Convex Optimisations

Arpan Dasgupta and Abhishek Mittal

IIITH Theory Group

October 31, 2020

What is Mathematical Optimisation

Optimisation is a common problem in various fields. Mathematical optimisation is the problem where we try to minimise or maximise the value of a function while satisfying some constraints.
The general optimisation problem is of the form -

$$\begin{aligned} \textit{minimise} \quad & f_0(x) \\ \text{subject to:} \quad & f_i(x) \leq 0 \quad i = 1 \ldots m \\ & h_i(x) = 0 \quad i = 1 \ldots p \end{aligned}$$

Optimisation is useful in many places like ML, Electronics, Manufacturing .etc.

Convex Function

Affine function - $f(ax + (1-a)y) = af(x) + (1-a)f(y)$ A convex function is one for which the epigraph is a convex set. Other equivalent definitions are -

1. $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$
2. $f(y) \geq f(x) + \nabla f(x)^T(y-x)$
3. $\nabla^2 f(x) \geq 0$

Convex Optimisation

An optimisation problem where the objective function, inequailty constraints is convex and the equality constraints are affine is called a convex optimisation problem.

$$\begin{array}{rll} \textit{minimise} & f_0(x) & \\ \text{subject to:} & f_i(x) \leq 0 & i = 1 \ldots m \\ & h_i(x) = 0 & i = 1 \ldots p \end{array}$$

All f_i are convex and h_i are affine.

Why Convex Optimisation

For convex functions -

- Local optimality implies global optimality. Which makes the convex objective much easier to solve.
- Duality such as min-max relation and separation theorem holds. (Not explained now)

Even though in practice most objective functions are non-convex, convex optimisation still helps as -

- Several non-convex functions can be converted into equivalent convex functions. ('convexification')
- Many non-convex problems can be estimated or a bound can be given by using convex functions.

Extended Value Function

It is often convenient to extend a convex function to all of Rn by defining its value to be outside its domain. The extended value function is defined as follows -

$$\tilde{f}(x) = \begin{cases} f(x), & x \in \text{dom } f \\ \infty, & x \notin \text{dom } f \end{cases}$$

An extended value function of a convex function is convex. This helps us to generalise the functions easily.

An Example

The pointwise maximum of a set of convex functions is convex.
Let $f(x) =$ sum of k largest elements of a vector.
Then, the function f is convex. **Why?**

Lagrangian

The Lagrange dual function incorporates the problem constraints into the problem statement by introducing additional terms into the objective.

The Lagrange Dual function is defined as -

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

Where:

f_0 is the objective

f_i's are inequality constraints

h_i's are equality constraints

Lagrange Dual Function

The Lagrange Dual Function is defined as :
$g(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu)$
g is always concave even if f is non-convex.
Lower Bound Property - $g(\lambda, \nu) \leq p^*$
Where p^* is the optimal value of function f.

Problem Statement

Assumption: A solver already exists for any kind of convex optimisation problem

Problem Statement

Assumption: A solver already exists for any kind of convex optimisation problem

Partition n elements into 2 groups, where putting 2 elements in the same group incurs a cost. We want to find the optimal/minimum value for the cost incurred.

Problem Statement

Assumption: A solver already exists for any kind of convex optimisation problem
Partition n elements into 2 groups, where putting 2 elements in the same group incurs a cost. We want to find the optimal/minimum value for the cost incurred.
More formally, there exists a matrix C whose $(i, j)^{th}$ element tells us the cost that would be incurred if the i^{th} element and the j^{th} element are put in the same group.

Steps to mathematically formulate any optimisation problem

Step 1: First define the optimisation variable

Steps to mathematically formulate any optimisation problem

Step 1: First define the optimisation variable
Step 2: Define the objective function

Steps to mathematically formulate any optimisation problem

Step 1: First define the optimisation variable
Step 2: Define the objective function
Step 3: Define the constraint functions both inequality and equality

Steps to mathematically formulate any optimisation problem

Step 1: First define the optimisation variable
Step 2: Define the objective function
Step 3: Define the constraint functions both inequality and equality
Step 4: Mathematical Formulation of the optimisation problem

Defining the Optimisation Variable

The optimisation variable x must represent the group that each element belongs to. i^{th} component of the vector is the i^{th} element in the set. So we can choose any 2 set of values to represent group 1 and group 2 respectively.

Defining the Optimisation Variable

The optimisation variable x must represent the group that each element belongs to. i^{th} component of the vector is the i^{th} element in the set. So we can choose any 2 set of values to represent group 1 and group 2 respectively.

Examples $\begin{bmatrix} 1 \\ -1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$

Defining the Optimisation Variable

The optimisation variable x must represent the group that each element belongs to. i^{th} component of the vector is the i^{th} element in the set. So we can choose any 2 set of values to represent group 1 and group 2 respectively.

Examples $\begin{bmatrix} 1 \\ -1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$ or $\begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$

Defining our Optimisation function

Input- vector x , information about the partition

Defining our Optimisation function

Input- vector x , information about the partition
Output - a scalar value that tells how much cost is incurred for this particular partition

Defining our Optimisation function

Input- vector x , information about the partition
Output - a scalar value that tells how much cost is incurred for this particular partition

For one element $x_i \sum_{j=1}^{N} x_j C_{ij}$

$$Total = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j C_{ij}$$

Defining our Optimisation function

Input- vector x , information about the partition

Output - a scalar value that tells how much cost is incurred for this particular partition

For one element $x_i \sum_{j=1}^{N} x_j C_{ij}$

$$Total = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j C_{ij}$$

Matrix Notation

$$\frac{1}{2} x^T C x$$

Defining our Optimisation function

Input- vector x , information about the partition
Output - a scalar value that tells how much cost is incurred for this particular partition

For one element $x_i \sum_{j=1}^{N} x_j C_{ij}$

$$Total = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j C_{ij}$$

Matrix Notation

$$\frac{1}{2} x^T C x$$

$$x^T W x$$

Defining our Optimisation function

Input- vector x , information about the partition
Output - a scalar value that tells how much cost is incurred for this particular partition

For one element $x_i \sum_{j=1}^{N} x_j C_{ij}$

$$Total = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j C_{ij}$$

Matrix Notation

$$\frac{1}{2} x^T C x$$

$$x^T W x$$

Quadratic form.

Quadratic forms

$$ax^2 + 2bxy + cy^2$$

Quadratic forms

$$ax^2 + 2bxy + cy^2$$

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Defining our constraints

$$x_i^2 = 1 \ \forall i = 1, 2, ..., n$$

Mathematical Formulation of the Problem

$$\textit{minimise} \quad x^T W x$$
$$\text{subject to:} \quad x_i^2 = 1 \quad i = 1 \ldots n$$

Objective function is a quadratic form.
Convex $\iff$ $W \succeq 0$
Problem is non convex
Objective function-may or may not be convex
Constraints - Not affine functions

Finding the dual function

Lagrange function: The penalty function for violating constraints

Finding the dual function

Lagrange function: The penalty function for violating constraints

$L(x, \lambda, \nu) = f_0(x) + \Sigma_{i=1}^{N} \lambda_i f_i(x) + \Sigma_{i=1}^{N} \nu_i h_i(x)$

Finding the dual function

Lagrange function: The penalty function for violating constraints

$L(x, \lambda, \nu) = f_0(x) + \Sigma_{i=1}^{N} \lambda_i f_i(x) + \Sigma_{i=1}^{N} \nu_i h_i(x)$

$f_i(x)$ inequality constraints

Finding the dual function

Lagrange function: The penalty function for violating constraints

$L(x, \lambda, \nu) = f_0(x) + \Sigma_{i=1}^{N} \lambda_i f_i(x) + \Sigma_{i=1}^{N} \nu_i h_i(x)$

$f_i(x)$ inequality constraints

$h_i(x)$ equality constraints

Finding the dual function

Lagrange function: The penalty function for violating constraints

$L(x, \lambda, \nu) = f_0(x) + \Sigma_{i=1}^{N} \lambda_i f_i(x) + \Sigma_{i=1}^{N} \nu_i h_i(x)$

$f_i(x)$ inequality constraints

$h_i(x)$ equality constraints

$L(x, \nu) = f_0(x) + \Sigma_{i=1}^{N} \nu_i h_i(x)$

where $h_i(x) = x_i^2 - 1$

Finding the dual function

Lagrange function: The penalty function for violating constraints

$L(x, \lambda, \nu) = f_0(x) + \Sigma_{i=1}^{N} \lambda_i f_i(x) + \Sigma_{i=1}^{N} \nu_i h_i(x)$

$f_i(x)$ inequality constraints

$h_i(x)$ equality constraints

$L(x, \nu) = f_0(x) + \Sigma_{i=1}^{N} \nu_i h_i(x)$

where $h_i(x) = x_i^2 - 1$

the dual function

$$g(\nu) = inf_x L(x, \nu)$$

Maths

$$g(\nu) = inf_x(x^T Wx + \Sigma_{i=1}^N \nu_i(x_i^2 - 1))$$

$$g(\nu) = inf_x(x^T Wx + \Sigma_{i=1}^N \nu_i x_i^2) - \mathbf{1}^T \nu$$

Now $\Sigma_{i=1}^N \nu_i x_i^2 = x^T diag(\nu) x$

x= $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

diag(x)= $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$

Maths contd.

$$g(\nu) = inf_x(x^T Wx + x^T diag(\nu)x) - \mathbf{1}^T \nu$$

$$g(\nu) = inf_x x^T (W + diag(\nu))x - \mathbf{1}^T \nu$$

Now if $W + diag(\nu)$ is positive semi definite then the minimum value of this expression $x^T(W + diag(\nu))x$ is 0 else it is $-\inf$

Maths contd.

Proof:

Let there be a matrix A which is not Positive semi definite then

Maths contd.

Proof:
Let there be a matrix A which is not Positive semi definite then There is some x such that $x^T Ax \leq a$ where a is some finite negative number

Maths contd.

Proof:
Let there be a matrix A which is not Positive semi definite then
There is some x such that $x^T Ax \leq a$ where a is some finite negative number
Multiply both sides by a arbitrary positive constant k^2

Maths contd.

Proof:
Let there be a matrix A which is not Positive semi definite then
There is some x such that $x^T Ax \leq a$ where a is some finite negative number
Multiply both sides by a arbitrary positive constant k^2

$$(kx)^T A(kx) \leq k^2 a$$

Maths contd.

Proof:
Let there be a matrix A which is not Positive semi definite then
There is some x such that $x^T Ax \leq a$ where a is some finite negative number
Multiply both sides by a arbitrary positive constant k^2

$$(kx)^T A(kx) \leq k^2 a$$

Take k to infinity

Maths contd.

Proof:
Let there be a matrix A which is not Positive semi definite then
There is some x such that $x^T Ax \leq a$ where a is some finite negative number
Multiply both sides by a arbitrary positive constant k^2

$$(kx)^T A(kx) \leq k^2 a$$

Take k to infinity
Hence proved that minimum value of $x^T Ax$ is $-\infty$

Maths contd.

$$g(\nu) = \begin{cases} -\mathbf{1}^T\nu & , W + diag(\nu) \succeq 0 \\ -\infty & , otherwise \end{cases}$$

Using the lower bound property

Lower bound property: $g(\lambda, \nu) \leq p^*$ if $\lambda \succeq 0$ where p^* is the optimal value of the objective function.

Interesting Part

Maximising $g(\nu)$ is same as minimizing $-g(\nu)$.

Interesting Part

Maximising $g(\nu)$ is same as minimizing $-g(\nu)$.

$$-g(\nu) = \begin{cases} \mathbf{1}^T\nu & , W + diag(\nu) \succeq 0 \\ \infty & , otherwise \end{cases}$$

Interesting Part

Maximising $g(\nu)$ is same as minimizing $-g(\nu)$.

$$-g(\nu) = \begin{cases} \mathbf{1}^T\nu & , W + diag(\nu) \succeq 0 \\ \infty & , otherwise \end{cases}$$

Voila

Interesting Part

Maximising $g(\nu)$ is same as minimizing $-g(\nu)$.

$$-g(\nu) = \begin{cases} \mathbf{1}^T\nu & , W + diag(\nu) \succeq 0 \\ \infty & , otherwise \end{cases}$$

Voila

This is a clear convex optimisation problem which needs to be minimised

Interesting Part

Maximising $g(\nu)$ is same as minimizing $-g(\nu)$.

$$-g(\nu) = \begin{cases} \mathbf{1}^T\nu & , W + diag(\nu) \succeq 0 \\ \infty & , otherwise \end{cases}$$

Voila

This is a clear convex optimisation problem which needs to be minimised
Dump it in the solver and get the maximum lower bound

Proof of why its convex

Domain : Linear combination of positive semi definite matrices

Proof of why its convex

Domain : Linear combination of positive semi definite matrices
Method 1: $\nabla(-g(\nu)) = \mathbf{1}$ and $\nabla^2(-g(\nu)) = 0$

Proof of why its convex

Domain : Linear combination of positive semi definite matrices
Method 1: $\nabla(-g(\nu)) = \mathbf{1}$ and $\nabla^2(-g(\nu)) = 0$
Method 2: Sum of r largest components of a vector is a convex function.

Proof of why its convex

Domain : Linear combination of positive semi definite matrices
Method 1: $\nabla(-g(\nu)) = \mathbf{1}$ and $\nabla^2(-g(\nu)) = 0$
Method 2: Sum of r largest components of a vector is a convex function.
Extended Value theorem

Is the dual function always concave

Coincidence ?

Is the dual function always concave

Coincidence ?

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

Is the dual function always concave

Coincidence ?

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

Pointwise infimum of a family of affine functions.

Is the dual function always concave

Coincidence ?

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

Pointwise infimum of a family of affine functions.

For each value of x: Linear combination of λ and ν and a constant

Random lower bound

$$W + diag(\nu) \succeq 0$$

Random lower bound

$$W + diag(\nu) \succeq 0$$

$$\nu = -\lambda_{min}(W)\mathbf{1}$$

Random lower bound

$$W + diag(\nu) \succeq 0$$

$\nu = -\lambda_{min}(W)\mathbf{1}$

$W - \lambda_{min}\mathbf{I} \succeq 0$

where **I** is the Identity matrix

Random lower bound

$$W + diag(\nu) \succeq 0$$

$$\text{WKT } \lambda_{min} = \inf_x \frac{x^T A x}{x^T x}$$

$\nu = -\lambda_{min}(W)\mathbf{1}$

$W - \lambda_{min}\mathbf{I} \succeq 0$

where $\mathbf{I}$ is the Identity matrix

Random lower bound

$$W + diag(\nu) \succeq 0$$

WKT $\lambda_{min} = \inf_{x} \frac{x^T A x}{x^T x}$

To prove $W - \lambda_{min}\mathbf{I} \succeq 0$

$\nu = -\lambda_{min}(W)\mathbf{1}$

$W - \lambda_{min}\mathbf{I} \succeq 0$

where $\mathbf{I}$ is the Identity matrix

Random lower bound

$$W + diag(\nu) \succeq 0$$

$\nu = -\lambda_{min}(W)\mathbf{1}$

$W - \lambda_{min}\mathbf{I} \succeq 0$

where $\mathbf{I}$ is the Identity matrix

WKT $\lambda_{min} = \inf_{x} \frac{x^T A x}{x^T x}$

To prove $W - \lambda_{min}\mathbf{I} \succeq 0$

$x^T(W - \lambda_{min}\mathbf{I})x \geq 0 \ \forall x$

Rearranging we get

$x^T W x \geq \lambda_{min}\|x\|^2$

Random lower bound

$$W + diag(\nu) \succeq 0$$

$\nu = -\lambda_{min}(W)\mathbf{1}$

$W - \lambda_{min}\mathbf{I} \succeq 0$

where $\mathbf{I}$ is the Identity matrix

WKT $\lambda_{min} = \inf_x \frac{x^T Ax}{x^T x}$

To prove $W - \lambda_{min}\mathbf{I} \succeq 0$

$x^T(W - \lambda_{min}\mathbf{I})x \geq 0 \ \forall x$

Rearranging we get

$x^T Wx \geq \lambda_{min}\|x\|^2$

$\lambda_{min} \leq \frac{x^T Wx}{\|x\|^2}$

Random lower bound

$$W + diag(\nu) \succeq 0$$

$\nu = -\lambda_{min}(W)\mathbf{1}$

$W - \lambda_{min}\mathbf{I} \succeq 0$

where $\mathbf{I}$ is the Identity matrix

WKT $\lambda_{min} = \inf_x \frac{x^T A x}{x^T x}$

To prove $W - \lambda_{min}\mathbf{I} \succeq 0$

$x^T(W - \lambda_{min}\mathbf{I})x \geq 0 \ \forall x$

Rearranging we get

$x^T W x \geq \lambda_{min}\|x\|^2$

$\lambda_{min} \leq \frac{x^T W x}{\|x\|^2}$

$$p^* \geq n\lambda_{min}$$

Alternate Approach

New constraints $\sum_{i=1}^{N} x_i^2 = n$.

Alternate Approach

New constraints $\sum_{i=1}^{N} x_i^2 = n$.

$x^T x = n$

Alternate Approach

New constraints $\sum_{i=1}^{N} x_i^2 = n$.

$x^T x = n$

More loose as x_i can take decimal values also.

Alternate Approach

New constraints $\sum_{i=1}^{N} x_i^2 = n$.

$x^T x = n$

More loose as x_i can take decimal values also.

Directly dump into a quadratic minimizer with norm constraint