Assignment 2

EE5121: Convex Optimisation

By

Mohammed Nawfal

ME21B046

Course Instructor:

Prof. Pravin Nair

Indian Institute of Technology, Madras

November, 2025

Contents

Chapter 1

Assignment 2

1.1 LINK TO CVXPY SOLUTIONS

The repository includes a Jupyter Notebook (.ipynb) containing well-documented CVXPY implementations for this assignment, ensuring reproducibility.
Link: https://github.com/mohammednawfal/EE5121-Assignment-2

1.2 QUESTION 1

Let $\{\phi_i\}_{i=1}^n \subset \mathbb{R}^k$ be feature vectors and let $\Phi \in \mathbb{R}^{n\times k}$ collect them by rows. Let $\mu \in \mathbb{R}^k$ be a target moment vector. Define the probability simplex:

$$\Delta_n := \left\{ p \in \mathbb{R}_+^n : \sum_{i=1}^n p_i = 1 \right\}.$$

Consider

$$\max_{p\in\mathbb{R}^n} \quad H(p) := -\sum_{i=1}^n p_i \log p_i$$

subject to

$$\sum_{i=1}^n p_i\phi_i = \mu, \qquad \sum_{i=1}^n p_i = 1, \qquad p_i \geq 0 \ \ \forall i,$$

with the convention $0 \log 0 := 0$.

The objective function $H(p)$ is concave. This follows from the fact that the function $f(x) = x \ln x$ is convex on $x \geq 0$, the sum of convex functions is also convex, and the negative of a convex function is concave. The equality constraints are affine. The inequality constraint is of the form $f_i(p) \geq 0$, where $f_i(p) = p_i$ for all i. $f_i(p) = p_i$ is concave (it is affine, hence it is both convex and concave). Thus, the above problem is a concave optimisation problem. Equivalently, it can be written as a convex optimisation problem by making the following changes:

1. $\max_{p \in \mathbb{R}^n} H(p) = -\min_{p \in \mathbb{R}^n} -H(p)$ Since $H(p)$ is concave, $-H(p)$ is convex.

2. Reverse the inequality constraint into $-p_i \leq 0$ for all i. Since p_i is affine, $-p_i$ is also affine, and hence convex as well.

With the above two changes, the above concave optimisation problem can be converted into an equivalent convex optimisation problem. For subsequent sections, we shall work with the convex problem.

1.2.1 Dual Optimisation Problem

The domain of the objective $-H(p)$ is defined on the interval $[0, \infty)$, with $0\log0 = 0$. This is because the argument of the logarithm must be non-negative. As a result, the domain of the objective already specifies the non-negativity of p. Thus, the inequality constraint $p_i \geq 0$ is redundant and is not required during the Lagrangian formulation.

Using dual variables $\theta \in \mathbb{R}^k$ and $\nu \in \mathbb{R}$, the Lagrangian can be written as,

$$\mathcal{L}(p, \theta, \nu) = \sum_{i=1}^{n} p_i \log p_i \; + \; \theta^\top \Big(\mu - \sum_{i=1}^{n} p_i \phi_i\Big) \; + \; \nu\Big(1 - \sum_{i=1}^{n} p_i\Big). \tag{1.1}$$

For notational purposes, we use p in the Lagrangian to denote p_i for all i. The dual function $g(\theta, \nu)$ defined as,

$$g(\theta, \nu) = \inf_{p \in \mathbb{R}^n_+} \mathcal{L}(p, \theta, \nu)$$

Taking the derivative of the Lagrangian with respect to p_i and equating it to 0,

$$\frac{\partial \mathcal{L}}{\partial p_i} = 1 + \ln p_i - \theta^T \phi_i - \nu$$

$$1 + \ln p_i - \theta^T \phi_i - \nu = 0 \tag{1.2}$$

$$\tilde{p}_i = e^{-1+\nu+\theta^T \phi_i} \tag{1.3}$$

Grouping terms in 1.1 that sum over p_i, we can write,

$$\mathcal{L}(p, \theta, \nu) = \sum_{i=1}^{n} p_i \big(\log p_i - \theta^\top \phi_i - \nu \big) \ + \ \theta^\top \mu \ + \ \nu$$

Using the relations in 1.2 and 1.3, we get,

$$\inf_p \mathcal{L}(p, \theta, \nu) = \sum_{i=1}^{n} -\tilde{p}_i + \theta^T \mu + \nu$$
$$\inf_p \mathcal{L}(p, \theta, \nu) = \sum_{i=1}^{n} -e^{-1+\nu+\theta^T \phi_i} + \theta^T \mu + \nu$$

Thus, the dual function is given by,

$$g(\theta, \nu) = \sum_{i=1}^{n} -e^{-1+\nu+\theta^T \phi_i} + \theta^T \mu + \nu \tag{1.4}$$

Therefore, the dual optimisation problem is given by,

$$- \max_{\theta \in \mathbb{R}^k,\, \nu \in \mathbb{R}} \ \sum_{i=1}^{n} -e^{-1+\nu+\theta^T \phi_i} \ + \ \theta^\top \mu \ + \ \nu \tag{1.5}$$

This can be rewritten as a minimisation problem,

$$\min_{\theta \in \mathbb{R}^k,\, \nu \in \mathbb{R}} \ \sum_{i=1}^{n} e^{-1+\nu+\theta^T \phi_i} \ - \ \theta^\top \mu \ - \ \nu \tag{1.6}$$

1.2.2 **Conditions for Strong Duality**

For strong duality to hold for the above primal-dual problems, we need to check the following:

1. Nature of the optimisation problem - if it is a convex/concave optimisation problem with valid constraints.

2. Slater's condition.

We have already proven that the above problem is a convex optimisation problem with valid constraints. Thus, it is only required to check Slater's condition.

Slater's condition states that for a convex optimisation problem, if $f_i(x)$ is convex and $g_j(x)$ is affine, and if there exists a $x_0 \in \text{relint}(D)$ such that $f_i(x_0) < 0 \;\; \forall i$ and $g_j(x_0) = 0 \;\; \forall j$, where D is the domain of the objective, then strong duality holds. Applying this to the primal problem, we get,

$$\begin{aligned} \text{relint}(D) &= \text{relint}(\mathbb{R}^n_+) = \{\, p \in \mathbb{R}^n : p_i > 0 \; \forall \, i \,\} && (1.7) \\ \sum_{i=1}^n p_i \phi_i &= \mu \rightarrow \Phi^T p = \mu && (1.8) \\ \sum_{i=1}^n p_i &= 1 && (1.9) \\ p_i &> 0 \;\; \forall i && (1.10) \end{aligned}$$

Both 1.7 and 1.10 imply that p_i must be strictly positive for all i. Combining this with 1.8 and 1.9, it implies that μ must be written as a convex combination of $\{\phi_i\}_{i=1}^n$ with strictly positive weights. Hence, μ must lie in the interior of the convex hull of $\{\phi_i\}_{i=1}^n$. To be more precise, it has to be in the relative interior of the convex hull spanned by all $\{\phi_i\}_{i=1}^n$.

$$\mu \in \text{relint}(\text{conv}\{\phi_1, \phi_2, \cdots, \phi_n\})$$

Equivalently, we can say that μ must be a linear combination with strictly positive coefficients of the columns of Φ^T. Under these conditions, Slater's conditions are satisfied, and hence strong duality holds for the problem.

1.2.3 **Gibbs Form of Primal Optimiser**

Let us assume that Slater's condition is satisfied, i.e., strong duality holds. Given that p^*, θ^*, and ν^* (which we assume) are the optimisers of the primal and dual problem, they must satisfy the KKT conditions at optimality. The KKT conditions are:

1. Stationarity: $\nabla_p \mathcal{L}(p^*, \theta^*, \nu^*) = 0$

2. Primal Feasibility: $\sum_{i=1}^n p_i^* \phi_i = \mu$ and $\sum_{i=1}^n p_i^* = 1$. Since p^* is optimal and Slater's condition holds, it is assumed that it satisfies $p^* > 0$.

3. Dual Feasibility: There are no dual feasibility variables, as there are no active inequality constraints.

4. Complementary Slackness: Again, it is not present due to the absence of active inequality constraints.

From the stationarity condition, evaluating it for a single p_i^*, on 1.2,

$$1 + \ln p_i^* - \theta^{*T} \phi_i - \nu^* = 0$$
$$p_i^* = e^{-1+\nu^*+\theta^{*T}\phi_i}$$
$$p_i^* \propto \exp(\theta^{*T}\phi_i)$$

We see that the primal maximiser has the Gibb's form given by $p_i^* \propto \exp(\theta^{*T}\phi_i)$. From the primal feasibility condition, we see that all p_i^* satisfy the equality constraints of the primal problem.

1.2.4 Dual function only in terms of θ

The dual optimisation problem given by 1.5 is unconstrained. To write the dual function only in terms of θ, we shall eliminate ν by writing it in terms of θ. Taking the derivative of $g(\theta, \nu)$ with respect to ν,

$$g(\theta, \nu) = e^{\nu-1} \sum_{i=1}^{n} e^{\theta^\top \phi_i} - \theta^\top \mu - \nu$$
$$\frac{\partial g(\theta, \nu)}{\partial \nu} = e^{\nu-1} \sum_{i=1}^{n} e^{\theta^\top \phi_i} - 1$$

Setting the derivative to zero gives,

$$e^{\nu-1} \sum_{i=1}^{n} e^{\theta^\top \phi_i} = 1 \quad \implies \quad e^{\nu-1} = \frac{1}{\sum_{i=1}^{n} e^{\theta^\top \phi_i}}$$

$$\nu - 1 = -\log\Big(\sum_{i=1}^{n} e^{\theta^\top \phi_i}\Big) \quad \implies \quad \nu^\star = 1 - \log\Big(\sum_{i=1}^{n} e^{\theta^\top \phi_i}\Big) \tag{1.11}$$

Therefore, at optimality, ν is related to θ via 1.11. Substituting $\nu^\star$ back into g gives,

$$g(\theta) = \log\Big(\sum_{i=1}^{n} e^{\theta^\top \phi_i}\Big) - \theta^\top \mu \tag{1.12}$$

This is the dual function exclusively in terms of θ. Substituting 1.11 into 1.2 at optimality, we get,

$$\begin{aligned} 1 + \ln p_i^* - \theta^{*T}\phi_i - \nu^* &= 0 \\ p_i^* &= e^{\nu^* - 1 + \theta^{*T}\phi_i} \\ p_i^* &= \frac{e^{\theta^{*T}\phi_i}}{\sum_{i=1}^{n} e^{\theta^{*T}\phi_i}} \end{aligned} \tag{1.13}$$

Therefore, θ^* governs the primal optimiser p^* by controlling the probability distribution.

1.2.5 **Solving Primal Problem**

To solve the primal problem, which is a concave optimisation problem, we shall pose it as a convex optimisation problem first. Thus, the convex optimisation problem can be written as,

$$\min_{p \in \mathbb{R}^n} \quad -H(p) := \sum_{i=1}^{n} p_i \log p_i$$

such that,

$$\sum_{i=1}^{n} p_i \phi_i = \mu, \qquad \sum_{i=1}^{n} p_i = 1, \qquad -p_i \leq 0 \ \ \forall i,$$

We shall use projected gradient descent (PGD) to solve the problem. The gradient of the objective $-H(p)$ is given by,

$$\nabla_p(-H(p)) = \mathbf{1} + \log p \tag{1.14}$$

Where $\mathbf{1} \in \mathbb{R}^n$ is a vector with all entries equal to 1. $\log p$ refers to applying the log function on all components of p. To avoid negative/zero values in the argument of the gradient, which may arise due to finite precision and floating-point rounding, it will be implemented as,

$$\nabla_p(-H(p)) = \mathbf{1} + \log\big(\max(p, 10^{-12})\big) \tag{1.15}$$

The update step in PGD is given by,

$$\tilde{y}_k = p_k - \alpha_k \nabla_p(-H(p_k))$$
$$p_{k+1} = \mathrm{Proj}_\Omega(\tilde{y}_k)$$

Where $\tilde{y}_k$ is the output after the gradient step, α_k is the step size, p_k is the probability vector at iteration k, and Ω is the constraint set. The projection operation is given by the following optimisation problem,

$$\min_{z\in\mathbb{R}^n} ||z - \tilde{y}_k||_2^2$$

such that,

$$\sum_{i=1}^{n} z_i\phi_i = \mu, \qquad \sum_{i=1}^{n} z_i = 1, \qquad -z_i \leq 0 \ \ \forall i,$$

The objective function can be rewritten as,

$$\begin{aligned} \|z - \tilde{y}_k\|_2^2 &= (z - \tilde{y}_k)^\top (z - \tilde{y}_k) \\ &= z^\top z - z^\top \tilde{y}_k - \tilde{y}_k^\top z + \tilde{y}_k^\top \tilde{y}_k \\ &= z^\top z - 2\,\tilde{y}_k^\top z + \|\tilde{y}_k\|_2^2 \end{aligned}$$

The above objective has the standard quadratic form $z^T Pz + q^T z + t$, where $P = I$, the identity matrix of size n, $q = -2\tilde{y}_k$, and $t = ||\tilde{y}_k||_2^2$. Thus, the projection problem is a quadratic programming problem with convex inequality constraints and affine equality constraints. This can be solved using CVXPY by posing it as a DCP problem.

The initial guess p_0 is obtained by randomly sampling from the interval $[0, 1]$ for all components, followed by normalising it by its sum so that it is a valid probability. The PGD algorithm is implemented as follows:

Algorithm 1 Projected Gradient Descent

```
Inputs: $\Phi \in \mathbb{R}^{n\times k}$, $\mu \in \mathbb{R}^k$, $\{\alpha_k\}_{i=1}^K$, K
Initialize $p_0 \in \Delta_n$
for $k = 1, 2, \ldots, K$ do
    Compute gradient according to 1.15
    Gradient step:  $\tilde{y}_k = p_k - \alpha_k \nabla(-H(p_k))$
    Projection: $p_{k+1} = \arg\min_{z\in\Omega} \|z - \tilde{y}_k\|_2^2$ using CVXPY
    Compute $-H(p_{k+1})$, $||\Phi^T p_{k+1} - \mu||_2$, $|\mathbf{1}^T p_{k+1} - 1|$ and store in a list
end for
return $p^* = p_K$, $\{-H(p_k)\}_{i=1}^K$, $\{||\Phi^T p_k - \mu||_2\}_{i=1}^K$, $\{|\mathbf{1}^T p_k - 1|\}_{i=1}^K$
```

To solve the convex primal problem, we have chosen a constant step size of **0.001** and $\mathbf{K} = \mathbf{1000}$.

$H(p_k)$ **vs k**

To obtain the objective value $H(p_k)$ of the actual concave problem, we just take the negative of the objective value of the convex problem at each iteration. The variation of $H(p_k)$ with k has been plotted below. We obtain the following results:

- $p^* = [0.1032, 0.1290, 0.1371, 0.1340, 0.1126, 0.0879, 0.0732, 0.0747, 0.0663, 0.0816]$
- $H(p^*) = 2.27005$.

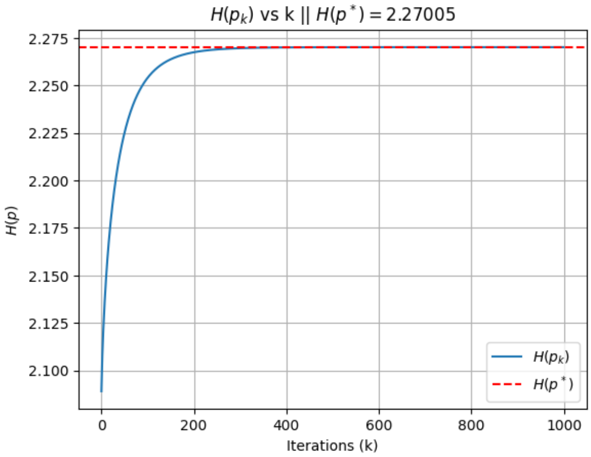

Figure 1.1: $H(p_k)$ vs k

Feasibility residuals vs k

We obtain the following residuals for the optimiser p^*:

- $||\Phi^T p^* - \mu||_2 = 0.0$
- $|\mathbf{1}^T p^* - 1| = 1.11 \times 10^{-16}$

The simplex residual is not exactly zero, but a very small number due to finite precision and floating-point rounding. The same reason is why the feasibility residuals do not monotonically reduce with iterations, but oscillate around very small values close to zero. Their variation with k has been plotted below.

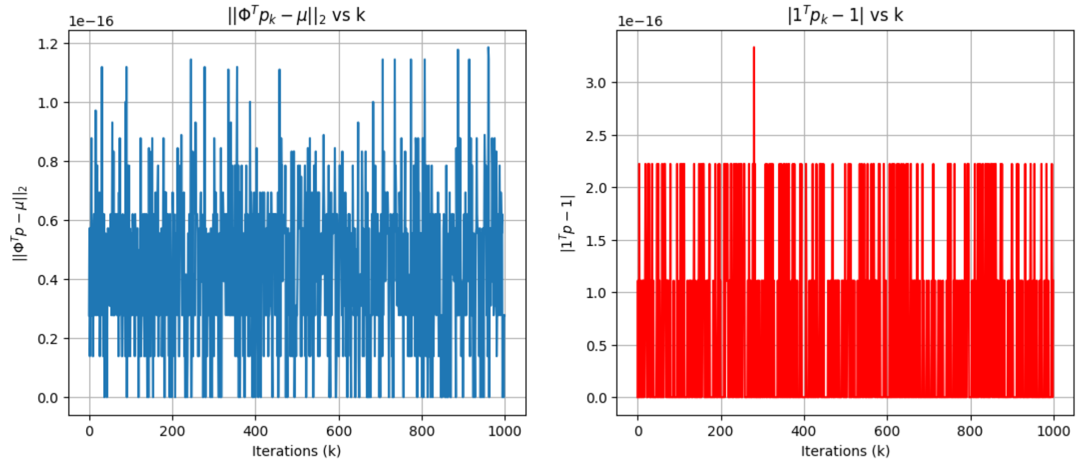

Figure 1.2: Feasibility Residuals vs k

1.2.6 Solving Dual Problem

The dual optimisation problem is given by 1.5. It is unconstrained, and can be directly solved by CVXPY by posing it as a DCP problem. While posing it as a DCP problem, the dual function has to be rewritten as,

$$
\begin{aligned}
g(\theta, \nu) &= \sum_{i=1}^{n} \exp(\nu - 1 + \theta^\top \phi_i) \;-\; \theta^\top \mu \;-\; \nu \\
&= \exp(\nu - 1) \sum_{i=1}^{n} \exp(\theta^\top \phi_i) \;-\; \theta^\top \mu \;-\; \nu \\
&= \exp\left(\nu - 1 + \ln\left(\sum_{i=1}^{n} \exp(\theta^\top \phi_i) \right) \right) \;-\; \theta^\top \mu \;-\; \nu.
\end{aligned}
$$

The convexity of the dual function remains unchanged with this manipulation. This dual objective follows DCP, and can be solved directly using CVXPY. We obtain the following results:

- $\theta^* = [0.0625, 0.3272]$
- $\nu^* = -1.3483$
- $g(\theta^*, \nu^*) = 2.27005$
- $\tilde{p}$, computed using the θ^* and ν^* = [0.1032, 0.1290, 0.1371, 0.1340, 0.1126, 0.0879, 0.0732, 0.0747, 0.0663, 0.0816]
- $||p^* - \tilde{p}||_\infty = 5.68 \times 10^{-5}$
- $|H(p^*) - g(\theta^*, \nu^*)| = 1.49 \times 10^{-8}$

The ∞-norm is not exactly zero, but a very small number, as well as the duality gap. This is again due to finite precision, floating-point rounding, and tolerances used in CVXPY solvers. The small values indicate that, ignoring the above effects, strong duality holds for the above primal-dual problems.

1.3 QUESTION 2

Consider binary labels $y_i \in -1, +1$ and features $x_i \in \mathbb{R}^d$ for $i = 1, 2, \cdots, n$. Consider the logistic regression problem,

$$\min_{w \in \mathbb{R}^d} L(w) = \sum_{i=1}^{n} \log(1 + \exp(-y_i x_i^T w)) \tag{1.16}$$

1.3.1 Correspondence to Probabilistic Binary Classifier

Let us consider two cases and observe the function in the summation.

- When $y_i = +1$, for the function to be minimised, $x^T w$ must assume a positive value, i.e., $x_i^T w \geq 0$.

- When $y_i = -1$, for the function to be minimised, $x^T w$ must assumed a negative value, i.e., $x_i^T w \leq 0$.

1.3.2 Solving the Minimisation Problem

The above minimisation problem can be shown to be a convex minimisation problem.

1. For any y_i, x_i, the function $f(w) = -y_i x_i^T w$ is affine, and hence convex.

2. Using composition, since $f(w)$ is convex, and e^x is convex and non-decreasing on its domain, $e^{f(w)}$ is also convex.

3. Let us consider the function $g(t) = \ln(1 + e^t)$. first and second-order derivatives of $g(t)$ are,

$$g'(t) = \frac{e^t}{(1 + e^t)} > 0 \quad \forall t \tag{1.17}$$

$$g''(t) = \frac{e^t}{(1 + e^t)^2} > 0 \quad \forall t \tag{1.18}$$

   From 1.17 and 1.18, we can say that $g(t)$ is non-decreasing and is convex. Using composition again, $g(f(w))$ is also convex.

4. Since the sum of convex functions is also convex, $L(w)$ is also a convex function. Thus, the above minimisation problem is a convex minimisation problem.

We shall solve the logistic regression problem using gradient descent. The gradient of the objective is given by,

$$\nabla_w L(w) = \sum_{i=1}^{n} \frac{-\exp(-y_i x_i^T w) y_i x_i}{1 + \exp(-y_i x_i^T w)} \tag{1.19}$$

Given the small size of the dataset ($n = 400$), we will set the number of iterations to $K = 10,000$. The step size used is 0.001. The following results are obtained after gradient descent. Note that these results are calculated for w_K, the final solution vector obtained, which is not the same as w^*. This is explained in the next subpart.

- $w_K = [-13.3987, 0.8813]$

- $L(w_K) = 0.5078$

- $||w_K||_2 = 13.4132$

The variation of $L(w_k)$ and $||w_k||_2$ with k has been plotted below. Both $L(w_k)$ and $||w_k||_2$ are plotted on a log-scale.

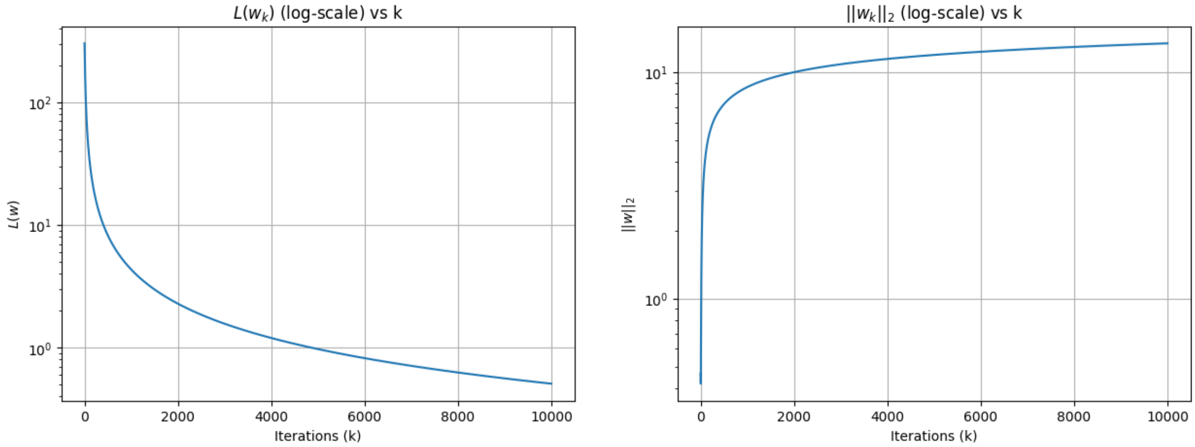

Figure 1.3: $L(w_k)$ (log-scale) and $||w_k||_2$ (log-scale) vs k

1.3.3 Reason for Failure to Achieve Finite Optimiser

We observe from the above plots that the norm of the optimiser continues to increase (and the objective decreases). Had the number of iterations been higher, the norm would have increased steadily, resulting in an even smaller objective without arriving at a steady finite optimiser. This is due to the following:

1. $x_i^T w = ||x_i|| ||w|| \cos\theta_i$. For a fixed label y_i, and x_i, the value of $x_i^T w$ is determined by $||w||$ and its direction θ_i.

2. If the two classes are linearly separable, i.e., there exists a w such that $y_i x_i^T w > 0$ for all i. Then cw, where $c > 0$, also satisfies the above separability criteria for all i.

3. A visualisation of the given dataset confirms the fact that the dataset is indeed linearly separable.

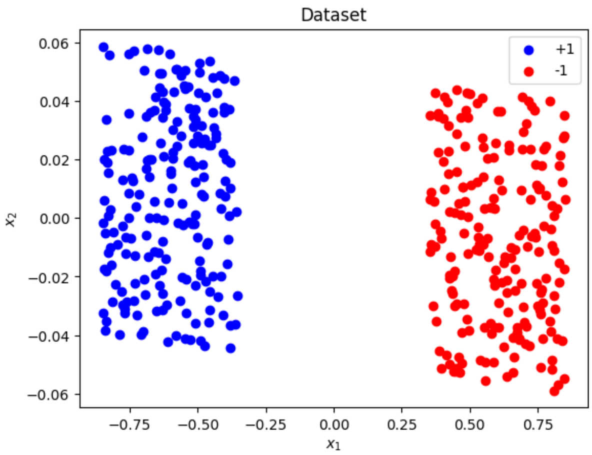

Figure 1.4: Dataset Visualisation

4. Thus, by increasing c, it would reduce the value of $L(w)$ since $\ln(1+\exp(-y_i x_i^T cw))$ decreases with an increase in c.

5. Since there is no constraint on $||w||_2$, the gradient descent algorithm produces a w with increasing magnitude with each step i.e., c increases. In the limit $K \to \infty$, $c \to \infty$, $||w||_2 \to \infty$, and $L(w) \to 0$.

1.3.4 Alternative Formulation

From the above reasoning, it is evident that $||w||_2$ needs to be constrained in some way. This can be achieved by introducing a regularising term into the original minimisation problem. Consider the regularised version of the logistic regression problem,

$$\min_{w \in \mathbb{R}^d} \tilde{L}(w) = \sum_{i=1}^{n} \log(1 + \exp(-y_i x_i^T w)) + C||w||_2^2 \tag{1.20}$$

Where C is some positive constant. A larger C would penalise $||w||_2$ more heavily. The above modified objective is convex, since the original objective is convex, and $C||w||_2^2$ is a convex function, and the sum of two convex functions is convex. The gradient of the modified objective is now,

$$\nabla_w \tilde{L}(w) = \sum_{i=1}^{n} \frac{-\exp(-y_i x_i^T w) y_i x_i}{1 + \exp(-y_i x_i^T w)} + 2Cw \tag{1.21}$$

We shall solve the modified problem using gradient descent. The number of iterations is set to $K = 10,000$. The step size used is 0.001. The values of C are chosen to be $[0.1, 0.5, 1, 2, 5]$ to evaluate the effect of C on w^* and $\tilde{L}(w^*)$. The variation of $\tilde{L}(w_k)$ and $||w_k||_2$ with k has been plotted below for various values of C. Both $\tilde{L}(w_k)$ and $||w_k||_2$ are plotted on a log-scale.

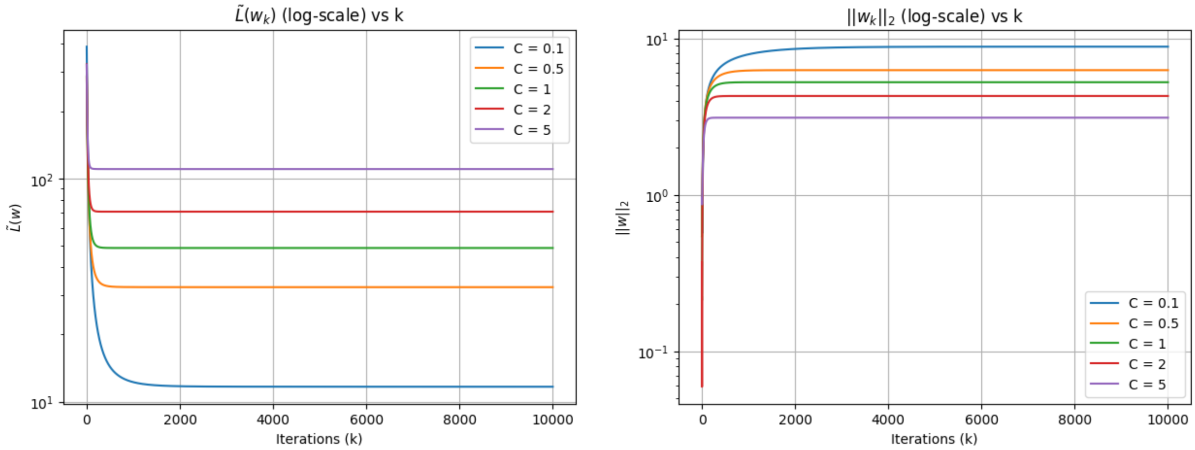

Figure 1.5: $\tilde{L}(w_k)$ (log-scale) and $||w_k||_2$ (log-scale) vs k

From the above plots, we make the following inferences,

1. Introducing the $C||w||_2^2$ term helped in preventing the blow up of $||w||_2$. $||w||_2$ does not blow up with each step, and instead converges to a finite value after a certain number of iterations.

2. Likewise, $\tilde{L}(w)$ also converges to some finite value after a certain number of iterations.

3. A finite optimiser w^* is obtained with this alternate formulation, which is evident from the convergence of both plots for various values of C.

4. A larger C results in a smaller norm for w^*, and a higher optimal value for $\tilde{L}(w^*)$. This makes sense since C indicates the penalty on the magnitude of w.

The optimiser w^*, $||w^*||_2$, and $\tilde{L}(w^*)$ for various values of C have been tabulated below.

C	Optimiser w^*	$\tilde{L}(w^*)$	$\|w^*\|_2$
0.1	$[-8.843\ \ 0.182]$	11.671	8.845
0.5	$[-6.261\ \ 0.082]$	32.583	6.261
1.0	$[-5.241\ \ 0.070]$	48.803	5.242
2.0	$[-4.281\ \ 0.057]$	70.979	4.281
5.0	$[-3.110\ \ 0.042]$	110.134	3.110

Table 1.1: Effect of regularisation parameter C on w^*, $\tilde{L}(w^*)$, and $||w^*||_2$

1.4 QUESTION 3

Let $G = (V, E)$ be a simple undirected graph with $|V| = n$. The vector-colouring SDP form yields the following optimisation problem.

$$\begin{aligned} \min_{G,\rho} \quad & \rho \\ \text{s.t.} \quad & G \succeq 0, \\ & G_{vv} = 1 \quad \forall v \in V, \\ & G_{uv} \leq \rho \quad \forall (u, v) \in E, \end{aligned}$$

where $G \in \mathbb{S}^n$ is a Gram matrix and $\rho \in \mathbb{R}$ bounds edge-wise inner products. The K-color vector feasibility threshold is

$$\rho \leq -\frac{1}{K-1}.$$

1.4.1 Dual Problem Derivation

Let G, ρ be the primal variables. Let us introduce multipliers $\lambda \in \mathbb{R}^n$ for $G_{vv} = 1$, $\alpha_{uv} \geq 0$ for all $(u, v) \in E$ for the constraint $G_{u,v} - \rho \leq 0$, and a dual slack $S \succeq 0$. For notational purposes, let α in the argument of $\mathcal{L}$ denote all α_{uv}.The Lagrangian is written as:

$$\mathcal{L}(G, \rho, \lambda, \alpha, S) = \rho + \sum_{v \in V} \lambda_v (G_{vv} - 1) + \sum_{(u,v) \in E} \alpha_{uv} (G_{uv} - \rho) - \text{Tr}(S^T G) \tag{1.22}$$

$\text{Tr}(S^T G)$ can be written as,

$$\text{Tr}(S^T G) =< S, G > \tag{1.23}$$

Now, $\sum_{v \in V} \lambda_v G_{vv}$ can be written as,

$$\sum_{v \in V} \lambda_v G_{vv} =< \operatorname{diag}(\lambda), G > \tag{1.24}$$

Likewise, $\sum_{(u,v) \in E} \alpha_{uv} G_{uv}$ can be written as,

$$\sum_{(u,v) \in E} \alpha_{uv} G_{uv} =< A, G > \tag{1.25}$$

We define the matrix $A \in \mathbb{S}^n$ by

$$A(\alpha)_{uv} \;=\; \begin{cases} \frac{1}{2}\,\alpha_{uv} & \text{if } (u,v) \in E,\ u \neq v, \\ \\ 0 & \text{otherwise.} \end{cases}$$

Combining 1.23, 1.24, 1.25, with 1.22, we get

$$\mathcal{L}(G, \rho, \lambda, \alpha, S) = \rho - \sum_{v \in V} \lambda_v - \sum_{(u,v) \in E} \alpha_{uv} \rho + < \operatorname{Diag}(\lambda), G > + < A, G > - < S, G >$$

$$\mathcal{L}(G, \rho, \lambda, \alpha, S) = \rho - \sum_{v \in V} \lambda_v - \sum_{(u,v) \in E} \alpha_{uv} \rho + < \operatorname{Diag}(\lambda) + A - S, G >$$

The dual function $g(\lambda, \alpha)$ is given by $g(\lambda, \alpha) = \inf_{G,\rho} \mathcal{L}(G, \rho, \lambda, \alpha, S)$. Taking the derivative with respect to G and ρ and setting it to 0, we get,

$$\frac{\partial \mathcal{L}}{\partial \rho} = 1 - \sum_{(u,v) \in E} \alpha_{uv}$$

$$1 - \sum_{(u,v) \in E} \alpha_{uv} = 0 \rightarrow \sum_{(u,v) \in E} \alpha_{uv} = 1 \quad \forall (u,v) \in E$$

$$\frac{\partial \mathcal{L}}{\partial G} = \operatorname{Diag}(\lambda) + A - S$$

$$\operatorname{Diag}(\lambda) + A - S = 0 \rightarrow S = A + \operatorname{Diag}(\lambda)$$

Substituting the above expressions in the Lagrangian, we get the dual function as,

$$g(\lambda, \alpha) = -\sum_{v\in V} \lambda_v \tag{1.26}$$

Therefore, the dual optimisation function can be written as,

$$\begin{gathered}
\max_{S,\alpha,\lambda\in\mathbb{R}^n} -\sum_{v\in V} \lambda_v \\
\text{s.t.} \quad \alpha_{uv} \geq 0 \quad \forall (u,v) \in E \\
S \succeq 0 \\
\sum_{(u,v)\in E} \alpha_{uv} = 1 \\
S = A + \mathrm{Diag}(\lambda)
\end{gathered}$$

With A constructed using α_{uv} as mentioned above. The maximisation can be equivalently converted into a minimsation by writing it as,

$$\max_{S,\alpha,\lambda\in\mathbb{R}^n} -\sum_{v\in V} \lambda_v = -\min_{S,\alpha,\lambda\in\mathbb{R}^n} \sum_{v\in V} \lambda_v \tag{1.27}$$

1.4.2 Solving Dual and Primal Problems

We are given two graphs:

1. $V = \{1, 2, 3, 4\}$, $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$.
2. $V = \{1, 2, 3, 4, 5\}$, $E = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 1)\}$.

For each of the graphs, the primal and dual problems are solved using CVXPY. Both problems are solved as minimisation problems. The dual optimal value d^*, the primal optimal value p^*, primal optimiser ρ^*, and the duality gap $|d^* - p^*|$, have been tabulated below.

Graph	**Primal Opt. Value**	**Dual Opt. Value**	**Duality Gap**
1	-0.333333	-0.333337	4.15×10^{-6}
2	-0.809017	-0.809017	3.40×10^{-7}

Table 1.2: Primal–Dual results and duality gaps

We see that the duality gaps are of the order $10^{-6} - 10^{-7}$, indicating that strong duality holds for the problems. It is not exactly zero due to floating-point rounding and finite precision.

1.4.3 Feasibility of 3-colourability

Part (a)

If the dual optimum $d^* > -\frac{1}{2}$, then by weak duality we have,

$$-\frac{1}{2} < d^* \leq p^* = \rho^* \tag{1.28}$$

However, the primal problem requires $\rho \leq -\frac{1}{2}$ for 3-colourability. From 1.28 it is clear that ρ^* cannot satisfy $\rho^* \leq -\frac{1}{2}$. Hence, the dual solution serves as a certificate for the infeasibility of vector 3-colouring.

Part (b)

For graph 1, we have $d^* = -0.333 > -0.5$. Using the above result, the graph is not 3-colourable. It is, in fact, 4-colourable if we solve for K using p^*.

For graph 2, we have $d^* = -0.809 < -0.5$. From the above result, the graph is 3-colourable. Solving for K, we see that $K \geq 2.236$. Since K must be an integer, we get $K \geq 3$.