REPORT

ON

CONVEX OPTIMIZATION – THEORY, APPLICATIONS AND MODELLING IN MATLAB with CVX

by

M. RAMCHANDRAN
(2013B4A7541G)

M.Sc. (Hons) Mathematics + B.E. (Hons) Computer Science

Prepared in partial fulfilment of

the Study Oriented Project Course

(**MATH F266**)

BIRLA INSTITUTE Of TECHNOLOGY AND SCIENCE , PILANI
K.K.BIRLA GOA CAMPUS
(December 2015)

Acknowledgement

I would like to thank Dr. Anil Kumar for giving me the opportunity to pursue the fascinating topic of Convex Optimization. He has helped me understand, question and think more about many concepts and has enriched my experience in this project greatly. I would also like to thank the Mathematics Department at BITS Pilani K.K. Birla Goa Campus for providing the necessary structure and format that encouraged me to undertake this project.

Contents

Acknowledgement i
Introduction 1
1. **Convex Sets** 2
2. **Convex Functions** 7
3. **Convex Optimization Problems** 14
4. **Modelling Problems in CVX** 16

Bibliography 33

Introduction

Optimization Problems arise in almost every field ranging from Computer Science to Biology to Mechanical Engineering. Convex Optimization is heavily used in problems that involve machine learning , portfolio optimization, device sizing in electrical circuits , transport planning , optimal design of mechanical structures etc., The general optimization problem is most often very difficult to solve and methods known involve some level of compromise e.g., very long computation time or not deterministically finding a solution. But there are exceptions - some problem classes can be solved efficiently and reliably. For e.g., least-squares problem, linear programming problems, convex optimization problems.

Even if a problem class can be solved efficiently it doesn't mean that we know analytical solutions to the problem, most often we have sufficiently fast algorithms which will exactly solve the problem. The least-squares problem and linear programming problem are in fact special cases of convex optimization. Convex Optimization problems are often difficult to recognize but can be reliably solved.

The goal of this project was to recognize and formulate problems as convex optimization problems , develop code for these problems using the dedicated Matlab package `CVX` and to analyse the optimal solution.

This report contains four units. The first unit talks about some common Convex Sets and builds a library of such sets. We discuss operations that will preserve convexity and some new concepts like Generalized Inequalities, Minimum and Minimal points and finally Dual Cones.

The second unit talks about common Convex Functions and again builds a library of such functions and operations that preserve the convexity of a function. We might not always be lucky to deal with convex functions and hence we will also look at two other types of functions.

The third unit talks about Convex Optimization Problems and introduces Linear Programming (LP), Quadratic Programming (QP), Quadratically Constrained Quadratic Programming (QCQCP), Second Order Cone Program (SOCP) and finally Semi-Definite Programming (SDP).

The fourth unit introduces the `CVX` package in MATLAB and discusses some computational examples taken from Stephen Boyd's book - *Convex Optimization*. Each problem involved modelling the question given in terms of an optimization problem appropriately and then using the `CVX` to solve it and report back the results

1. Convex Sets

1.1. Definitions and Examples.

Linear Combination

For a collection of vectors x_i in a vector space V, a vector of the form

$$x = \sum_{i \in J} \theta_i x_i$$

is called a linear combination of x_i

Line

For any two points $x_1, x_2 \in \mathbb{R}^n$ the line through these two points is a set of points of the form

$$L = \{\theta x_1 + (1-\theta)x_2, \quad \theta \in \mathbb{R}\}$$

Affine Set

A set that contains the line through any two distinct points in the set is called an affine set.

Convex Combination

A convex combination of a set of points x_i is any point x of the form

$$x = \sum_{i \in J} \theta_i x_i$$

where $\theta_i \in [0,1]$ and $\sum_{i \in J} \theta_i = 1$

Convex Set

A set that contains the line segment between any two points in the set is called a convex set.

$$x_1, x_2 \in A \Rightarrow \theta x_1 + (1-\theta)x_2 \in A$$

where, $\theta_i \in [0,1]$

Convex Hull

A convex hull of a set S is the set of all convex combinations of points of S.

Conic Combination

A conic combination of a set of points x_i is a point of the form

$$x = \sum_{i \in J} \theta_i x_i$$

where, $\theta_i \geq 0 \quad \forall i \in J$

Convex Cones

A set that contains all conic combinations of points of a set S.

Hyperplanes

Given a non-zero direction vector a and a scalar b, A set of the form $\{x \mid a^T x = b\}$ is called a Hyperplane and denoted as $H(a,b)$. Hyperplanes are both convex and affine.

Halfspaces

Given a non-zero direction vector a and a scalar b, A set of the form $\{x \mid a^T x \leq b\}$ is called a Halfspace and denoted as $\tilde{H}(a,b)$. Halfspaces are convex.

Norm-ball

Given a point x_c and radius r and any norm $||.||$, the Euclidean Ball centered at x_c is -

$$B(x_c, r) = \{x \ \ | \ \ ||x - x_c|| \leq r\}$$

Ellipsoid

An Ellipsoid centered at x_c is -

$$E(x_c) = \{x \ \ | \ \ (x - x_c)^T P^{-1}(x - x_c) \leq 1 \, P \in S^n_{++}\}$$

Norm-cone

Associated with the norm, it is the set $\{(x, t) \ \ | \ \ ||x|| \leq t\}$

Second Order Cone

The norm cone for the Euclidean Norm.

1.2. Operations that preserve Convexity of Sets.

Affine Maps

An Affine map is defined as,

$$f : \mathbb{R}^n \to \mathbb{R}^m$$
$$f(x) = Ax + b \qquad x \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$$

Properties

(a) Image of a convex set is convex
(b) Inverse Image of a convex set is Convex

Intersection

Intersection of finitely many Convex Sets are Convex.

Perspective Maps

A perspective map is defined as -

$$P : \mathbb{R}^{n+1} \to \mathbb{R}^n$$

$$P(x, t) = x/t, \quad x \in \mathbb{R}^n, \quad t > 0$$

The perspective map models the action of a pinhole camera. Consider an opaque horizontal plane representing a camera ($x_3 = 0$) with a pinhole at the origin and the image plane ($x_3 = -1$). If the object is placed at x above the camera, then the image through the pinhole is formed at the point $(-\frac{x_1}{x_3}, -\frac{x_2}{x_3}, -1)$. The last component is always -1 and hence can be dropped. The result of the perspective map gives the (x, y) co-ordinates of the image.

Properties

(a) Line Segments of a convex subset are mapped to line segments
This can be verified from definition.

(b) If $C \subset \text{dom}\{P\}$ is convex then its image $P(C)$ is convex.

For convex subset C and any two points x,y the line segment between x and y are mapped to a line segments between x' and y' and thus any x',y and the line segment between them are contained in $P(C)$. Hence $P(C)$ is convex

Examples

(a) When $C = \text{conv}\{(v_1, t_1), (v_2, t_2), \dots, (v_n, t_n)\}, v_i \in \mathbb{R}^n, t_i > 0$,
Then
$$P(C) = \text{conv}\left(\frac{v_1}{t_1}, \frac{v_2}{t_2}, \cdots, \frac{v_n}{t_n}\right), v_i \in \mathbb{R}^n, t_i > 0$$

(b) When $C = \{(v, t) \mid f^T v + gt = h\}$ (Hyperplane), then
$$P(C) = \{z \mid f^T z + g = \frac{h}{t}\}, t > 0$$

1.3. Generalized Inequalities. The real line has an ordering of real numbers and hence we can compare any two real numbers and conclude one of them to be bigger or smaller or that both are equal.

Such a way of comparing vectors is not so straightforward. All but one component of a vector a may be bigger than a vector b, or $|a|$ could be bigger than $|b|$. This problem arises because $\mathbb{R}^n$ doesn't have a linear ordering.

Thus we have to extend our definitions to help us compare vectors. Traditionally while optimizing a real valued function, the definitions of a supremum or maximum is quite clear. But our new definitions will help us analyse the situations where we will optimize vector valued functions in $\mathbb{R}^n$.

We will define a proper cone and then a 'generalized' inequality associated with proper cones and using these analyse the conventional notions of minimum value.

Proper cones

A convex cone $K \subseteq \mathbb{R}^n$ is a proper cone if -

(a) K is closed (i.e. contains boundary)
(b) K is solid (i.e. contains interior)
(c) K is pointed (i.e. contains no line, only rays)

Examples

(a) The Non-negative Orthant
$$\mathbb{R}^n_+ = \{x \in \mathbb{R}^n | \ \ x_i \geq 0 \ \ \forall \ \ 0 \leq i \leq n\}$$

(b) Non-negative Polynomials on $[0, 1]$,
$$K = \{x \in \mathbb{R}^n | \ \ \sum_{i=1}^{n} x_i t^{i-1} \geq 0, \quad t \in [0, 1]\}$$

(c) Positive Semidefinite Cone - $K = S^n_+$

Generalized Inequality

For two vectors x, y and a proper cone K, we say that x is 'lesser' than y with resect to cone K if $(y - x)$ is in K and denote is as $x \preceq_K y$
$$x \preceq_K y \quad \Leftrightarrow \quad y - x \in K$$

Examples

(a) Component-wise Inequality ($K = \mathbb{R}^n_+$)

$$x \preceq_{\mathbb{R}^n_+} y \quad \Leftrightarrow \quad y - x \in \mathbb{R}^n_+ \quad \Leftrightarrow \quad y_i - x_i \geq 0$$

(b) Matrix Inequality ($K = S^n_+$) $X \preceq_{S^n_+} Y \Leftrightarrow Y - X$ is positive semi-definite

Properties

(a) If $x \preceq_K y$ and $u \preceq_K v$ then, $x + u \preceq_K y + v$

(b) The relation $x \preceq_K y$ is transitive, reflexive, antisymmetric and preserved under non-negative scaling.

(c) The relation $\preceq_K$ is not a linear Ordering. It is possible that - $x \npreceq_K y$ and also $y \npreceq_K x$. If for a given x,y, one of the two relations are true then x and y are said to be comparable.

Minimum and Minimal

$x \in S$ is called the **minimum** element of S wrt $\preceq_K$ if $y \in S \Rightarrow x \preceq_K y$

$x \in S$ is called the **minimal** element of S wrt $\preceq_K$ if $y \in S, y \preceq_K x \Rightarrow y = x$

There is a subtle difference between the two definitions and both definitions converge into one in the case of $\mathbb{R}$. Informally speaking, for any point x, the upper-right quadrant and lower-left quadrant (or their equivalent in higher dimensions) are the set of points that are comparable to x. The upper-right quadrant contains points that are comparable to x and bigger than x. We will denote this by $x + K$. The lower-left quadrant contains points that are comparable to x and lesser than it. We will denote this by $x - K$. Therefore we have an alternate way of looking at the definitions.

x is minimum if $S \subseteq x + K$.

x is minimal if $(x - k) \cap S = \{x\}$

A point can be both minimal and minimum or neither. If a point is minimum then it is also minimal.

1.4. Hyperplane Theorems. Hyperplanes are the extension of 2-D planes to n dimensions and statements about tangents to a curve can be extended to n-dimensions and they provide some useful characterizations of convexity.

Separating Hyperplanes

If C and D are disjoint convex sets then $\exists a \neq 0, b$ such that,

$$a^T x \leq b \quad \forall \quad x \in C$$

$$a^T x \geq b \quad \forall \quad x \in D$$

Then the Hyperplane $\{x \mid a^T x = b\}$ separates C and D.

Supporting Hyperplanes

A supporting hyperplane to a set C at a boundary point x_0 is a hyperplane $\{x \mid a^T x = a^T x_0\}$ where $a \neq 0$ and

$$a^T x \leq a^T x_0 \quad \forall \quad x \in C$$

If C is a convex set then there exists a supporting hyperplane at every boundary point of C

1.5. Dual Cones Characterization.

Dual Cones

Given a cone K, the dual cone K^* is defined as,

$$K^* = \{y \mid y^T x \geq 0 \quad \forall \quad x \in K\}$$

K^* is always a convex cone since it is the intersection of homogenous halfspaces.

Geometrically, $y \in K^*$ iff $-y$ is a normal to the hyperplane that supports K at origin. If K is thin,sharp then K^* is thick and blunt.

Examples

(a) Dual cone of a subspace $V \subseteq \mathbb{R}^n$ is the subspace $V^\perp$

(b) $\mathbb{R}^n_+, S^n_+$ are self dual cones.

Properties

(a) If a convex cone K is proper then K^* is also proper.

(b) If K is convex, then $(K^*)^* = K$

Dual Generalized Inequality

When K is proper, since K^* is also proper, we can have generalised inequalities wrt to the dual cone K^*.

$$x \preceq_K y \quad \Leftrightarrow \quad \lambda^T x \leq \lambda^t y \;\; \forall \;\; \lambda \succeq_{K^*} 0$$

$$x \prec_K y \quad \Leftrightarrow \quad \lambda^T x < \lambda^t y \;\; \forall \;\; \lambda \succeq_{K^*} 0$$

Minimum and Minimal Characterisation

Minimum

x is Minimum wrt $\preceq_K$ iff $\forall \;\; \lambda \succ_{K^*} 0$, x is the unique minimizer of $\lambda^T z$ over S,

Or

For any $\lambda \succ_{K^*} 0$, the Hyperplane - $\{z \;\; | \;\; \lambda^T(z - x) = 0\}$ is a strict supporting hyperplane to S at x.

(Both of these can be obtained from the definition)

Minimal

If $\lambda \succ_{K^*} 0$ and x minimizes $\lambda^T z$ over $z \in S$, then x is Minimal.

Its interesting to note that the converse is false, A point x can be minimal and also not a minimizer of $\lambda^T z$ over S for all arbitrary λ. This can be visualised when S is not convex. When S is convex, and x is minimal then there exists a $\lambda \succeq_{K^*} 0$ such that x is a unique minimizer of $\lambda^T z$ over S. Also, any minimizer of $\lambda \succeq_{K^*} 0$ is not a minimal point.

2. **Convex Functions**

Convexity of Functions can play a crucial role in deciding how hard it is to minimize or maximise the functions. Convex Functions have unique global minimums and hence a lot can be said by just characterizing a function as convex.

We shall define them, illustrate the most common examples and use these as building blocks to build more complex convex functions.

We shall state some of the useful operations that preserve convexity of functions and some common characteristics.

Later we shall re-examine our definitions with the added tools of generalized inequalities of the previous chapter.

We might not always have the comfort of convexity and thus to deal with other situations we shall relax our convexity constraint and see what other weaker properties we could use to characterize functions and how they play a role. Each of these weaker definitions will preserve some subset of properties we expect of convex functions.

2.1. Definitions and Examples. A function $f : S \to \mathbb{R}$ is said to be convex, if the following holds -

$$x, y \in S \Rightarrow f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

Examples on $\mathbb{R}$

(1) Affine Functions : $f(x) = ax + b$

(2) Exponential : $f(x) = e^{ax}$

(3) Powers : $f(x) = x^{\alpha}$ on $\mathbb{R}_{++}$ for $\alpha \geq 1$ or $\alpha \leq 0$

(4) Powers of Absolute Vale : $f(x) = |x|^p$ on $\mathbb{R}$ for $p \geq 1$

(5) Negative Entropy : $f(x) = x \cdot \log(x)$ on $\mathbb{R}_{++}$

Examples on $\mathbb{R}^n, \mathbb{R}^{m \times n}$

(1) Affine functions on $\mathbb{R}^n$: $f(x) = a^T x + b$

(2) Norm Functions on $\mathbb{R}^n$: $f(x) = ||x||_p = (\sum |x_i|^p)^{\frac{1}{p}}, p \geq 1$

(3) Affine functions on $\mathbb{R}^{m \times n}$: $f(X) = tr(A^T X) + b = \sum\sum A_{ij} X_{ij} + b$

(4) Spectral Norm on $\mathbb{R}^{m \times n}$: $f(X) = ||X||_2 = \sigma_{max}(X) = (\lambda_{max}(X^T X))^{\frac{1}{2}}$

2.2. Properties and Characterisations.

Restriction to a line

Let $f : \mathbb{R}^n \to \mathbb{R}$, Define a function $g : \mathbb{R} \to \mathbb{R}$ with domain of g as the set $\{t \mid x + tv \in \text{dom}\{f\}\}$ for any $x \in \text{dom}\{f\}, v \in \mathbb{R}^n$ -

$$g(t) = f(x + tv)$$

Then f is convex iff g is convex.

We can use this to show that functions like $f(x) = log(det(X))$ are concave.

First Order and Second Order Conditions

If a function f is differentiable with an open domain, then gradient is defined as

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \cdots, \frac{\partial f(x)}{\partial x_n} \right)$$

If a function f is twice differentiable with an open domain, then the Hessian is defined as

$$\nabla^2 f(x)_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \cdot \partial x_j}$$

First Order Condition A differentiable f with convex domain is convex iff,

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

Second Order Condition A twice differentiable function with open domain is convex iff

$$\nabla^2 f(x) \succeq 0$$

Examples The following functions can be shown to be convex by using the first or second order conditions.

(a) Quadratic Function : $f(x) = \frac{1}{2}x^T Px + q^T x + r \quad (P \in S^n)$

(b) Least Squares Objective : $f(x) = \frac{1}{2}||Ax - b||_2^2$

(c) Log-sum-exp : $f(x) = \log\left(\sum e^{x_k}\right)$

(d) Quaratic-over-Linear : $f(x, y) = \frac{x^2}{y}$

Epigraphs and Sub-level Sets

The α-sublevel set of f is the set

$$C_\alpha = \{x \in \mathrm{dom}\{f\} \quad | \quad f(x) \leq \alpha\}$$

The epigraph of f is the set -

$$Epi(f) = \{(x, t) \in \mathbb{R}^{n+1} \quad | \quad x \in \mathrm{dom}\{f\}, f(x) \leq t\}$$

(a) Sublevel sets of convex functions are convex but converse is false.
(b) f is convex iff its epi-graph is a convex set.

Extended Value Extensions The function $\tilde{f}$ is called the extended value extension of f and is defined as,

$$\tilde{f}(x) = f(x), \quad x \in \mathrm{dom}\{f\}$$
$$\tilde{f}(x) = \infty, \quad x \notin \mathrm{dom}\{f\}$$

When f is convex, so is $\tilde{f}$

Jensen's Inequality If f is convex then

$$f(Ez) \leq Ef(z)$$

where, E is the expectation of the random variable Z.

2.3. Operations that preserve Convexity of Sets. To establish convexity of a function, some practical methods are -

(1) Verify Definition (often by restricting to a line)
(2) For twice-differentiable functions, show that $\nabla^2 f(x) \succeq 0$ (can be very tedious)
(3) Show that f can be obtained from simple convex functions by operations that preserve convexity.

Now we shall see some useful operations that preserve convexity of functions

Non-negative Weighted Sums

(a) $\alpha f(x)$ is convex if $f(x)$ is convex.
(b) $f_1 + f_2$ is convex if f_1, f_2 are convex.
(c) $\sum f_i \alpha_i$ is convex if f_i are convex

Composition with Affine

$f(Ax + b)$ is convex if f is convex.

Examples

(a) Log-barrier for Linear Equalities :

$$f(x) = -\sum log(b_i - a_i^T x)$$

(b) Norm of Affine Function :

$$f(x) = ||Ax + b||$$

Pointwise Maximum

If $f_1, f_2, \cdots, f_m$ are convex, Then their pointwise-maximum defined below is convex.

$$f(x) = \max\{f_1, f_2, \cdots, f_m\}$$

Examples

(a) Piecewise Linear Function :

$$f(x) = \max(a_i^T x + b_i)$$

(b) Sum of r largest components of $x \in \mathbb{R}^n$

$$f(x) = x_{[1]} + x_{[2]} + \ldots + x_{[n]}$$

where, $x_{[i]}$ is the i'th largest component of x.

Pointwise Supremum

If $f(x, y)$ is convex in x for each $y \in A$, then their pointwise supremum as defined below is convex.

$$g(x) = \sup_{y \in A} \quad f(x, y)$$

Examples

(a) Distance to the farthest point in set C :

$$f(x) = \sup_{y \in C} \quad ||x - y||$$

(b) Support Function of a set C :

$$S_c(x) = \sup_{y \in C} \quad y^T x$$

Notes

If C is a convex set then a support function is a way of representing C as an intersection of a collection of half-spaces. Assume that $C \subseteq \mathbb{R}^n$. Consider some point x and $S_C(x)$, then

$$y^T x \leq S_C(x) \;\; \forall \;\; y \in C$$

Observe that this is a description of a half-space $\tilde{H}(x, S_C(x))$, therefore, $C \subseteq \tilde{H}(x, S_C(x))$ for all x and hence,

$$C = \bigcap \tilde{H}(x, S_c(x)) \;\; \forall x$$

Properties

(i) $S_B = S_{conv(B)}$

(ii) $S_{A+B} = S_A + S_B$

(iii) $S_{A \cup B}(x) = \max\{S_A(x), S_B(x)\}$

(iv) If B is a closed convex set, then $A \subseteq B$ iff $S_A(x) \leq S_B(x) \;\; \forall \;\; x$

(v) If C and D are closed convex sets, then they are equal iff their support functions are equal.

Composition with Scalar function

Let $g : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$ and define $f(x) = h(g(x))$. Then f is convex if -

(a) g convex, h convex and $\tilde{h}$ is nondecreasing.

(b) g concave h convex and $\tilde{h}$ is nonincreasing.

Examples

(a) $e^{g(x)}$ is convex if $g(x)$ is convex

(b) $\frac{1}{g(x)}$ is convex if $g(x)$ is concave and positive

Vector Composition

Let $g : \mathbb{R}^n \to \mathbb{R}^k$ and $h : \mathbb{R}^k \to \mathbb{R}$ and define $f(x) = h(g(x)) = h(g_1(x), g_2(x), \ldots, g_k(x)$. Then f is convex if -

(a) g_i convex, h convex and $\tilde{h}$ is nondecreasing.

(b) g_i concave h convex and $\tilde{h}$ is nonincreasing.

Examples

(a) $\log \sum_{i=1}^{m} e^{g(x)}$ is convex if g_i are convex

(b) $\sum_{i=1}^{m} \log(g(x))$ is convex if $g(x)$ is concave and positive

Minimization

If $f(x, y)$ is convex in (x, y) and C is a convex set then, $g(x)$ as defined is convex.

$$g(x) = \inf_{y \in C} f(x,y)$$

Examples

(a) $f(x,y) = x^T Ax + 2x^T By + y^T Cy$ with

$$s = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

minimizing over y gives

$$g(x) = x^T(A - BC^{-1}B^T)x$$

g is convex $\Rightarrow$ Schur Complement $(A - BC^-1B^T) \succeq 0$

(b) Distance to a set - $d(x,s)$ is convex when S is convex.

$$d(x,s) = \inf_{y \in S} ||x - y||$$

Perspective of a Function

Perspective of a function $f : \mathbb{R}^n \to \mathbb{R}$ is a function $g : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ defined as -

$$g(x,t) = t \cdot f\left(\frac{x}{t}\right)$$

with $\mathrm{dom}(g) = \{(x,t) \ \ | \ \ \frac{x}{t} \in \mathrm{dom}(f), \ \ \ t > 0\}$

Examples

(a) $x^T x$ is convex $\Rightarrow g(x,t) = \frac{x^T x}{t}$ is convex for $t > 0$

(b) Negative Logarithm : $f(x) = -log(x)$ is convex, $\Rightarrow g(x,t) = tlog(t) - tlog(x)$ is convex on $\mathbb{R}^2_{++}$

Conjugate of a Function

The conjugate of a function $f : \mathbb{R}^n \to \mathbb{R}$ is a function $f^* : \mathbb{R}^n \to \mathbb{R}$ defined as -

$$f^*(y) = \sup_{x \in \mathrm{dom}(f)} \left(y^T x - f(x)\right)$$

$\mathrm{dom}(f^*) = y \in \mathbb{R}^n$ where, $f^*(y)$ is finite or where $(y^T x - f(x))$ is bounded on domain of f.
f^* is convex since it is a pointwise supremum function that are convex in y (irrespective of f)

Examples

(a) Affine Function : $f(x) = ax + b$
Here $(yx - ax - b)$ is bounded iff $y = a$, Therefore domain of f^* is the singleton set $\{a\}$.
Hence, $f^*(x) = -b$.

(b) Negative Logarithm : $f(x) = -log(x)$
Here $(yx + logx)$ is unbounded above if $y \geq 0$ and maximum at $x = \frac{-1}{y}$ otherwise.
Therefore domain of f^* is $\{y \ \ | \ \ y < 0\} = -\mathbb{R}_{++}$.
Hence, $f^*(y) = -log(-y) - 1$ for $y < 0$

(c) Exponential Function : $f(x) = e^x$
Here $(xy - e^x)$ is unbounded if $y < 0$ and for $y > 0$ the maximum is at $x = log(y)$.
Hence, $f^*(y) = ylog(y) - y$ for $y > 0$ and $f^*(0) = 0$

Properties Frenchel's Inequality states -

$$f(x) + f^*(y) \geq x^T y$$

Notes Consider a situation where we have n resources to produce a product for sale. The resource vector is $r = (r_1, r_2, \ldots, r_n)$ and $S(r) =$ is the sales revenue from product produced (as function of consumed resources). The price of resource is p_i. Thus, total amount paid for resources is $p^T r$ and profit derived is $S(r) - p^T r$

If the prices of the resources are fixed and we can vary resource consumption, then maximum profit is,

$$M(P) = \sup_r \left(S(r) - p^T r\right)$$

which in terms of conjugate functions is, $M(P) = -S^*(-P)$

2.4. Other Functions.

Quasi-convex Functions

A function $f : \mathbb{R}^n \to \mathbb{R}$ is quasi-convex (QC) if it has a convex domain and all its sub-level sets S_α are convex.

$$S_\alpha = \{x \in \mathrm{dom}(f) \;\; | \;\; f(x) \leq \alpha\}$$

Examples

(a) $f(x) = \sqrt{|x|}$ is QC.

(b) $f(x)$ is quasilinear on $\mathbb{R}_{++}$

(c) $f(x, y) = xy$ is quasiconcave on $\mathbb{R}^2_{++}$

(d) Linear Fraction Function is quasilinear ($dom(f) = \{x \;\; | \;\; c^T x + d > 0\}$)

$$f(x) = \frac{a^T x + b}{c^T x + d}$$

(e) Distance Ratio is QC ($\mathrm{dom}(f) = \{x \;\; | \;\; ||x - a||_2 \leq ||x - b||_2\}$)-

$$f(x) = \frac{||x - a||_2}{||x - b||_2}$$

Properties

(a) f is quasiconcave if $-f$ is quasiconvex, and quasilinear if its both quasiconcave and quasiconvex

(b) Quasi-convex functions satisfy the modified Jensen's Inequality ($\forall \;\; \theta \in [0, 1]$) -

$$f\left(\theta x + (1 - \theta)y\right) \leq max\{f(x), f(y)\}$$

(c) First Order Conditions - A differential function with a convex domain is quasi-convex iff

$$f(y) \leq f(x) \Rightarrow \nabla f(x)^T (y - x) \leq 0$$

(d) Sums of Quasiconvex Functions need not be quasiconvex

Log-Concave Functions

A function f is log-concave if $\log(f)$ is concave ($\forall \;\; \theta \in [0, 1]$) -

$$f\left(\theta x + (1 - \theta)y\right) \leq f(x)^\theta f(y)^{1-\theta}$$

Examples

(a) Powers x^α on $\mathbb{R}_{++}$ is log-convex for $\alpha \leq 0$ and log-concave for $\alpha \geq 0$

(b) Many commonly known probability densities are log-concave. Eg. Normal, Cumulative Gaussian

Properties

(a) A twice differentiable function f with convex domain is log-concave iff

$$f(x) \cdot \nabla^2 f(x) \preceq \nabla f(x) \cdot \nabla f(x)^T$$

(b) Product of log-concave functions is log-concave

(c) Sum of log-concave functions need not be log-concave.

2.5. Convexity with Generalized Inequalities.

Monotonicity

Suppose $K \subset \mathbb{R}^n$ is a proper cone with an associated generalized inequality $\preceq_K$.
A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be K-decreasing if

$$x \preceq_K y \Rightarrow f(x) \leq f(y)$$

K-Convexity

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is K-convex if domain of f is convex and -

$$f(\theta x + (1-\theta)y) \preceq_K \theta f(x) + (1-\theta) f(y)$$

Examples

(a) $f : S^m \to S^m$ defined as $f(X) = X^2$ is S_+^m-convex

(b) $f : S^m \to S^m$ defined as $f(X) = X^{-1}$ is S_+^m-convex

Properties

$f : \mathbb{R}^n \to S^m$ is S_+^m-convex is equivalent to saying that $g(X) = z^T f(X) z$ is convex in X for all z

Epigraphs and Sublevel sets

(a) f is K-convex $\Leftrightarrow$ $epi_k(f)$ is convex

(b) f is K-convex $\rightarrow$ $S_\alpha(f)$ is convex

3. Convex Optimization Problems

Standard Convex Problem.

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0 \qquad i = 1, \ldots, m \\ & A^T x = b \end{aligned}$$

where $f_0, f_1, \ldots, f_m$ are all convex and the equality constraints are affine.
There can be different variations on the conditions that lead to identified problems and sub-problems

(1) If all the constraints are affine, then its called a Linear Program - LP

(2) If the objective function f_0 is quasiconvex, then it is a quasiconvex problem - QC

(3) If there are no explicit constraints then it is called an Unconstrained Problem

(4) If there are only equality constraints then it is called an Equality Constrained Problem

(5) If the objective function is constant or there isn't one, then it becomes a Feasibility Problem.

Quadratic Programming.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T P x + q^T x + r \\ \text{subject to} \quad & Gx \preceq h \\ & A^T x = b \end{aligned}$$

Here, $P \in S^n_+$ and thus the objective is convex quadratic. We are minimizing a quadratic function over a polyhedron

Quadratically Constrained Quadratic Programming.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} \quad & \frac{1}{2} x^T P_i x + q_i^T x + r_i \leq 0 \\ & A^T x = b \end{aligned}$$

Here, $P_i \in S^n_+$ and thus the objective and the constraints are convex quadratic. We are minimizing a quadratic function over an intersection of ellipsoids.

Second Order Cone Programming.

$$\begin{aligned} \text{minimize} \quad & f^T x \\ \text{subject to} \quad & ||A_i x + b_i||_2 \leq c_i^T x + d_i \quad 1 \leq i \leq m \\ & Fx = g \end{aligned}$$

Here the inequality constraints are - second-order cone constraints that is,

$(A_i x + b_i, c_i^T x + d_i) \in$ second order cone in $\mathbb{R}^{n_i+1}$

If $n_i = 0$ then it reduces to an LP and if $c_i = 0$ it reduces to a QCQP, thus it is more general.

LP $\subset$ QP $\subset$ QCQP $\subset$ SOCP

Geometric Programming.

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 1 \\ & h_j(x) = 1 \end{aligned}$$

where, f_i are posynomial and h_j are monomials.
It can be transformed to a convex problem.

Generalized Inequality Constraints.

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \preceq_{K_i} 0 \qquad i = 1, \ldots, m \\ & A^T x = b \end{aligned}$$

Here each f_i constraint function is K_i convex (or convex wrt proper cone K_i)

Conic Form Problem.

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Fx + g \preceq_K 0 \qquad i = 1, \ldots, m \\ & A^T x = b \end{aligned}$$

Its a special case of Generalized Inequality Problem with affine objective and a single proper cone K
When K is the non-negative orthant it reduces to LP and thus it extends LP to non-polyhedral cases.

Semi-Definite Programming.

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & F_1 x_1 + \ldots + F_n x_n + G \leq 0 \qquad i = 1, \ldots, m \\ & A^T x = b \end{aligned}$$

Here, $F_i, G \in S^k$ and the inequality constraints are called - Linear Matrix Inequality (LMI)
SOCP is a subcase of SDP and thus formulaion as SDP is powerful.
If all the matrices G, F_i are diagonal then it reduces to an LP.

4. Modelling Problems in CVX

Introduction. CVX is a free matlab package written by Stephen Boyd and Michael Grant with convex optimization modelling in mind. It can be downloaded and installed from - http://cvxr.com/cvx/download/

In its default mode, CVX supports a particular approach to convex optimization that is called disciplined convex programming (DCP). Under this approach, convex functions and sets are built up from a small set of rules from convex analysis, starting from a base library of convex functions and sets.

As seen before in this report, we have mentioned most of base library of knowledge of convex structures.

Constraints and objectives that are expressed using these rules are automatically transformed to a canonical form and solved by calling appropriate solvers.

A sample program with CVX looks like -

```
cvx_begin
   variable x
   minimize    c'*x
   subject to  A'x <= b
cvx_end
```

All CVX specifications within a matlab command must be within a `cvx_begin` and `cvx_end` . When the execution reaches `cvx_end`, the variables defined by the same keyword witin the program is updated with a optimal solution (if found). We can use `cvx_status` to check the status of a problem after running it and `cvx_optval` to obtain the optimum value of the problem specified.

4.1. Hello World in CVX.

Problem.

Use CVX to verify the optimal values for the problem given below for two sample objective functions,

(1) $f_0(x_1, x_2) = x_1 + x_2$
(2) $f_0(x_1, x_2) = x_1^2 + 9x_2^2$

$$\begin{aligned} \text{minimize} \quad & f_0(x_1, x_2) \\ \text{subject to} \quad & \\ & 2x_1 + x_2 \geq 1 \\ & x_1 + 3x_2 \geq 1 \\ & x_1 \geq 0 \\ & x_2 \geq 0. \end{aligned}$$

Solution.

```
cvx_begin
    variable x1;
    variable x2;
    minimize (x1+x2)
    subject to
        2*x1+x2 >= 1
```

```
        x1+3*x2 >=1
        x1>=0
        x2>=0
cvx_end

cvx_status

cvx_optval

%2
cvx_begin
    variable x1;
    variable x2;
    minimize (x1^2+9*x2^2)
    subject to
        2*x1+x2 >= 1
        x1+3*x2 >=1
        x1>=0
        x2>=0
cvx_end

cvx_status
```

4.2. Square LP.

Problem.

To solve the square LP problem -

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & A^T x \le b \end{aligned}$$

where A is square and non-singular

Theoretically, the optimum value is given by -

$$\begin{aligned} y* &= c^T A^{-1} b \quad \text{if} \quad c^T A^{-1} \le 0 \\ &= -\infty \qquad \text{otherwise} \end{aligned}$$

Solution.

```
n = 2;
c1 = [-1 ;-2];
A1 = [1 0; 0 1];
%A2 = [1 9;13 4];
b = [2 ; 3];
```

```
x_theory = c'*inv(A)*b;

fprintf(1,'Computing the optimal solution ...');
cvx_begin
    variable x(n);
    minimize (c'*x)
    A1*x <= b
cvx_end
fprintf(1,'Done! \n');

disp('------------------------------------------------------------------------');
disp('The computed optimal solution is: ');
disp(x);
disp('The given optimal solution is: ');
disp(x_theory);
```

4.3. Heuristic Boolean LP.

Problem.

In a Boolean linear program, the variable x is constrained to have components equal to zero or one:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & x_i \in \{0,1\}, \quad i = 1, \dots, n \end{array}$$

In general, such problems are very difficult to solve, even though the feasible set is finite (containing at most 2^n points). In a general method called relaxation, the constraint that x be zero or one is replaced with the linear inequalities $0 \leq x \leq 1$

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & 0 \leq x_i \leq 1 \quad i = 1, \dots, n \end{array}$$

We refer to this problem as the LP relaxation of the Boolean LP. The LP relaxation is far easier to solve than the original Boolean LP.

(a) Show that the optimal value of the LP relaxation is a lower bound on the optimal value of the Boolean LP . What can we say about the Boolean LP if the LP relaxation is infeasible?

(b) It sometimes happens that the LP relaxation has a solution with $x \in \{0,1\}$. What can we say in this case?

(c) Let p^* be the optimal value of the Boolean LP and let x^{rlx} be a solution of the LP relaxation of the problem, so $L = c^T x^{rlx}$ is a lower bound on p. The relaxed solution x^{rlx} can also be used to guess a Boolean point $\tilde{x}$, by rounding its entries, based on a threshold $t \in [0,1]$:

$$\tilde{x}_i = 1 \quad x_i^{rlx} \geq t$$
$$= 0 \quad \text{otherwise,}$$

for i = 1, . . ., n. Evidently $\tilde{x}$ is Boolean (i.e., has entries in $\{0, 1\}$). If it is feasible for the Boolean LP, then it can be considered a guess at a good, if not optimal, point for the Boolean LP.

Its objective value, $U = c^T\tilde{x}$, is an upper bound on p. If U and L are close, then $\tilde{x}$ is nearly optimal; specifically, $\tilde{x}$ cannot be more than $(U - L)$-suboptimal for the Boolean LP.

This rounding need not work; indeed, it can happen that for all threshold values, $\tilde{x}$ is infeasible. But for some problem instances, it can work well. Of course, there are many variations on this simple scheme for (possibly) constructing a feasible, good point from x^{rlx}

Finally, we will generate problem data randomly.

We can think of x_i as a job we either accept or decline, and c as the (positive) revenue we generate if we accept job i. We can think of $Ax \preceq b$ as a set of limits on m resources. A_{ij}, which is positive, is the amount of resource i consumed if we accept job j ; b_i, which is positive, is the amount of resource i available.

The objective is to find a solution of the relaxed LP and examine its entries. To note the associated lower bound L and carry out threshold rounding for (say) 100 values of t, uniformly spaced over $[0, 1]$. For each value of t, we will note the objective value $c^T\tilde{x}$ and the maximum constraint violation $\max_i(A\tilde{x}b)$

We will plot the objective value and the maximum violation versus t and indicate on the plot the values of t for which $\tilde{x}$ is feasible, and those for which it is not.

We can find a value of t for which $\tilde{x}$ is feasible, and gives minimum objective value, and note the associated upper bound U. Give the gap $U - L$ between the upper bound on p^* and the lower bound on p

Solution.

(a) The feasible set of the relaxation includes the feasible set of the Boolean LP. It follows that the Boolean LP is infeasible when the relaxation is infeasible, and that the optimal value of the relaxation problem is less than or equal to the optimal value of the Boolean LP.

(b) The optimal solution of the relaxation is also optimal for the Boolean LP in this case.

(c) The Matlab code -

```
rng(0,'v5uniform');
n=100;
m=300;
A=rand(m,n);
b=A*ones(n,1)/2;
c=-rand(n,1);

cvx_begin
    variable x(n);
    minimize (c'*x)
    subject to
        A*x <= b
        0 <= x
        x <= 1
```

```
cvx_end

xrl = x
L=cvx_optval

thres=0:0.01:1;
max_violation = zeros(length(thres),1);
objective = zeros(length(thres),1);
for i=1:length(thres)
xbool = (xrl>=thres(i));
max_violation(i) = max(A*xbool-b);
objective(i) = c'*xbool;
end

% find least upper bound and associated threshold
i_feasible=find(max_violation<=0)
U=min(objective(i_feasible))
%U=min(obj(find(maxviol <=0)))
t=min(i_feasible)
min_thresh=thres(t)
U-L

% plotting objective and max violation versus threshold
subplot(2,1,1)
plot(thres(1:t-1),max_violation(1:t-1),'r',thres(t:end), ...
max_violation(t:end),'b','linewidth',2);
xlabel('threshold');
ylabel('max violation');
subplot(2,1,2)
hold on;
plot(thres,L*ones(size(thres)),'k','linewidth',2);
plot(thres(1:t-1),objective(1:t-1),'r',thres(t:end), ...
objective(t:end),'b','linewidth',2);
xlabel('threshold');
ylabel('objective');
```

Figure 1 shows the plots between the threshold and objective value and maximum violations. The lower bound found for the relaxed LP is $L = 33.1672$. The threshold value $t = 0.6006$ gives the best objective value for feasible $\tilde{x} : U = 32.4450$. The difference is 0.7222. So $\tilde{x}$, with $t = 0.6006$, can be no more than 0.7222 suboptimal.

In figure 1, the red lines indicate values for thresh-holding values which give infeasible $\tilde{x}$, and the blue lines correspond to feasible $\tilde{x}$.

We see that the maximum violation decreases as the threshold is increased. This is actually specific to our problem instance since the constraint matrix A only has non-negative entries. At a threshold of 0, all jobs are selected, which is an infeasible solution. As we increase the threshold, jobs are removed in sequence (without adding new jobs), which monotonically decreases the maximum violation.

For a general boolean LP, the corresponding plots need not exhibit monotonic behaviour.

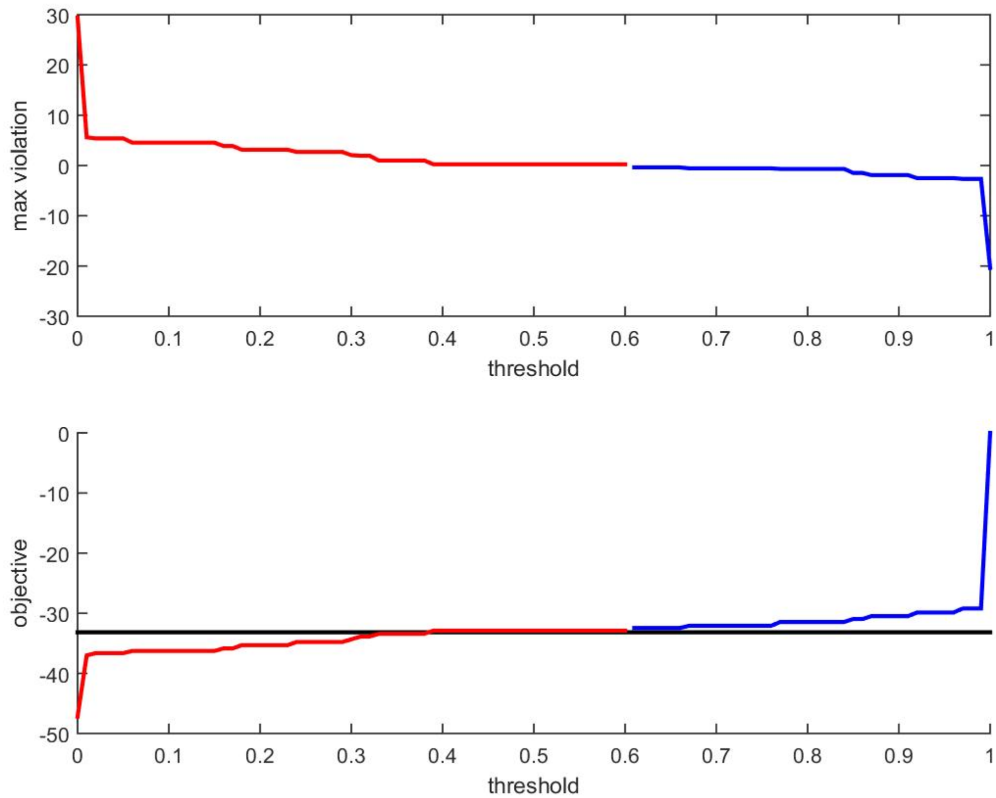

FIGURE 1. HeuresticBooleanLP

4.4. Quasiconvex Optimization.

Problem.

To solve a quasi-convex optimization problem by using bisection method and the idea of convex feasibility. Here ceil(x) represents the ceiling function which evaluates to the least integer greater than x.

$$\begin{aligned} \text{minimize} \quad & \text{ceil}(x) \\ \text{subject to} \quad & 3x > 5 \end{aligned}$$

Solution.

Epigraph of Quasi-convex functions are convex sets and we will use this fact to get a convex representation of a quasi-convex function $f_0(x)$

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0 \\ & Ax = b \end{aligned}$$

is equivalent to the convex feasibility problem -

$$\begin{aligned} \text{Find} \quad & x \\ \text{subject to} \quad & f_i(x) \leq 0 \\ & Ax = b \\ & \phi_t(x) \leq 0 \end{aligned}$$

where $\phi_t(x)$ is a convex representation of the quasiconvex function $f_0(x)$ such that,

$$\phi_t(x) \leq 0 \Leftrightarrow f_0(x) \leq t$$

Thus, if for a given t, the convex feasibility problem has a solution, then we can conclude that our optimal value p^* of the quasiconvex problem is lesser than t.

Thus our algorithm will be,

Bisection Algorithm for QC Problems

Given $l \leq p^* \leq u$, tolerance $\epsilon > 0$

Repeat

– $t := \frac{(l+u)}{2}$

– Solve the convex feasibility problem for t

– if feasible then, $u := t$

– else, $l := t$

– until $|u - l| \leq \epsilon$

This algorithm requires exactly $\log_2(\frac{u-1}{\epsilon})$ iterations. Here we implement this algorithm for a sample function and check the results

```
%   over the interval (a,b) = (0,128) with tolerance - 1
%   answer should be got in log_2 ( (b-a)/e) ) iterations
%   last interval - (1,2)
%   optimal point estimate - 1.5
%   optimal value estimate - 2

a = 0;
b = 128;
n = 70;
e = 1;

for i = 1:n
    t = (a + b)/2;
    cvx_begin
        variable x
        minimize (0)
        subject to
        x <= t
        3*x + 5 > 10
    cvx_end

    if(strcmp(cvx_status,'Solved')==1)
        b = t;
    else
        a = t;
    end
    if((b-a)<=e)
        break;
    end
end

% set the best estimate for x and the error bound
x = (a + b)/2
e = (b-a)/2
```

4.5. Quadratic Programming.

Problem.

Show how to model a quadratic programming problem in CVX.

Solution.

```
%          minimize   1/2x'*P*x + q'*x + r
%          subject to   -1 <= x_i <= 1    for i = 1,2,3
```

```
% Manual input data, can be varied to get different problems.
P = [2 1; -1 2];
q = [-1 ; 1];
r = 0.5;
n = 2;

cvx_begin
    variable x(n)
    minimize ((1/2)*quad_form(x,P) + q'*x + r)
    x >= -1;
    x <=  1;
cvx_end

disp('----------------------------------------------------------------------');
disp('The optimal solution is: ');
disp(x);
```

4.6. Separable Hyperplanes.

PROBLEM.

Suppose you are given two sets of points in R, $\{v_1, v_2, \ldots, v_K\}$ and $\{w_1, w_2, \ldots, w_L\}$. Formulate the following as an LP feasibility problem.

Determine a hyperplane that separates the two sets, i.e., nd $a, b \in \mathbb{R}$ and with $a \neq 0$ such that

$$a^T v_i \leq b \quad i = 1, ..., K$$
$$a^T w_i \geq b \quad i = 1, ..., L$$

SOLUTION.

The conditions given form a set of K + L linear inequalities in the variables a, b, which we can write in matrix form as $Bx \succeq 0$ where

$$B = \begin{pmatrix} -(v_1)^T & 1 \\ ... & ... \\ -(v_K)^T & 1 \\ -(w_1)^T & -1 \\ ... & ... \\ -(w_L)^T & -1 \end{pmatrix} \in \mathbb{R}^{(K+L)\times(n+1)} \quad , \quad x = \begin{pmatrix} a \\ b \end{pmatrix}$$

To ensure that we get a non-trivial solution, we will impose an additional constraint $1^T Bx = \alpha$ where α is any positive real number.

```
% problem of the form Bx = 0 and 1'Bx=1
% sample points, can be varied.

v1 = [1; 0];
v2 = [1; -1];
v3 = [0; 0.5];
```

```
w1 = [4 ; 5];
w2 = [3 ; 9];
w3 = [2.5 ; 2];
n=2;

B = [-v1' 1; -v2' 1; -v3' 1; -w1' -1; -w2' -1; -w3' -1];
one = [1;1;1;1;1;1];

cvx_begin
    variable x(n+1)
    minimize (0)
    B*x >= 0
    one'*(B*x) == 1


cvx_end

B*x
fprintf(1,'Done! \n');

disp('----------------------------------------------------------------------------');
disp('The computed optimal solution is: ');
disp(x);
```

4.7. Optimal Activity Level.

Problem.

We consider the selection of n nonnegative activity levels, denoted $x_1, \dots, x_n$. These activities consume m resources, which are limited. Activity j consumes $A_{ij}x_j$ of resource i, where A_{ij} are given. The total resource consumption is additive, so the total of resource i consumed is $c_i = \sum A_{ij}x_j$ (Ordinarily we have $A_{ij} \geq 0$ i.e, activity j consumes resource i, but we allow the possibility that $A_{ij} < 0$ which means that activity j actually generates resource i as a by-product). Each resource consumption is limited: we must have $c_i \leq c_i^{\max}$, where $c_i^{\max}$ are given. Each activity generates revenue, which is a piecewise-linear concave function of the activity level:

$$
\begin{aligned}
r_j(x_j) &= p_j x_j && 0 \leq x_j \leq q_j \\
&= p_j q_j + p_j^d (x_j - q_j) && x_j \geq q_j
\end{aligned}
$$

Here $p_j > 0$ is the basic price, $q_j > 0$ is the quantity discount level, and p^d is the quantity discount price, for (the product of) activity j. (We have $0 < p_j^d < p_j$) . The total revenue is the sum of the revenues associated with each activity, i.e., $\sum_{j=1}^{n} r_j(x_j)$. The goal is to choose activity levels that maximize the total revenue while respecting the resource limits. Show how to formulate this problem as an LP.

Solution.

We can model the problem as -

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{n} r_j(x_j) \\ \text{subject to} & Ax \preceq c^{\max} \\ & x \succeq 0 \end{array}$$

We have a concave objective function and linear equality constraints and hence this is a convex optimization problem. The issue here is to find a closed form expression for the revenue which has been given case-wise. One way to deal with this is to take a lower bound of the two cases at all times and just maximise this lower bound. This would work here since,

$$r_j(x_j) = \min\{p_j x_j, \quad p_j q_j + p_j^d(x_j - q_j)\}$$

Note that, $r_j(x_j) \geq u_j$ iff

$$p_j x_j \geq u_j \quad \text{and} \quad p_j q_j + p_j^d(x_j - q_j) \geq u_j$$

Thus we can form our original problem in an equivalent LP form as,

$$\begin{array}{ll} \text{maximize} & 1^T u \\ \text{subject to} & Ax \preceq c^{\max} \\ & x \succeq 0 \\ & p_j x_j \geq u_j \\ & p_j q_j + p_j^d(x_j - q_j) \geq u_j \end{array}$$

```
A = [1 2 0 1; 0 0 3 1; 0 3 1 1; 2 1 2 5; 1 0 3 2];
c_max = [100;100;100;100;100];
p = [3;2;7;6];
p_disc = [2;1;4;2];
q = [4;10;5;10];

n =  4;
m = 5;
o = [1;1;1;1]

cvx_begin
   variables u(n) x(n)
   maximize (o'*u)
   x >= 0
   A*x <= c_max
   for j=1:n
   p(j)*x(j) >= u(j)
   p(j)*q(j) + p_disc(j)*(x(j)-q(j)) >= u(j)
   end
   cvx_end
```

```
disp('--------------------------------------------------------------------------');
disp('The computed optimal solution is: ');
disp(x);
disp('The computed optimal revenues is: ');
disp(u);
```

4.8. Chebyshev Center of a Polyhedra.

PROBLEM.

The goal is to find the largest inscribed Euclidean ball (i.e. its center and radius) that lies in a polyhedron described by $P = \{x \mid a_i^T x \leq b_i, i = 1, \ldots, m\}$ where x is in $\mathbb{R}^n$ by formulating it as an LP

SOLUTION.

Although at first look the idea of finding a euclidean ball satisfying a property doesn't seem to involve linear constraints. LP by nature usually involves constraints of half-spaces and hyperplanes which are "flat" surfaces in n dimensions. But here, we are introducing curvature by asking to find a specific euclidean ball.

Nevertheless, careful analysis will show that this is possible.

Suppose $B_2(x_c, r)$ is the largest inscribed euclidean ball,

$$B_2(x_c, r) = \{x_c + u \mid ||u_2|| \leq r\}$$

For $B_2(x_c, r)$ to be contained within the polyhedra P,

$$\begin{aligned} a_i^T x \leq b_i \;\; \forall \;\; x \in B \quad &\iff \quad \sup\{a_i^T(x_c + u) \mid ||u||_2 \leq r\} \leq b_i \\ &\iff \quad a_i^T x_c + r||a_i||_2 \leq b_i \end{aligned}$$

We can now find x_c, r by solving the LP -

$$\begin{aligned} \text{maximize} \quad & r \\ \text{subject to} \quad & a_i^T x_c + r||a_i||_2 \leq b_i \quad \forall i \end{aligned}$$

The question of finding the deepest point inside a 3D object is similar to finding the Chebyshev center of a polyhedra.

```
% Generate the input data
a_1 = [ 2; 1];
a_2 = [ 2; -1];
a_3 = [-1; 2];
a_4 = [-1; -2];
b = ones(4,1)

cvx_begin
    variable r(1)
    variable x_c(2)
```

```
    maximize ( r )
    a_1'*x_c + r*norm(a_1,2) <= b(1);
    a_2'*x_c + r*norm(a_2,2) <= b(2);
    a_3'*x_c + r*norm(a_3,2) <= b(3);
    a_4'*x_c + r*norm(a_4,2) <= b(4);
cvx_end

% Plotting the figure to show the ball generated
x = linspace(-2,2);
theta = 0:pi/100:2*pi;
plot( x, -x*a_1(1)./a_1(2) + b(1)./a_1(2),'b-');
hold on
plot( x, -x*a_2(1)./a_2(2) + b(2)./a_2(2),'b-');
plot( x, -x*a_3(1)./a_3(2) + b(3)./a_3(2),'b-');
plot( x, -x*a_4(1)./a_4(2) + b(4)./a_4(2),'b-');
plot( x_c(1) + r*cos(theta), x_c(2) + r*sin(theta), 'r');
plot(x_c(1),x_c(2),'k+')
xlabel('x_1')
ylabel('x_2')
title('Chebyshev center and corresponding ball for 2D polyhedron');
axis([-1 1 -1 1])
axis equal
```

Figure 2 shows the chebyshev center of a 2D Polyhedron. The blue lines represent the inequality constraints that define our polyhedron and the circle in red represents the inscribed circle of largest radius.

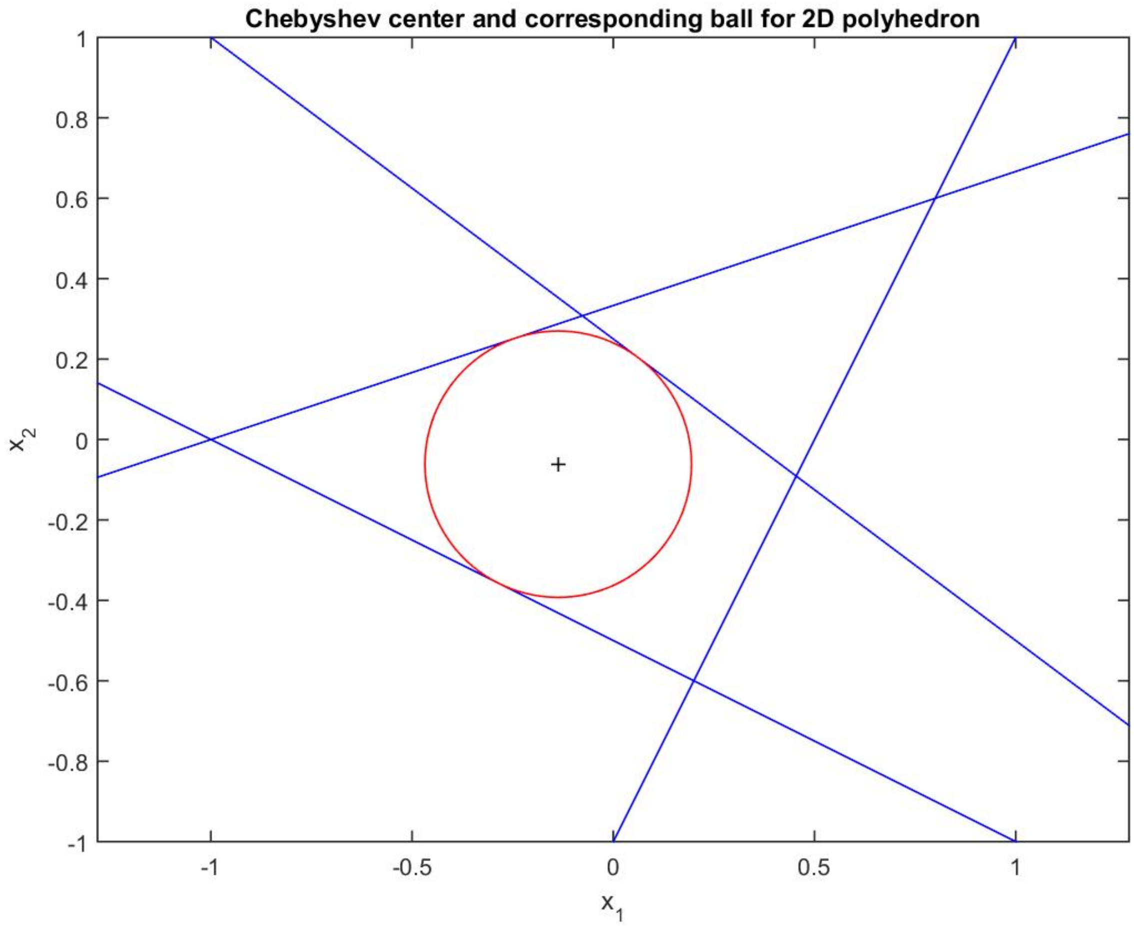

Figure 2. Chebyshev Center

4.9. Geometric Programming.

Problem.

Solve a simple GP problem using CVX

Solution.

With GP, CVX has a special mode keyword - gp - that has to added to ”cvx begin” to tell CVX that its a GP problem and to then check for the constraints to be posynomials or monomials accordingly
Here we solve the problem-

$$\begin{aligned} \text{minimize} \quad & (\max\{p(x), q(x)\}) \\ \text{subject to} \quad & x \geq 2 \\ & y \geq 2 \end{aligned}$$

where $p(x), q(x)$ are posynomials.

```
% here, p(x) = x + y^3
% q(x) = x^2y
% Solution -
%t =  10.000000002202405
%x =   2.000000002559305
%y =   2.000000000213355

cvx_begin gp
    variables t x y
    minimize( t )
    subject to
        (x+y^3)/t <= 1
        (x^2*y)/t <= 1
        x >= 2
        y >= 2
cvx_end

% if p(x) = x^2 + y^2 and q(x) = xy, then p(x) >= q(x) therefore we will
% be minimizing p(x) alone
% thus the solution will be x=2 and y=2 and optval = 8

cvx_begin gp
    variables t x y
    minimize( t )
    subject to
        (x^2+y^2)/t <= 1
        (2*x*y)/t <= 1
        x >= 2
        y >= 2
cvx_end
```

4.10. Semi-definite Programming.

PROBLEM.

Solve a simple SDP problem using CVX

SOLUTION.

With SDP, CVX has a special mode keyword - sdp - that has to added to "cvx begin" to tell CVX that its a SDP problem and to then check for the constraints to be Linear-Matrix-Inequalities (LMI)

```
%SDP simple example
randn('state',0);
n = 4;
A = randn(n);
%to make it symmetric
A = 0.5*(A'+A); %A = A'*A;
B = randn(n); B = B'*B;

c = -1;

cvx_begin sdp
   variable t
   minimize ( c*t )
   A >= t * B;
cvx_end

disp('-----------------------------------------------------------------------');
disp('The optimal t obtained is');
disp(t);
```

Bibliography

[1] Boyd, S., Vandenberghe, L.,*Convex Optimization*, Cambridge University Press 2004, 1st edition, 2004.

[2] Bazaraa, M., Sherali, H.,Shetty, C.,*Nonlinear Programming - Theory and Algorithms*, John Wiley & Sons, second edition, 1993.

[3] CVX : Matlab Software for DCP , `http://cvxr.com/cvx` , March 2014

[4] EE364a : Convex Optimization I , `http://stanford.edu/class/ee364a/`