bernstein approximation to (3) is

$$\begin{aligned} \mathcal{O}^{Bernstein} = & \max_{\tau \in \mathbb{R},} (\tau - 1) \\ & \bar{\mathbf{x}} = (x_0, x_1, \dots, x_n)^T \geq 0 \\ \text{s.t. } & \inf_{t > 0} \left(g_0(\bar{\mathbf{x}}) + \sum_{j=1}^d t \Lambda_j(t^{-1} g_j(\bar{\mathbf{x}})) - t \log \alpha \right) \leq 0, \\ & \sum_{j=0}^n x_j \leq 1. \end{aligned} \quad (4)$$

- It can be shown that (4) can be formulated into (P*) as

$$\begin{aligned}
 \mathcal{O}^P &= \min -\tau \\
 \text{s.t. } & x_0 + \sum_{j=1}^n x_j + s_x = 1, \quad g_0 + \left(\sum_{j=1}^d s_j \right) - (\log \alpha) t_0 = 0, \\
 & g_0 - \tau + x_0 = 0, \quad g_j + x_j = 0, \quad j = 1, \dots, n, \\
 & g_{n+l} + \sum_{j=1}^n \gamma_{jl} x_j = 0, \quad l = 1, \dots, q, \\
 & w_k^j - v_k^j g_j + s_j = 0, \quad j = 1, \dots, d, \quad k = 1, \dots, N_j, \\
 & \sum_{k=1}^{N_j} p_k^j u_k^j - t_0 = 0, \quad j = 1, \dots, d, \\
 & t_0 - t_k^j = 0, \quad j = 1, \dots, d, \quad k = 1, \dots, N_j.
 \end{aligned} \tag{5}$$

with decision variables

$$\tau \in \mathbb{R}$$

$$x_0, x_1, \dots, x_n, s_x \geq 0$$

$$g_0, g_1, \dots, g_d \in \mathbb{R}$$

$$t_0 \geq 0$$

$$s_1, \dots, s_d \in \mathbb{R}$$

$$(w_k^j, u_k^j, t_k^j) \in \mathcal{K}_{\text{exp}}, j = 1, \dots, d, k = 1, \dots, N_j.$$

Example: Convex approximation of chance constrained problems

- ▶ Define $\mathcal{O}^{nominal} = \max \{\rho_i \mid i = 1, \dots, n\} = \rho_n$.
 - ▶ This is the optimal objective of (3) with all \tilde{r}_i replaced by their respective means.
- ▶ Note that $\mathcal{O}^{nominal} \geq \mathcal{O}^{chance} \geq \mathcal{O}^{Bernstein} = -\mathcal{O}^P - 1$.

Example: Convex approximation of chance constrained problems

- ▶ For a randomly generated instance of (4) with $n = 100$, $q = 4$, $\alpha = 0.05$, $\epsilon = 10^{-5}$, $\Delta = 0.005$ and appropriate values of distributional parameters, the resulting standard form (P') has $A \in \mathbb{R}^{2187 \times 3491}$, $\text{density}(A) = 0.0012058$, $N_l = 539$, $N_q = 0$, $N_e = 2952$.
- ▶ The dimension of (the exponential cone part of) A depends chiefly on the “shape” of the \mathcal{LN} distributions and precision of the discretization.

Example: Convex approximation of chance constrained problems

- ▶ Our solver took 13.65 seconds to obtain an “optimal” solution.
- ▶ CVX (Grant and Boyd, 2013) took 255.65 seconds before termination *without* a solution.
 - ▶ It is the most well developed algebraic modeling tool for specifying and solving convex optimization problems.
 - ▶ It used successive approximation and called SDPT3 several times to solve the resulting SOCP instances.
 - ▶ These SOCP instances have very large dimensions.
 - ▶ SOCP approximation scheme is not reliable in general, although SOCP solvers are very efficient and robust.
- ▶ For smaller instances, CVX and our solve give the same solutions.

Example: Convex approximation of chance constrained problems

- ▶ The solution obtained through Bernstein approximation might be too conservative.
 - ▶ The solution is *too* reliable while the objective value is not satisfactory.
- ▶ In practice, *tuning* might be necessary.
 - ▶ Set a large α' in (P1).
 - ▶ Find a solution to (P2) and approximate the *empirical risk* α^* of the solution through simulation.
 - ▶ Vary α' until α^* is “close” to α .

Example: Robust choice model

- ▶ Consider a customer and J types of crackers.
- ▶ Each type of crackers has Q attributes.
 - ▶ For example, price, net weight, whether it is displayed in a conspicuous spot.
 - ▶ Attributes might be *collinear* (correlated).
- ▶ The probability of the customer choosing type j is

$$p_j = \frac{\exp(\alpha_j + \mathbf{x}'_j \boldsymbol{\beta})}{\sum_{l=1}^J \exp(\alpha_l + \mathbf{x}'_l \boldsymbol{\beta})}$$

where \mathbf{x}_j denotes a vector of attribute values.

Example: Robust choice model

- ▶ Suppose we have some data.
 - ▶ There have been N (independent) purchases.
 - ▶ For each purchase, the attributes of all types of crackers $\mathbf{X} = \{\mathbf{x}_{nj}\} \subset \mathbb{R}^Q$ and the customer's choice $\mathbf{Y} = (y_{nj}) \subset \{0, 1\}^{N \times J}$ are recorded.
- ▶ We want to find $\hat{\alpha}, \hat{\beta}$, estimates of α, β .
 - ▶ The classical approach is maximum-likelihood estimation, which is thoroughly discussed in Chapter 3 of Train's (2009) book *Discrete Choice Methods with Simulation*.

Example: Robust choice model

- ▶ We find $\hat{\alpha}, \hat{\beta}$ by solving the following optimization problem

$$\mathcal{O}^{rc} = \max_{\alpha, \beta} \min_{(z_{nj}) \in \mathcal{Z}(\Gamma)} \sum_{n=1}^N \sum_{j=1}^J z_{nj} \log \frac{\exp(\alpha_j + \mathbf{x}'_{nj}\beta)}{\sum_{l=1}^J \exp(\alpha_l + \mathbf{x}'_{nl}\beta)} \quad (6)$$

where

$$\mathcal{Z}(\Gamma) = \left\{ (z_{nj}) \in \mathbb{R}_+^{N \times J} \mid \begin{array}{l} \sum_{j=1}^J z_{nj} = 1, \forall n, \\ \sum_{n=1}^N z_{n\hat{j}_n} \geq N - \Gamma \end{array} \right\}$$

and \hat{j}_n is such that $y_{n\hat{j}_n} = 1$ ($y_{nj} = 0$ for $j \neq \hat{j}_n$).

- ▶ The parameter Γ accounts for “irrational” choices.
- ▶ Letting $\Gamma > 0$ usually yields “better” estimates by allowing customs to depart from their usual behavioral patterns occasionally.

Example: Robust choice model

- By taking the dual of the inner minimization problem which is LP on (z_{nj}) , it can be shown that (6) is *equivalent* to

$$\begin{aligned}\mathcal{O}^{rc} &= \\ &\max_{(a_n), b, \alpha, \beta} \left(\sum_{n=1}^N a_n \right) + (N - \Gamma) b \\ \text{s.t.} \quad &a_n + \mathbb{I}_{\{j=\hat{j}_n\}} \cdot b \leq -\log \sum_{l=1}^J \exp \left(\alpha_l - \alpha_j + (\mathbf{x}_{nl} - \mathbf{x}_{nj_n})' \beta \right), \\ &n = 1, \dots, N, \quad j = 1, \dots, J.\end{aligned}\tag{7}$$

Example: Robust choice model

- ▶ Problem (7) can be formulated into (P^*) and hence solved by our solver.
- ▶ An instance of $N = 3000$ ($J = 4$, $Q = 3$) can be solved to satisfactory accuracy in 10 hours.
- ▶ CVX can only handle instances with $N \leq 150$, which can be loaded and solved in minutes by our solver.

Example: Robust choice model

- ▶ Numerical experiments with real purchase record data.
 - ▶ Fix two sets S_1, S_2 of distinct observations with $|S_1| = |S_2| = 250$.
 - ▶ Use S_1 to compute the estimates $\hat{\alpha}, \hat{\beta}$ and compute the *likelihood* of them given S_2 .

Example: Robust choice model

- ▶ When some of the data departs from the distributional assumption, a positive Γ gives “better” estimates.
- ▶ This model has several (potential) advantages, given that the difficulty in computation is (partially) addressed.
 - ▶ A systematic way to tune the parameter?
 - ▶ How to assess *goodness-of-fit*?
 - ▶ Any theoretical justification for the improved performance?
 - ▶ Connection to regularized regression and other models (Shafieezadeh-Abadeh, Esfahani and Kuhn, 2015)?

The algorithm

- Recall the standard conic form (P) and consider the following pair of primal and dual problems

$$\begin{aligned} \text{Primal: } & \min c^T x \\ & \text{s.t. } Ax = b, x \in \mathcal{K} \\ \text{Dual: } & \max b^T y \\ & \text{s.t. } A^T y + z = c, z \in \mathcal{K}^*, y \in \mathbb{R}^m \end{aligned} \tag{PD}$$

where $c, x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $\mathcal{K} \subset \mathbb{R}^n$ is a *proper* cone. Let $m \leq n$ and $\text{rank}(A) = m$.

The algorithm

- If there exist $x \in \text{ri}(\mathcal{K})$ such that $Ax = b$ and $z \in \text{ri}(\mathcal{K}^*)$ such that $A^T y + z = c$, then *strong duality* holds for (PD'). In this case, the following KKT system is necessary and sufficient for optimality of (x, y, z)

$$\begin{aligned}Ax - b &= 0 \\A^T y + z - c &= 0 \\x^T z &= 0 \\x \in \mathcal{K}, z \in \mathcal{K}^*, y &\in \mathbb{R}^m.\end{aligned}\tag{8}$$

The algorithm

- We introduce the (full) *homogeneous self-dual embedding model* of (PD) (Ye, Todd, & Mizuno, 1994; Toh, Todd & Tütüncü, 2006),

$$\min \bar{\alpha}\theta$$

$$\text{s.t.} \quad \begin{bmatrix} 0 & -A & b & -\bar{b} \\ A^T & 0 & -c & \bar{c} \\ -b^T & c^T & 0 & -\bar{g} \\ \bar{b}^T & -\bar{c}^T & \bar{g} & 0 \end{bmatrix} \begin{bmatrix} y \\ x \\ \tau \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ z \\ \kappa \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \bar{\alpha} \end{bmatrix} \quad (\text{HSD})$$

$$x \in \mathcal{K}, \quad z \in \mathcal{K}^*, \quad \tau \geq 0, \quad \kappa \geq 0, \quad y \in \mathbb{R}^m, \quad \theta \in \mathbb{R},$$

where given $(x^0, y^0, z^0, \tau^0, \kappa^0, \theta^0)$ such that $x^0 \in \text{ri}(\mathcal{K})$, $z^0 \in \text{ri}(\mathcal{K}^*)$, $\tau^0, \kappa^0, \theta^0 > 0$, set

$$\begin{aligned} \bar{b} &= \frac{1}{\theta^0} (b\tau^0 - Ax^0), & \bar{c} &= \frac{1}{\theta^0} (c\tau^0 - A^T y^0 - z^0), \\ \bar{g} &= \frac{1}{\theta^0} (c^T x^0 - b^T y^0 + \kappa^0), & \bar{\alpha} &= \frac{1}{\theta^0} ((x^0)^T z^0 + \tau^0 \kappa^0). \end{aligned}$$

The algorithm

- ▶ The properties of (HSD) are summarized as follows. For any given $(x^0, y^0, z^0, \tau^0, \kappa^0, \theta^0)$ such that $x^0 \in \text{ri}(\mathcal{K}_{\text{exp}})$, $z^0 \in \text{ri}(\mathcal{P}_{\text{exp}})$ and $\tau^0, \kappa^0, \theta^0 > 0$, the auxiliary parameters \bar{b} , \bar{c} , \bar{g} , $\bar{\alpha} > 0$ and hence (HSD) are well-defined.
 1. The problem (HSD) is self-dual.
 2. $(x, y, z, \tau, \kappa, \theta) = (x^0, y^0, z^0, \tau^0, \kappa^0, \theta^0)$ is a strictly feasible (primal and dual) solution.
 3. The optimal objective is always 0.
 4. Assume $(x, y, \tau, z, \kappa, \theta)$ is feasible. Then $\theta \geq 0$ and $x^T z + \tau \kappa = \bar{\alpha} \theta$. Furthermore, the solution is optimal if and only if $\theta = 0$, in which case $x^T z = \tau \kappa = 0$.
 5. Assume $(x, y, \tau, z, \kappa, 0)$ is an optimal solution. If $\tau > 0$ then $(x, y, z)/\tau$ is an optimal solution to (PD). If $\kappa > 0$ then either $b^T y > 0$ or $c^T x < 0$ or both hold.
 - ▶ If $b^T y > 0$ then (PD) is primal-infeasible.
 - ▶ If $c^T x < 0$ then (PD) is dual infeasible.
 6. For any $\epsilon \geq 0$, there exists a feasible solution of (HSD) with objective value equal to ϵ (Freund, 2005).

The algorithm

- ▶ The goal is to find an optimal solution to (HSD).
- ▶ Define a *central path* $\mathcal{C} = \{\bar{x}_\mu \mid \mu \in (0, 1]\}$ that connects the initial iterate $\bar{x}^0 = (x^0, y^0, z^0, \tau^0, \kappa^0, \theta^0)$ to an optimal solution of (HSD).
 - ▶ An explicit system of equations can be established when \mathcal{K} has a *logarithmically homogeneous self-concordant barrier*.
- ▶ The algorithm approximately traces the central path towards the direction of decreasing μ .
 - ▶ Based on the current iterate which is (usually) near the central path, compute the search direction by linearizing the system of equations governing the central path.
 - ▶ The search direction is a linear combination of the predictor direction (roughly tangent to the central path) and the corrector direction (roughly normal toward the central path).

The algorithm

- ▶ The termination conditions are (roughly) as follows. Consider a given relative accuracy ϵ .
 - ▶ Declare optimality and return the solution $(x, y, z)/\tau$ if

$$\|Ax - \tau b\|_{\infty} \leq \epsilon \cdot \max \{1, \|[A, b]\|_{\infty}\}, \quad (9)$$

$$\|A^T y + z - c\tau\|_{\infty} \leq \epsilon \cdot \max \{1, \|A^T, I, -c\|_{\infty}\}, \quad (10)$$

$$|c^T x/\tau - b^T y/\tau| \leq \epsilon \cdot (1 + |b^T y/\tau|). \quad (11)$$

- ▶ Declare primal and/or dual infeasibility if (9), (10) and

$$|-c^T x + b^T y - \kappa| \leq \epsilon \cdot \max \{1, \|[-c^T, b^T, 1]\|_{\infty}\}, \quad (12)$$

$$\tau \leq \epsilon \cdot 10^{-2} \cdot \{1, \kappa\}. \quad (13)$$

If $b^T y > 0$ ($c^T x < 0$), declare primal (dual) infeasibility.

- ▶ Declare that the problem is ill-posed if

$$\kappa \leq \epsilon \cdot 10^{-2} \cdot \min \{1, \tau\}, \quad \mu \leq \epsilon \cdot 10^{-2} \cdot \mu^0.$$

Plans for the next step

- ▶ On the algorithm and implementation.
 - ▶ Incorporate warm start and iterative refinement strategies.
 - ▶ Integrate the codes into SDPT3.
 - ▶ Build a package that can be called by Python, Julia and so on.
 - ▶ Try fundamentally different methods (PPA, ALM and so on) that might be less accurate but potentially more *scalable*.
- ▶ On application.
 - ▶ Demonstrate the advantages of the robust choice model and other models that quantify distributional uncertainties.

Thank you for your attention!

- ▶ Questions or comments?