Convex Optimisation

Yuanda Dong, z5383121

5 December 2021

Abstract

Convex optimisation and mathematical optimisation in general is a vast topic. I'll try to cover the theory of convex optimisation quickly without sacrificing too much rigour, and include also a couple of applications. Much of this project is a summary of Professor Boyd's book on convex optimisation chapter 1-6. In particular, I will not cover numerical algorithms for solving convex optimisation problems. However, it is true that we can solve convex problems up to thousands of variables and constraints, and it is also true that most problems of practical importance are non-convex.

Contents

1 Mathematical optimisation

This should serve as an introduction to mathematical optimisation, and the role of convex optimisation.

Any equality/inequality constrained optimisation problems can be framed into the following standard form. For example, if a problem requires maximisation instead of minimisation, one could minimise the negative objective function $-f_0(x)$; if some inequality constraints are $\geq$ instead of $\leq$, one could replace $f_i(x) \geq 0$ by $-f_i(x) \leq 0$.

Definition 1. Optimisation problem in standard form

$$\begin{aligned} \text{minimize } & f_0(x) \\ \text{subject to } & f_i(x) \leq 0, \quad i = 1, ..., m \\ & h_i(x) = 0, \quad i = 1, ..., p \end{aligned}$$

where

- $x \in \mathcal{R}^n$ is the optimisation variable
- $f_0 : \mathcal{R}^n \rightarrow \mathcal{R}$ is the objective function
- $f_i : \mathcal{R}^n \rightarrow \mathcal{R}$ are the inequality constraint functions
- $h_i : \mathcal{R}^n \rightarrow \mathcal{R}$ are the equality constraint functions

- We denote the optimal value as $p^* = \inf\{f_0(x) \mid f_i(x) \leq 0,\ i = 1, ..., m,\ h_i(x) = 0,\ i = 1, ..., p\}$.
- A point $x \in dom\ f_0$ is feasible if it satisfies the constraint functions. Among these feasible points, x is optimal if $f_0(x) = p^*$ [which may not be unique].
- We define $p^* = \infty$ if the problem is infeasible, $p^* = -\infty$ if the problem is unbounded from below. This also makes intuitive sense.
- Also there is the notion of local optimality. We say x is locally optimal if there exists a ball centred around x such that x is optimal within the ball. Formally, $\exists R > 0$, such that x is optimal for

$$\begin{aligned} \text{minimize (over z) } & f_0(z) \\ \text{subject to } & f_i(z) \leq 0, \quad i = 1, ..., m \\ & h_i(z) = 0, \quad i = 1, ..., p \\ & ||z - x||_2 \leq R \end{aligned}$$

- We will show later for convex problems, any locally optimal points are globally optimal.

Use of mathematical optimisation is ubiquitous. For example, Least-squares problem minimises $||Ax - b||_2^2$, and is used for regression / data fitting. It is well-known that Least-squares has an analytical solution $x^* = (A^T A)^{-1} A^T b$. Linear programs (LP) minimises linear objective function, subject to linear constraints, and it is known that max-flow min-cut is reducible to an LP [In fact the equality of max-flow and min-cut follows from strong duality which we will explore later]. However, linear program does not admit an analytical solution.

Their common parent is convex optimisation in which the objective and the inequality constraints f_i are convex, and equality constraints h_i are affine. It is often the case that, problems may not look like convex, but under some transformations can be made into a convex problem. Once we formulated the problem into a convex program, it can be solved efficiently and reliably. It should be mentioned that convex optimisation plays an important role even in non-convex problems. For example, in Lagrangian relaxation [which we will explore later], the convex dual problem is solved and it provides a lower bound on the optimal value of the primal. Moreover, convex optimisation is used as the basis for several heuristics for solving non-convex problems.

2 Convex sets

In order to study convex optimisation properly, we shall start with basic definitions of convex sets and convex functions. We will also see a Calculus for constructing complex convex functions from basic functions. Many important notions relevant for convex optimisations will also be defined.

2.1 Affine and convex sets

Here is a quick reference to affine/convex sets and some important examples.

- line through $x_1,\ x_2$: $y = \theta x_1 + (1-\theta)x_2 = x_2 + \theta(x_1 - x_2),\ \theta \in R$
- Affine sets: if $x_1,\ x_2$ are in the set $\mathcal{C}$, then the line through $x_1,\ x_2$ lies in the set. Examples include the solution set of a system of linear equation $\{x \mid Ax = b\}$, since every affine combinations [coefficients add up to 1] of a solution is yet another solution. It can be shown that every affine set can be expressed as $\mathcal{C} = V + x_0 = \{v + x_0 \mid v \in V\}$, where V is a vector subspace, i.e. closed under linear combinations and containing the zero vector, offset x_0 can be chosen as any element in $\mathcal{C}$.
- Convex sets: if $x1,\ x2$ are in the set $\mathcal{C}$, then for $0 \leq \theta \leq 1$, the line segment $\theta x_1 + (1-\theta)x_2$ is also in the set. Geometrically speaking, there exists a straight path between any two elements in convex sets.
- Cone: if $x \in \mathcal{C}$, then for $\theta \geq 0$, we have $\theta x \in C$.
- Convex cone: the set $\mathcal{C}$ is a cone and is convex, i.e. any points of the form $x = \theta_1 x_1 + \theta_2 x_2$ with $\theta_1 \geq 0,\ \theta_2 \geq 0$. It should be clear that it sweeps out an area like 1, since the line segment connecting x_1 and x_2 is in the set, and for $\theta \geq 0,\ \theta x$ is also in the set.
- Hyperplane (affine and convex): $\{x \mid a^T x = b\}$, solution set of a single linear equation
- Half-space (convex): $\{x \mid a^T x \leq b\}$, solution set of a single linear inequality
- Ball (convex): with centre x_c and radius r, $B(x_c, r) = \{x \mid ||x - x_c||_2 \leq r\} = \{x_c + ru \mid ||u||_2 \leq 1\}$
- Ellipsoid (convex): $\{x \mid (x - x_c)^T P^{-1}(x - x_c) \leq 1\} = \{x_c + Au \mid ||u||_2 \leq 1\}$, $P \in \mathcal{S}^n_{++}$, P is symmetric positive definite. The centre of the ellipsoid is x_c, semi-axes are determined by the eigenvalues / eigenvectors of P^{-1}, or the singular values/ singular vectors of A.

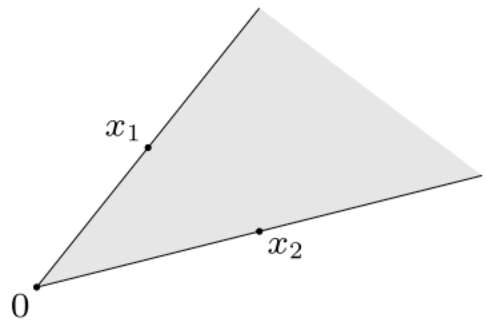

Figure 1: Convex cone

- Norm ball (convex): with centre x_c and radius r, $B(x_c, r) = \{x \mid ||x - x_c|| \leq r\}$, where $||\cdot||$ is any norm
- Norm cone (convex cone): $\{(x, t) \mid ||x|| \leq t\}$ If the norm is Euclidean norm, then it's called second-order cone.
- Polyhedron (convex): solution set of finitely many linear equalities and inequalities $Ax \preccurlyeq b$, $Cx = d$, where $\preccurlyeq$ is component-wise inequality. Polyhedron is the intersection of finite number of half-spaces and hyperplanes.
- Positive semi-definite cone (convex cone): $S^n_+ = \{X \in S^n \mid X \succcurlyeq 0\}$, $x \in S^n_+$ iff $z^T X z \geq 0$ for all z.
- Positive definite cone (convex cone): $S^n_{++} = \{X \in S^n \mid X \succ 0\}$, $x \in S^n_{++}$ iff $z^T X z > 0$ for all z.

There are mappings that preserve convexity of a set. It's clear that the intersection of any number of convex sets is convex. Moreover, If a set $\mathcal{C}$ is convex, then the image and inverse image of the set under following functions are convex.

- Affine functions ($f(x) = Ax + b$), examples include scaling, translation, and projection
- Perspective mapping
- Linear-fractional function

2.2 Generalised inequality

Inequalities can be defined in a more general sense, with respect to a proper cone.

Definition 2. Proper cone and generalised inequality
A convex cone $K \subseteq R^n$ is a proper cone if

- K is closed (contains its boundary)
- K is solid (interior is non-empty)
- K is pointed (contains no line)

Generalised inequalities can be defined with respect to a proper cone K:

$$x \preccurlyeq_K y \iff y - x \in K, \qquad x \prec_K y \iff y - x \in int\ K$$

Example. Generalised inequalities

- component-wise inequality ($K = R^n_+ = \{x \in R^n \mid x_i \geq 0,\ i = 1, ..., n\}$)

$$x \preccurlyeq_{R^n_+} y \iff x_i \leq y_i\ i = 1, ..., n$$

- matrix inequality ($K = S^n_+$)

$$X \preccurlyeq_{S^n_+} Y \iff Y - X \text{ positive semi-definite}$$

Note that $\preccurlyeq_K$ is a partial ordering: we can have elements that are incomparable. For example $(1, 0)$ and $(0, 1)$ is incomparable with respect to the positive orthant. Hence, we have the notion of minimum and minimal elements. $x \in S$ is the minimum elements of S with respect to $\preccurlyeq_K$ if it is comparable to all other elements y in S, and $x \preccurlyeq_K y$. $x \in S$ is the minimal elements of S with respect to $\preccurlyeq_K$ if $x \preccurlyeq_K y$ for every y comparable to x.

2.3 Dual cones

Definition 3. Dual cone of a cone K

$$K^* := \{y \mid y^T x \geq 0 \text{ for all } x \in K\}$$

More precisely, it should be

$$K^* := \{y \mid\ < y, x > \geq 0 \text{ for all } x \in K\}$$

where $< \cdot, \cdot >$ is an inner product.

Example. Dual cones

- positive orthant, positive definite cones [one should take the trace as the inner product], second order cones are self-dual, i.e. $K^* = K$.
- $K = \{(x, t) \mid ||x||_1 \leq t\} : \quad K^* = \{(x, t) \mid ||x||_\infty \leq t\}$

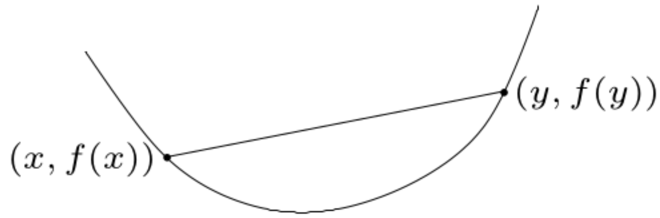

Figure 2: Convex functions

The dual cones of proper cones are proper, and hence also define generalised inequities:

$$y \succeq_{K^*} 0 \iff y \in K^* \iff y^T x \geq 0, \forall x \succeq_K 0$$

Proposition 4. *dual characterisation of minimum/minimal elements*

- x is minimum element of S with respect respect to $\preccurlyeq_K$ iff $\forall \lambda \succeq_K 0$, x is the unique minimiser of $\lambda^T z$ over $z \in S$.
- If x minimises $\lambda^T z$ over S for some $\lambda \succ_{K^*} 0$, then x is minimal

Proof. We prove the first statement by writing down the definitions.

$$x \leq_K y \quad \forall y \in S$$
$$y - x \in K \quad \forall y \in S$$
$$\lambda^T (y - x) \geq 0 \quad \forall \lambda \in K^*, y \in S$$
$$\lambda^T x \leq \lambda^T y \quad \forall \lambda \in K^*, y \in S$$
$$\text{x minimises } \lambda^T z$$

The second statement can be proved by contradiction. Assume $\exists \lambda \succeq_{K^*}$, $\lambda^T x < \lambda^T z$ for all $z \in S$. That is x minimises $\lambda^T z$ over S, but x is not minimal. That means $\exists z \in S, z \neq x, z \preccurlyeq_K x$. Then, $\lambda^T (x - z) > 0$, i.e. $\lambda^T x > \lambda^T z$↯. □

3 Convex functions

3.1 Basic properties

Definition 5. Convex functions
A function $f : R^n \to R$ is convex if *dom* f is a convex set and $\forall x, y \in$ *dom* f and $0 \leq \theta \leq 1$, we have

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

Geometrically, the line segment between any two points $(x, f(x))$ lies above the graph2. f is concave if $-f$ is convex. f is strictly convex if in the above definition , every $\leq$ is replaced with $<$. It's also clear that affine functions are both convex and concave.

Example. a rather comprehensive list can be found `https://www.cvxpy.org/tutorial/functions/index.html#scalar-functions`

- affine function: $a^T x + b$
- affine function: $tr(A^T X) + b$
- exponential $e^{\alpha x}, \forall \alpha \in R$
- powers: x^α on R_{++}, for $\alpha \geq 1$ or $\alpha \leq 0$
- negative entropy: $xlogx$ on R_{++}
- norm: $||x||_p$ for $p \geq 1$
- operator norm: $||X||_2 = \sigma_{max}(X) = (\lambda_{max}(X^T X))^{1/2}$, the maximum singular value [maximum amount X can stretch a vector by, it's equal to the square root of maximum eigenvalue of $X^T X$].

Example. Concave functions

- affine function: $a^T x + b$
- powers: x^α on R_{++}, for $0 \leq \alpha \leq 1$
- logarithm: $log\ x$ on R_{++}

The following definition provides a practical way of checking a function is not convex.

Definition 6. Restriction of convex function to a line
$f : R^n \to R$ is convex $\iff$ the function $g : R \to R$, $g(t) = f(x + tv)$ is convex
where $dom\ g = \{t | x + tv \in dom\ f\}$, $x \in dom\ f, v \in R^n$

Then one can check if a function is not convex, by forming two points $x_1 := x + t_1 v$ and $x_2 := x + t_2 v$, and check whether $f(x_1/2 + x_2/2) \leq f(x_1)/2 + f(x_2)/2$. If the inequality is violated, then one know for sure f is not convex. It should be noted that this method of checking two random points can't be used to establish convexity. However, the above definition is an iff, so can be used to prove a function is convex. We will see an example.

Example. $f : S^n \to R,\ f(X) = log\ det(X),\ dom\ f = S^n_{++}$ is concave

$$\begin{aligned}
g(t) &= log\ det(X + tV) \\
&= log\ det(X^{1/2}(I + tX^{-1/2} V X^{-1/2}) X^{1/2}) \\
&= log\ \left\{ det(X^{1/2}) det(I + tX^{-1/2} V X^{-1/2}) det(X^{1/2}) \right\} \\
&= 2log\ det(X^{1/2}) + log\ det(I + tX^{-1/2} V X^{-1/2}) \\
&= log\ det(X) + \sum_{i=1}^{n} log(1 + t\lambda_i), \text{ where } \lambda_i \text{ are eigenvalues of } X^{-1/2} \mathrm{V} X^{-1/2}
\end{aligned}$$

Since log of an affine function $1 + t\lambda_i$ is concave, sum of concave functions is concave, and $log\ det(X)$ is a constant, so $g(t)$ is concave in t [These operations preserving convexity will be defined later]. Hence f is concave.

Now, we define the first and second conditions of convex functions. In principle, one can establish convexity by showing these conditions.

Definition 7. First-order condition
Suppose f differentiable (i.e. *dom* f is open and $\nabla f(x)$ exists for every point x $\in$ *dom* f), then

$$f \text{ convex} \iff f(y) \geq \underbrace{f(x) + \nabla f(x)^T (y - x)}_{\text{First-order Taylor Expansion at x}}$$

for all $x, y \in$ *dom* f

The above definition says the first-order Taylor expansion of f is a global under-estimator. In particular if $\nabla f(x) = 0$, then $f(x) \leq f(y)\ \forall x, y \in$ *dom* f, x is a global minimum.

Definition 8. Second-order condition
Suppose f twice differentiable (i.e. *dom* f is open and the Hessian $\nabla^2 f(x) \in S^n$ exists for every point x $\in$ *dom* f), then

$$f \text{ convex} \iff \nabla^2 f(x) \succeq 0$$

for all $x \in$ *dom* f

Example. Convex functions

- Least Square objective

$$\begin{aligned} f(x) &= ||Ax - b||_2^2 \\ \nabla f(x) &= 2A^T(Ax - b) \\ \nabla^2 f(x) &= 2A^T A \succeq 0 \end{aligned}$$

- quadratic function

$$\begin{aligned} f(x) &= (1/2)x^T \overbrace{P}^{WLG.\ P \in S^n} x + q^T x + r \\ \nabla f(x) &= Px + q \\ \nabla^2 f(x) &= P \end{aligned}$$

  is convex iff $P \succeq 0$.

3.2 Operations preserving convexity

We shall state the general theorem [which CVXPY uses] for constructing convex functions, a tutorial can be found `https://dcp.stanford.edu/rules`. One could also consider atomic convex function as leaves of an expression tree, and operations preserving convexity combines nodes. After that we will look at some particular examples. However, it should be noted that although the composition theorem always results in the correct curvature classification, it may flag expressions as unknown. The order in which the nodes are combined is important for the curvature determination. Moreover, some functions can still be convex even if the composition theorem is inconclusive.

Theorem 9. *Curvature Rules for vector composition*

$f(expr_1, expr_1, ..., expr_n)$ is convex if f is a convex function and for each $expr_i$ one of the following conditions holds:

- f is increasing in argument i and $expr_i$ is convex
- f is decreasing in argument i and $expr_i$ is concave
- $expr_i$ is affine

$f(expr_1, expr_1, ..., expr_n)$ is concave if f is a concave function and for each $expr_i$ one of the following conditions holds:

- f is increasing in argument i and $expr_i$ is concave
- f is decreasing in argument i and $expr_i$ is convex
- $expr_i$ is affine

$f(expr_1, expr_1, ..., expr_n)$ is affine if f is a affine function and each $expr_i$ is affine.

Example. Operations preserving convexity

- Non-negative multiple: αf is convex if f is convex and $\alpha \geq 0$
- Sum: $f_1 + f_2$ is convex if f_1 and f_2 are convex (extends to integrals and infinite sums)
- Composition with affine function: $f(Ax + b)$ is convex if f is convex
- Point-wise maximum: $f(x) = max\{f_1(x), ..., f_m(x)\}$ is convex if $f_1, ..., f_m$ are convex. For example, piece-wise linear function: $f(x) = max_{1 \leq i \leq m}\{a_i^T x + b_i\}$.
- Point-wise supremum: If $f(x, y)$ is convex in $x, \forall y \in \mathcal{A}$, then $g(x) := \sup_{y \in \mathcal{A}} f(x, y)$ is convex. For example, distance to farthest point in a set C: $f(x) = \sup_{y \in C} ||x - y||$
- Minimisation: If $f(x, y)$ is convex in (x, y) and C is a convex set, then $g(x) = \inf_{y \in C} f(x, y)$ is convex

3.3 Epigraph, sub-level set and conjugate functions

These are important notions in convex analysis. We define them here in case they are used later in the project.

Definition 10. α sub-level set of $f : R^n \to R$

$$C_\alpha = \{x \in dom\ f \mid f(x) \leq \alpha\}$$

α sub-level sets is a convex function are convex sets [converse is false].

Definition 11. epigraph of $f : R^n \to R$

$$epi\ f = \{(x, t) \mid x \in dom\ f, t \geq f(x)\}$$

f is convex $\iff$ $epi\ f$ is a convex set

Definition 12. The conjugate of f is

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

The conjugate function is convex even if f is not.

Let's see a simple example of conjugate function. There is also a table of conjugate functions online that we can look up. The notion of conjugate function will be used to study duality.

Example. $f(x) = \frac{1}{2}x^T Qx,\ Q \in S^n_{++}$

$$\begin{aligned} f^*(y) &= \sup_x (y^T x - \frac{1}{2}x^T Qx) \\ \frac{\partial}{\partial x} y^T x - \frac{1}{2}x^T Qx = y - Qx = 0 \implies x = Q^{-1}y & \\ f^*(y) &= y^T Q^{-1} y - \frac{1}{2} y^T Q^{-1} Q Q^{-1} y \\ f^*(y) &= \frac{1}{2} y^T Q^{-1} y \end{aligned}$$

3.4 Log-concave and Log-convex functions

It's often the case that in Statistics, one look at the log-likelihood function. In fact, the log-likelihood function in exponential family is concave.

Definition 13. Log concavity
A positive function f is log-concave if $log\ f$ is concave:

$$\begin{aligned} log\ f(\theta x + (1-\theta)y) &\geq \theta log\ f(x) + (1-\theta) log\ f(y) \\ f(\theta x + (1-\theta)y) &\geq f(x)^\theta f(y)^{1-\theta} \end{aligned}$$

Log-concave functions have several desirable properties.

- product of log-concave functions is log-concave [sum of log-concave functions not necessarily log-concave]
- integration: if $f : R^n \times R^m \to R$ is log-concave, then $g(x) := \int f(x, y)dy$ is log-concave
- convolution $f * g$ of log-concave functions f, g, $(f * g)(x) := \int f(x - y)g(y)dy$ is log-concave

Example. If $C \subseteq R^n$ is convex, and y is a random variable with log-concave PDF, then

$$f(x) = Pr(x + y \in C)$$

is log-concave.

Proof. Define $g(u)$ as the indicator

$$g(u) := \begin{cases} 1 & u \in C \\ 0 & u \notin C \end{cases}$$

$$log\ g(u) = \begin{cases} 1 & u \in C \\ -\infty & u \notin C \end{cases} \text{ is concave}$$

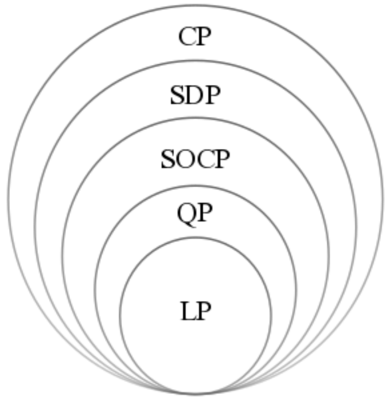

Figure 3: hierarchy of optimisation problem

Hence $f(x) = \int g(x+y)p(y)dy$ is an integral of product of log-concave functions, so it's log-concave. □

4 Convex Optimisation problems

We shall see an hierarchy of optimisation problems, the list goes like this: LP, QP, SOCP, SDP, CP3. Oh, there is also GP, that is one level below CP.

4.1 Convex optimisation in standard form

Recall the standard form of an optimisation problem. For convex problems, we require $f_i(x)$ to be convex, $h_i(x)$ to be affine.

Definition 14. Convex program in standard form

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \leq 0, \quad i = 1, ..., m \\ &\qquad\qquad a_i^T x = b_i, \quad i = 1, ..., p \iff Ax = b \end{aligned}$$

where

- $x \in \mathcal{R}^n$ is the optimisation variable
- $f_0 : \mathcal{R}^n \to \mathcal{R}$ is the objective function
- $f_i : \mathcal{R}^n \to \mathcal{R}$ are the inequality constraint functions

We have the following important properties for convex optimisation problems.

Proposition 15. *The feasible set of a convex problem is convex.*

Proof. It follows from "intersection of convex sets is convex". Since the feasible set is the intersection of *dom* $f_i, i = 0, ..., m$, 0 sub-level sets of $f_i, i = 1, ..., m$, and affine set defined by $Ax = b$. □

Proposition 16. *Any locally optimal point of a convex problem is globally optimal.*

Proof. Suppose x is locally optimal, but there exists a feasible y with strictly smaller objective value, $f_0(y) < f_0(x)$. x is locally optimal means there exists an $R > 0$ such that

$$||z - x||_2 \leq R \implies f_0(z) \geq f_0(x)$$

Let's consider $z = \theta y + (1 - \theta)x$, and let $\theta = \frac{R}{2||y-x||_2}$. Since x is locally optimal, it must be $||y - x||_2 > R$, so $0 < \theta < 1/2$. Now since z is a convex combination of x and y, z is feasible. Moreover, z is within the ball centred around x, since

$$\begin{aligned}
||z - x||_2 &= ||\theta y + (1 - \theta)x - x||_2 \\
&= ||\theta(y - x)||_2 \\
&= ||\frac{R(y - x)}{2||y - x||_2}||_2 \\
&= \frac{R}{2}
\end{aligned}$$

We have a contradiction:

$$\begin{aligned}
f_0(z) &\leq \theta f_0(x) + (1 - \theta)f_0(y) \\
&\leq f_0(x) \lightning
\end{aligned}$$

$\square$

Remark. There is also an optimality criterion for differentiable f_0 for convex optimisation problems, it says x is optimal $\iff \nabla f_0(x)^T(y-x) \geq 0$ for all feasible y. That is, one can check if a one is not optimal, by finding a y that violates the inequality. The criterion simplifies if we have unconstrained optimisation or just equality constrained ones.

4.2 Hierarchy of optimisation problems

4.2.1 Linear programs

In linear program, everything (the objective and constraint functions) are affine.

Definition 17. Linear program

$$\begin{aligned}
\text{minimize } & c^T x + d \\
\text{subject to } & Gx \preccurlyeq h \\
& Ax = b
\end{aligned}$$

The feasible set of a linear program is a polyhedron. Many well-studied problems are linear programs, for example the diet problem, fractional knapsack, shortest path, max-flow etc. Although integer knapsack problem is of course ILP [Integer linear program which is NP]. We shall look at another well-studied problem called Boolean LP, and it will illustrate some important ideas. Simple implementations using CVXPY can be found in appendices. A quick start guide to CVXPY can be found `https://www.cvxpy.org/`.

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

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } Ax \preccurlyeq b \\ &\qquad x_i \in \{0,1\}, i = 1, ..., n \end{aligned}$$

Boolean LP is NP-hard, but we shall consider the LP relaxation of the Boolean LP:

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } Ax \preccurlyeq b \\ &\qquad 0 \leq x_i \leq 1, i = 1, ..., n \end{aligned}$$

in which, the constraint on x_i is replaced with a linear inequality $0 \leq x_i \leq 1$. It's clear that the optimal value of the relaxed LP provides a lower bound on the Boolean LP.

A heuristic for solving the boolean LP is based on its relaxation. The solution x^{rlx} for the relaxed LP can be used to make a guess $\hat{x}$ of the Boolean LP, by rounding its entries based on a threshold $t \in [0,1]$:

$$\hat{x} = \begin{cases} 1 & x_i^{rlx} \geq t \\ 0 & otherwise \end{cases}$$

4.2.2 Quadratic programming

In QP, the objective is a convex quadratic. Least square problem $min\ ||Ax - b||_2^2$ is a QP, since

$$\begin{aligned} ||Ax - b||_2^2 &= (Ax - b)^T (Ax - b) \\ &= (x^T A^T - b^T)(Ax - b) \\ &= x^T (A^T A) x - 2(A^T b)^T x + ||b||_2^2 \end{aligned}$$

Definition 18. QP

$$\begin{aligned} &\text{minimize } (1/2) x^T P x + q^T x + r \\ &\text{subject to } Gx \preccurlyeq h \\ &\qquad Ax = b \end{aligned}$$

where $P \in S_+^n$

We shall see an application to portfolio optimisation once we see how to do multi-objective optimisation [In portfolio optimisation we wish to minimise the negative expected return and the variance of the return.

4.2.3 Second order cone programming

Definition 19. SOCP

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

where $A_i \in R^{n_i \times n}, F \in R^{p \times n}$

SOCP generalises LP and QP, it's called SOCP because the inequality second order cone constraints: $(A_i x + b_i, c_i^T x + d_i) \in$ second order cone in R^{n_i+1}. We shall see an application to robust linear programming.

In robust LP, we want to handle uncertainty in problem parameters. For example [this is the deterministic approach], we solve

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } a_i^T x \leq b_i \text{ for all } a_i \in \mathcal{E}_i, i = 1, ..., m \end{aligned}$$

where $\mathcal{E}_i = \{\bar{a}_i + P_i u \mid ||u||_2 \leq 1\}$, is a ellipsoid with centre $\bar{a}_i$, semi-axes determined by singular values / singular vectors of P_i. This is equivalent to the following SOCP:

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } \bar{a}_i{}^T x + ||P_i^T x||_2 \leq b_i, i = 1, ..., m \end{aligned}$$

since

$$\begin{aligned} \sup_{u:||u||_2 \leq 1} (\bar{a}_i + P_i u)^T x &= \sup_{u:||u||_2 \leq 1} \bar{a}_i{}^T x + u^T (P_i^T x) \\ &= \bar{a}_i{}^T x + ||P_i^T x||_2 \end{aligned}$$

There is also a stochastic model of robust LP, in which a_i is modelled as a random variable and is assumed multivariate normal $a_i \sim N_p(\bar{a}_i, \Sigma_i)$. Then, $a_i^T x$ has distribution $a_i^T x \sim N(\bar{a}_i{}^T x, x^T \Sigma_i x)$.

$$\begin{aligned} Pr(a_i^T x \leq b_i) &= Pr(z \leq \frac{b_i - a_i^T x}{\sqrt{x^T \Sigma_i x}}), z \sim N(0,1) \\ &= \Phi(\frac{b_i - a_i^T x}{||\Sigma_i^{1/2} x||_2}) \end{aligned}$$

The stochastic LP:

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } Pr(a_i^T x \leq b_i) \geq \eta, i = 1, ..., m \end{aligned}$$

is therefore equivalent to the SOCP:

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } \Phi(\frac{b_i - a_i^T x}{||\Sigma_i^{1/2} x||_2}) \geq \eta, i = 1, ..., m \\ &\text{or } \bar{a}_i{}^T x + \Phi^{-1}(\eta) ||\Sigma_i^{1/2} x||_2 \leq b_i \end{aligned}$$

Theorem 2.B.4 **Positive (Semi)definiteness of 2×2 Block Matrices**

Suppose $A \in \mathcal{M}_n$ is invertible and $S = C - B^*A^{-1}B$ is the Schur complement of A in the self-adjoint 2×2 block matrix

$$Q = \begin{bmatrix} A & B \\ B^* & C \end{bmatrix}.$$

Then Q is positive (semi)definite if and only if A and S are positive (semi)definite.

Figure 4: Shur Complement

4.2.4 Semi-definite programming

Finally, let's look at SDP.

Definition 20. SDP

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } x_1F_1 + x_2F_2 + ... + x_nF_n + G \preccurlyeq 0, \quad i = 1, ..., m \\ &\qquad Ax = b \end{aligned}$$

This generalises SOCP since the inequality constraint $||A_ix + b_i||_2 \leq c_i^T x + d_i$ is exactly the requirement for the Shur complement 4 for the following SDP:

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } \begin{bmatrix} (c_i^T x + d_i)I & A_ix + b_i \\ (A_ix + b_i)^T & c_i^T x + d_i \end{bmatrix} \succeq 0, \quad i = 1, ..., m \end{aligned}$$

One application is in matrix norm minimisation in which we want to minimise $||A(x)||_2 = \sigma_{max}(A(x)) = (\lambda_{max}((A(x)^T(A(x)))^{1/2}$, $A(x) = A_0 + x_1A_1 + ... + x_nA_n$. This is equivalent to the following SDP [again can be shown through the Shur complement]:

$$\begin{aligned} &\text{minimize } t \\ &\text{subject to } \begin{bmatrix} tI & A(x) \\ A(x)^T & tI \end{bmatrix} \succeq 0 \end{aligned}$$

4.3 Multi-objective optimisation

In multi-objective optimisation or vector optimisation, we want to minimise an objective $f_0 : R^n \to R^q$, with respect to some proper cone $K \in R^q$. One would expect to find a set of minimal x rather than the minimum.

Definition 21. Vector optimisation problem

$$\begin{aligned}\text{minimise (w.r.t. K) } & f_0(x)\\ \text{subject to } & f_i(x) \leq 0, i = 1, ..., m\\ & Ax = b\end{aligned}$$

where $f_0(x)$ is K-convex i.e. $f(\theta x + (1-\theta)y) \preceq_K \theta f(x) + (1-\theta)f(y)$
$f_1(x), ..., f_m(x)$ are convex

We denote the set of attainable objective values

$$\mathcal{O} = \{f_0(x) \mid x \text{ feasible}\}$$

then,

- a feasible x is optimal if $f_0(x)$ is the minimum of $\mathcal{O}$
- Pareto optimal if $f_0(x)$ is a minimal value of $\mathcal{O}$

If there exists an optimal x, then the objectives are not competing. On the other hand, if there are multiple Pareto optimal points, then there is a trade-off between the objectives. But how to we find Pareto optimal points ?

Recall the dual cone K^* of K is defined as

$$K^* := \{y \mid y^T x \geq 0 \text{ for all } x \in K\}.$$

If x minimises $\lambda^T z$ over K for some $\lambda \in K^*$[*i.e.* $\lambda \succeq_{K^*} 0$], then x is minimal. Therefore, to find Pareto optimal points, we minimise the positive weighted sum

$$\lambda^T f_0(x) = \lambda_1 f_{0,1}(x) + ... + \lambda_q f_{0,q}(x)$$

for some $\lambda \succeq_{K^*} 0$.

Example. In regularised least-square, we want to minimise $||Ax - b||_2^2$ while keeping $||x||_2^2$ small. We set $\lambda = (1, \gamma)$ [R_+^2 is self-dual], so we minimise $||Ax - b||_2^2 + \gamma||x||_2^2$.

Remark. There are many reasons for why we want to keep $||x||_2^2$ small. In particular one could show, regularised least-square is equivalent to stochastic least square in which the matrix A is assumed $A = \bar{A} + U$, where U is a random matrix, $E[U] = 0$, and $E[U^T U] = \delta I$, and we wish to minimise the expected value $E[||Ax - b||_2^2]$.

Now, let's look at an application to portfolio optimisation. There is an implementation using CVXPY in appendices. In portfolio optimisation, we minimise the negative expected return, and variance of the return, subject to the constraint: the portfolio variable x satisfies $1^T x = 1$. So the problem can be formulated as

$$\begin{aligned}\text{minimise (w.r.t. } R_+^2) & - \mu^T x + \gamma x^T \Sigma x\\ \text{subject to } & 1^T x = 1\\ & x \succeq 0 \text{ [for long only portfolio]}\end{aligned}$$

for some $\gamma > 0$.

- $p \in R^n$ is vector of relative asset price changes in one market period, and is assume to have $E[p] = \mu$, $Var[p] = \Sigma$.
- $\mu^T x$ is the expected return, $x^T \Sigma x$ is the variance of the return.

Or equivalently the objective can be written as $max\ \mu^T x - \gamma x^T \Sigma x$, where $\mu^T x - \gamma x^T \Sigma x$ is the risk-adjusted return. The idea is: there is a trade-off curve between expected return and the risk.

4.4 Quasi-convex optimisation

Let's briefly mention quasi-convex optimisation, in which the objective f_0 is quasi-convex.

Definition 22. Quasi-convex functions
$f : R^n \to R$ is quasi-convex if $dom\ f$ is convex, and sub-level sets

$$S_\alpha = \{x \in dom\ f \mid f(x) \leq \alpha\}$$

are convex for all α.

Quasi-convex problems can have locally optimal points that are not globally optimal. Quasi-convex problems are solved using a binary search, on the optimal values of the related [through the sub-level sets, that is, for a fix α, $f_0(x) \leq \alpha \iff$ the constraint $\phi_\alpha(x) \leq 0$ holds, for a related convex in x function ϕ] convex feasibility problems. Hence the running time for solving a quasi-convex problem is of logarithmic multiple of convex problems.

5 Duality

Why bother with the dual problem of an optimisation problem? For a general optimisation problem, the dual problem maximises a related objective and is always convex. It provides a cheap [to compute] lower bound on the primal. If the primal problem is convex [and Slater condition is satisfied], then the optimal value of the dual is equal to the optimal value of the primal.

5.1 Lagrangian Dual function

Recall optimisation problems in standard form:

$$\begin{aligned} \text{minimize } & f_0(x) \\ \text{subject to } & f_i(x) \leq 0, \quad i = 1, ..., m \\ & h_i(x) = 0, \quad i = 1, ..., p \end{aligned}$$

with $x \in R^n$, domain $\mathcal{D}$,and optimal value p^*

Definition 23. Lagrangian dual function
Lagrangian is a function $\mathcal{L} : R^n \times R^m \times R^p \to R$, with $dom\ \mathcal{L} = D \times R^m \times R^p$,

$$\mathcal{L}(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

where λ_i and ν_i are the Lagrange multipliers associated with $f_i(x) \le 0$ and $h_i(x) = 0$

The Lagrange dual function $g : R^m \times R^p \to R$ is

$$\begin{aligned} g(\lambda, v) &= \inf_{x \in \mathcal{D}} \mathcal{L}(x, \lambda, \nu) \\ &= \inf_{x \in \mathcal{D}} \left\{ f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right\} \end{aligned}$$

Since the Lagrange dual function $g(\lambda, \nu)$ is point-wise infimum of an affine function in (λ, ν), $g(\lambda, \nu)$ is concave in (λ, ν).

Proposition 24. *The dual function $g(\lambda, \nu)$ is a lower bound on p^* if $\lambda \succeq 0$.*

Proof. The following holds for all feasible x

$$\begin{aligned} f_0(x) &\ge \mathcal{L}(x, \lambda, \nu) \text{ since } [\lambda_i f_i(x) \le 0,\ \nu_i h_i(x) = 0] \\ &\ge \inf_{x \in \mathcal{D}} \mathcal{L}(x, \lambda, \nu) \\ &= g(\lambda, \nu) \end{aligned}$$

In particular, it holds for x^*, so $p^* = f_0(x^*) \ge g(\lambda, \nu)$ □

Let's look at an interesting application of Lagrange dual function to (non-convex) two-way partitioning problem. Also, we will look at a formulation using Boolean relaxation, and show it's equivalent to the Lagrange relaxation. Two-way partitioning problem is

$$\begin{aligned} &\text{minimize } x^T W x \\ &\text{subject to } x_i^2 = 1, \quad i = 1, ..., n \end{aligned}$$

, where x_i takes values $\{-1, 1\}$, if $x_i = 1$ then it is assigned to one partition, -1 the other partition. $W_{i,j}$ represents the cost of assigning i, j to the same partition. This is of course, a non-convex problem. We form the dual function $g(\nu)$, and take a particular ν to form a famous lower bound.

$$\begin{aligned} g(\nu) &= \inf_x (x^T W x + \sum_i \nu_i (x_i^2 - 1)) \\ &= \inf_x x^T (W + diag(\nu)) x - \mathbf{1}^T \nu \\ &= \begin{cases} -\mathbf{1}^T v & W + diag(\nu) \succeq 0 \\ & -\infty\ otherwise \end{cases} \end{aligned}$$

since W is real symmetric, we are minimising a quadratic form. Now if we take $\nu = -\lambda_{min}(W)\mathbf{1}$, then $p^* \ge n\lambda_{min}(W)$.

Alternatively, we can derive the same bound using Boolean relaxation, it gives rise to an approximation algorithm for the (non-convex) two-way partitioning problem. Since $x_i^2 = 1, i = 1, ..., n$, i.e. x is a vertex of a hypercube in R^n, we know for sure $\sum_{i=1}^n x_i^2 = n$. Moreover, since $\inf_{x:||x||_2^2=1} x^T W x = \lambda_{min}(W)$, then $\inf_{x:||x||_2^2=n} x^T W x = n\lambda_{min}(W)$, where x_{min} is a eigenvector associated with λ_{min}. This suggests an approximation algorithm [spectral partitioning], we take $\hat{x}_i = Sign(x_{min,i})$. We could also do a greedy improvement on $\hat{x}$, by flipping the bits of $\hat{x}$ until the objective doesn't improve.

5.2 Relationship with dual function

If we have just linear equality and inequality constraints $Ax \preceq b, Cx = b$, then the derivation of the dual can be simplified using the conjugate function.

$$\begin{aligned}
g(\lambda, \nu) &= \inf_{x \in dom\ f_0} (f_0(x) + \lambda^T(Ax - b) + \nu^T(Cx - d)) \\
&= \inf_{x \in dom\ f_0} (f_0(x) + (A^T\lambda + C^T\nu)^T x) - b^T\lambda - d^T\nu \\
&= -\sup_{x \in dom\ f_0} (-(A^T\lambda + C^T\nu)^T x - f_0(x)) - b^T\lambda - d^T\nu \\
&= -f_0^*(-A^T\lambda - C^T\nu) - b^T\lambda - d^T\nu
\end{aligned}$$

5.3 Weak and strong duality

We define the Lagrange dual problem. Recall $g(\lambda, \nu)$ is lower bound on p^* if $\lambda \succeq 0$. The dual problem aims to maximises $g(\lambda, \nu)$ in (λ, ν), subject to $\lambda \succeq 0$.

Definition 25. Lagrange dual problem

$$\begin{aligned}
&\text{maximise } g(\lambda, \nu) \\
&\text{subject to } \lambda \succeq 0
\end{aligned}$$

- For a general primal, the dual problem is a convex optimisation problem since the we are maximising a concave function, subject to an affine inequality constraint.
- (λ, ν) are dual feasible if $\lambda \succeq 0$, $(\lambda, \nu) \in dom\ g$.
- We denote the optimal value d^*, then weak duality says $d^* \leq p^*$.
- Strong duality says: $d^* = p^*$, i.e. the duality gap is 0. Strong duality holds if the problem is convex and Slater's constraint qualification holds. Note that there are other types of constraint qualification.

Definition 26. Slater's constraint qualification
Strong duality holds for a convex problem

$$\begin{aligned}
&\text{minimize } f_0(x) \\
&\text{subject to } f_i(x) \leq 0, \quad i = 1, ..., m \\
&Ax = b
\end{aligned}$$

if it is strictly feasible, i.e. $\exists x \in int\ D : f(x) \leq 0, \quad i = 1, ..., m, \quad Ax = b$.

Now, we shall state the widely-used KKT conditions, which is an optimality condition for convex optimisation problems.

Definition 27. KKT conditions

1. primal constraints: $f_i(x) \leq 0, i = 1, ..., m \quad h_i(x) = 0, i = 1, ..., p$
2. dual constraints: $\lambda \succeq 0$
3. complementary slackness: $\lambda_i f_i(x) = 0, i = 1, ..., m$
4. gradient of $\mathcal{L}$ w.r.t. x, $\nabla_x \mathcal{L}$ vanishes: $\nabla f_0(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) + \sum_{i=1}^p \nu_i \nabla h_i(x) = 0$

If strong duality duality holds and x, λ, ν are optimal, then they must satisfy the KKT conditions. The converse is also true, if $\tilde{x}, \tilde{\lambda}, \tilde{\nu}$ satisfy the KKT conditions for a convex problem, then they are optimal.

To see why we include 3 and 4 in the KKT conditions, assume x^* is primal optimal, (λ^*, ν^*) is dual optimal.

$$
\begin{aligned}
f_0(x^*) &= g(\lambda^*, \nu^*) \\
&= \inf_x \left\{ f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x) \right\} \\
&\leq f_0(x^*) + \sum_{i=1}^m \lambda_i^* f_i(x^*) + \sum_{i=1}^p \nu_i^* h_i(x^*) \\
&\leq f_0(x^*) \text{ [since the primal constrints hold]}
\end{aligned}
$$

Therefore, $f_0(x^*) = \inf_x \{f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x)\} = f_0(x^*) + \sum_{i=1}^m \lambda_i^* f_i(x^*) + \sum_{i=1}^p \nu_i^* h_i(x^*)$. That is, x^* minimises $\mathcal{L}(x, \lambda^*, \nu^*)$. Since $\mathcal{L}(x, \lambda^*, \nu^*)$ is convex in x, we minimises $\mathcal{L}(x, \lambda^*, \nu^*)$ by setting $\nabla_x \mathcal{L} = 0$. Moreover, $\lambda_i^* f_i(x^*) = 0$ for $i = 1, ..., m$. This also implies if

$$\lambda_i^* > 0 \implies f_i(x^*) = 0, \qquad f_i(x^*) < 0 \implies \lambda_i^* = 0$$

One particular application of KKT conditions is in support vector machine [I will omit the details here]. In which, we derive the dual optimisation problem for the linear SVM. Then, we change the scalar product in the dual objective, to some other kernel functions. This leads to non-linear SVMs.

5.4 Perturbation and sensitivity analysis

Definition 28. Perturbed problem and its dual

$$
\begin{aligned}
\text{Perturbed problem: } & \text{minimise: } f_0(x) \\
& \text{subject to: } f_i(x) \leq u_i, i = 1, ..., m \\
& \qquad\qquad\quad h_i(x) \leq v_i, i = 1, ..., p \\
\\
\text{dual: } & \text{maximise: } g(\lambda, \nu) - u^T \lambda - v^T \nu \\
& \text{subject to: } \lambda \succeq 0
\end{aligned}
$$

The perturbed problem is convex if the unperturbed problem is convex. We denote $p^*(u,v)$ the optimal value of the perturbed problem, as a function of (u,v). We want to learn about $p^*(u,v)$ from the unperturbed problem. That is, we want to know the sensitivity of the objective function to some perturbation in the constraint functions. There are global and local analysis, let's just look at a global result.

Assume strong duality holds for the unperturbed problem, and λ^*, ν^* are dual optimal for unperturbed problem. Then weak duality for the perturbed problem suggests

$$\begin{aligned} p^*(u,v) &\geq g(\lambda^*, \nu^*) - u^T\lambda^* - v^T\nu^* \\ &= \underbrace{p^*(0,0)}_{\text{unperturbed optimal value}} - u^T\lambda^* - v^T\nu^* \\ &= p^*(0,0) - \sum_{i=1}^{m} \lambda_i u_i - \sum_{i=1}^{p} \nu_i v_i \end{aligned}$$

This says [notice the asymmetry here]

- if λ_i^* is large, then p^* increases greatly if we tighten constraint i ($u_i < 0$)
- if λ_i^* is small, then p^* does not decrease much if we loosen constraint i ($u_i > 0$)
- One could also make similar arguments for ν_i.

6 Miscellaneous applications

6.1 Penalty function approximation

In norm approximation, we minimise $||Ax - b||$. If $||\cdot||$ is Euclidean norm, then we get least-square. Penalty function approximation is a generalisation of norm approximation, in which we

$$\begin{aligned} \text{minimise: } & \phi(r_1) + ... + \phi(r_m) \\ \text{subject to: } & r = Ax - b \end{aligned}$$

where $A \in R^{m\times n}, \phi : R \to R$ is a convex penalty function.

Some commonly used penalty functions are: absolute value $\phi(u) = |u|$, quadratic $\phi(u) = u^2$, dead-zone linear with some width α, log-barrier with some limit α 5etc. Different penalty function result in different penalty for errors. For example, quadratic penalty function, i.e. least-square, penalises very little for small errors r_i, while penalising heavily for large errors r_i. On the other hand, absolute value function, i.e. L_1 norm minimisation, penalises equally for large and small errors. Therefore absolute value penalty fuction, there is stronger incentive [which is the slope of penalty function] for the optimisation algorithm to minimise the some error r_i to 0, that is you would get a sparse r. This of course comes at a price [it would results in more large error r_i compared to quadratic], but same goes for the quadratic. One should choose the penalty function to suit particular applications. For example, Huber penalty function penalises linearly for large r_i, while penalising quadratically for small r_i.

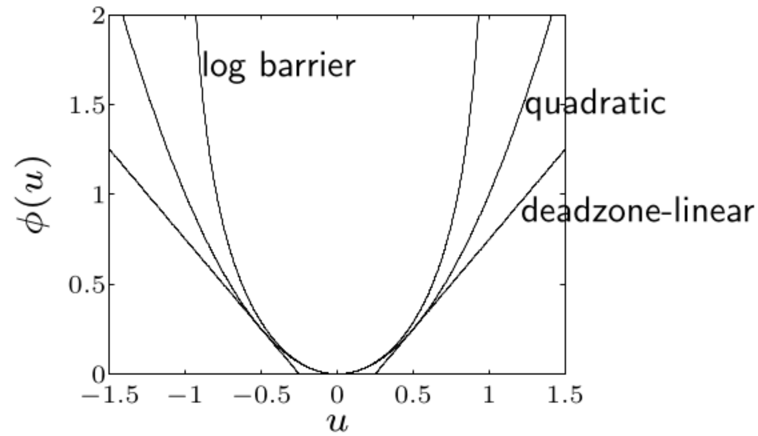

Figure 5: common penalty functions

7 Bibliography

This project is based on Professor Boyd and Vandenberghe's book on convex optimisation. See Boolean LP and portfolio_optimisation for two examples using CVXPY.