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

Yuan Gao[1]

Supervised by

Prof. Kim-Chuan Toh[1]
Prof. Melvyn Sim[2]

1 Department of Mathematics, National University of Singapore
2 Department of Decision Sciences, National University of Singappore

Honour's Project Introductory Talk
Oct 2016

Contents

$*$ It 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, SeDuMi.

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 caling 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.
  - $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}$.
  - $\mathcal{Q}^p = \left\{(x_1, \cdots, x_p)^T \in \mathbb{R}^p \mid x_1 \geq \sqrt{x_2^2 + \cdots + x_p^2}\right\}$.
  - $\mathcal{K}_{\exp}^0 = \left\{(x, y, z)^T \in \mathbb{R}^3 \mid z > 0, \exp\left(\frac{x}{z}\right) \leq \frac{y}{z}\right\}$.
  - $\mathcal{K}_{\exp} = \text{closure}\left(\mathcal{K}_{\exp}^0\right)$.
- It can be shown that

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

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

Current implementation

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

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

  where $K_i \subset \mathbb{R}^{n_i}$ is one of the following.
  - nonnegative orthant $\mathbb{R}_+^{n_i}$.
  - product of second-order cones $\mathcal{Q}^{q_1} \times \cdots \times \mathcal{Q}^{q_{k_i}}$, $\sum_{j=1}^{k_i} q_j = n_i$.
  - product of the exponential cone $(\mathcal{K}_{\exp})^{k_i}$, $3k_i = n_i$.
  - 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*

$$\begin{aligned} \min_x \ & \sum_{i=1}^N c^T x \\ \text{s.t.} \ & \sum_{i=1}^N Ax = b, \ x \in \mathcal{K} \end{aligned} \tag{P'}$$

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

Example: Asset allocation via satisficing measure

- Brown, Giorgi and Sim (2009) consider the following entropic prospective satisficing measure (EPSM) of a random variable

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

Example: Asset allocation via satisficing measure

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

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

  - This example is for illustration purpose.

Example: Asset allocation via satisficing measure

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

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

with decision variables

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

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 $\mathbb{P}\left(F\left(\boldsymbol{x}, \tilde{\boldsymbol{\xi}}\right) \leq 0\right) \geq 1 - \alpha$, where $\boldsymbol{x}$ is the decision vector, $\tilde{\boldsymbol{\xi}}$ 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

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

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

Example: Convex approximation of chance constrained problems

- We characterize $\tilde{r}_j,\ j = 0, \cdots, n$ as follows.
  1. The returns satisfy $\tilde{r}_0 = r_0 = 1$, $\mathbb{E}(\tilde{r}_j) = 1 + \rho_j,\ j = 1, \cdots, n$.
  2. For $j = 1, \cdots, n$, $l = 1, \cdots, q$, 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

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

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

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

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 $q = n + d$, $\bar{\boldsymbol{x}} = (x_0, x_1, \cdots, x_n)$,

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

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

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

- For each j, $k = 1, \cdots, N_j$, let $\tilde{\xi}_j \in \left\{ v_k^j \mid k = 1, \cdots, N_j \right\}$ and $\mathbb{P}\left(\tilde{\xi}_j = v_k^j\right) = p_k^j$. Let $M_j : z \to \sum_{k=1}^{N_j} p_k^j \exp\left(v_k^j z\right)$ 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

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

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

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

with decision variables

$$
\begin{aligned}
&\tau \in \mathbb{R} \\
&x_0, x_1, \cdots, x_n, s_x \geq 0 \\
&g_0, g_1, \cdots, g_d \in \mathbb{R} \\
&t_0 \geq 0 \\
&s_1, \cdots, s_d \in \mathbb{R} \\
&\left(w_k^j, u_k^j, t_k^j\right) \in \mathcal{K}_{\exp},\ j = 1, \cdots, d,\ \ k = 1, \cdots, N_j.
\end{aligned}
$$

Example: Convex approximation of chance constrained problems

- Define $\mathcal{O}^{nominal} = \max\{\rho_i \mid i = 1, \cdots, n\} = \rho_n$.
  - 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 $A \in \mathbb{R}^{2187 \times 3491}$, $density(A) = 0.0012058$, $N_l = 539$, $N_q = 0$, $N_e = 2952$.
- The dimension of (the exponential 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 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.
  - 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

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

where $\boldsymbol{x}_j$ 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 $\boldsymbol{X} = \{\boldsymbol{x}_{nj}\} \subset \mathbb{R}^Q$ and the customer's choice $\boldsymbol{Y} = (y_{nj}) \subset \{0,1\}^{N \times J}$ are recorded.
- We want to find $\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\beta}}$, maximum-likelihood estimates of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$.

Example: Robust choice model

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

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

where

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

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

  - 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

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

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

- 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

- Coming soon...

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 are less accurate but 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?