COMP0083 Convex Optimisation Assignment

Jan 9, 2023

Part 1: Questions with multiple answers

1.1

The function for (a):

$$\max\{ax+b, x^4-5, e^{x^2}\}$$

is a convex function.

1.2

For the function:

$$f(x) = \begin{cases} -x & \text{if } x \in]-1,0] \\ x^2 & \text{if } x \geq 0 \end{cases}$$

the sub-differential is:

$$\partial f(x) = \begin{cases} -1 & \text{if } x \in]-1,0[\\ [0,1] & \text{if } x = 0 \\ 2x & \text{if } x > 0 \end{cases}$$

corresponding to Figure (a).

1.3

For a function:

$$f(x) = \langle Ax, x\rangle + \langle x, b\rangle + c$$

where A is a square matrix not necessarily symmetric, the gradient is (a):

$$\nabla f(x) = A^* x + Ax + b$$

1.4

The Fenchel conjugate of $f(x) = g(2x)$ is (a):

$$f^*(u) = g^*(u/2)$$

1.5

The solution to the dual problem is (c):

$$\bar{u} = (\mathbf{K} + \lambda n \mathbf{Id})^{-1} y$$

Part 2: Theory on convex analysis and optimization

2.1

2.1.1

Given :

$$f(x) = \begin{cases} +\infty & \text{if } x \leq 0 \\ -\log x & \quad \text{if } x > 0 \end{cases}$$

The Fenchel conjugate is defined:

$$f^*(u) = \sup_{x \in \mathcal{X}} \{ \langle x, u \rangle + \log x \}$$

To find the supremum, we can take the partial derivative with respect to x, set to zero, and solve for x:

$$\frac{\partial}{\partial x}(ux + logx) = u + \frac{1}{x} = 0$$

Thus, the supremum above is solved when $x = \frac{-1}{u}$:

$$f^*(u) = u\left(\frac{-1}{u}\right) + \log\left(\frac{-1}{u}\right)$$

Simplifying, we have the Fenchel conjugate:

$$f^*(u) = -(1 + \log u)$$

2.1.2

Given:

$$f(x) = x^2$$

The Fenchel conjugate is defined:

$$f^*(u) = \sup_{x \in \mathcal{X}} \{ \langle x, u \rangle - x^2 \}$$

We can compute the partial derivative:

$$\frac{\partial}{\partial x}(ux - x^2) = u - 2x = 0$$

Thus, the supremum is solved when $x = \frac{u}{2}$:

$$f^*(u) = u\left(\frac{u}{2}\right) - \left(\frac{u}{2}\right)^2$$

Simplifying, we have the Fenchel conjugate:

$$f^*(u) = \frac{u^2}{4}$$

2.1.3

Given:

$$f(x) = i_{[0,1]}$$

The Fenchel conjugate is defined:

$$f^*(u) = \sup_{x \in \mathcal{X}} \{\langle x, u\rangle - i_{[0,1]}\}$$

Thus,

$$f^*(u) = \sup_{x \in [0,1]} \{\langle x, u\rangle\}$$

We can see the Fenchel conjugate is:

$$f^*(u) = \max(0, u)$$

2.2.1

Given f a proper convex function, to prove by induction Jensen's inequality:

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i)$$

for all $x_1, ...x_n \in \mathcal{X}$ and for all $\lambda_1, ..., \lambda_n \in \mathbb{R}_+$ with $\sum_{i=1}^{n} \lambda_i = 1$, we start with the definition of convexity which states:

$$f(\lambda_1 x_1 + \lambda_2 x_2) \leq \lambda_1 f(x_1) + \lambda_2 f(x_2)$$

for all $x1, x2 \in \mathcal{X}$ and $\lambda_1, \lambda_2 \in \mathbb{R}_+$ with $\lambda_1 + \lambda_2 = 1$. In this base case $n = 2$.
Our inductive step will prove that the inequality continues to hold for $n + 1$:

$$f\left(\sum_{i=1}^{n+1} \lambda_i x_i\right) = f\left(\sum_{i=1}^{n} \lambda_i x_i + \lambda_{n+1} x_{n+1}\right)$$

We can insert the term $(1 - \lambda_{n+1})$:

$$f\left(\sum_{i=1}^{n+1} \lambda_i x_i\right) = f\left((1 - \lambda_{n+1}) \sum_{i=1}^{n} \frac{\lambda_i}{1 - \lambda_{n+1}} x_i + \lambda_{n+1} x_{n+1}\right)$$

If we define $\bar{x} = \sum_{i=1}^{n} \frac{\lambda_i}{1-\lambda_{n+1}} x_i$ and we are back to our $n = 2$ base case, so we know from convexity:

$$f\left((1-\lambda_{n+1})\bar{x} + \lambda_{n+1}x_{n+1}\right) \leq (1-\lambda_{n+1})f\left(\bar{x}\right) + \lambda_{n+1}f\left(x_{n+1}\right)$$

Rewriting this,

$$f\left(\sum_{i=1}^{n+1} \lambda_i x_i\right) \leq (1-\lambda_{n+1})f\left(\sum_{i=1}^{n} \frac{\lambda_i}{1-\lambda_{n+1}} x_i\right) + \lambda_{n+1}f\left(x_{n+1}\right)$$

We know that the first term on the right hand side:

$$f\left(\sum_{i=1}^{n} \frac{\lambda_i}{1-\lambda_{n+1}} x_i\right) = \frac{1}{1-\lambda_{n+1}} f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \frac{1}{1-\lambda_{n+1}} \sum_{i=1}^{n} \lambda_i f\left(x_i\right) = \sum_{i=1}^{n} \frac{\lambda_i}{1-\lambda_{n+1}} f\left(x_i\right)$$

Thus can upper bound our previous inequality:

$$f\left(\sum_{i=1}^{n+1} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \frac{\lambda_i}{1-\lambda_{n+1}} f\left(x_i\right) + \lambda_{n+1}f\left(x_{n+1}\right) = \sum_{i=1}^{n+1} \frac{\lambda_i}{1-\lambda_{n+1}} f\left(x_i\right)$$

proving our inductive step and Jensen's inequality as required. □

2.2.2

The characterisation of convexity states:

$$\text{f is convex} \leftrightarrow \langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0$$

$\forall x, y \in dom f$.
For $f(x) = -\log(x)$, $\nabla f(x) = \frac{-1}{x}$:

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle = \langle \frac{1}{y} - \frac{1}{x}, x - y \rangle$$

If $x > y$ we have the following:

$$\frac{1}{y} > \frac{1}{x}$$
$$x - y > 0$$
$$\frac{1}{y} - \frac{1}{x} > 0$$

and

$$\langle \frac{1}{y} - \frac{1}{x}, x - y \rangle > 0$$

If $x < y$ we have the following:

$$\frac{1}{y} < \frac{1}{x}$$
$$x - y < 0$$
$$\frac{1}{y} - \frac{1}{x} < 0$$

and

$$\langle \frac{1}{y} - \frac{1}{x}, x - y \rangle > 0$$

If $x = y$ we have the following:

$$\frac{1}{y} = \frac{1}{x}$$
$$x - y = 0$$
$$\frac{1}{y} - \frac{1}{x} = 0$$

and

$$\langle \frac{1}{y} - \frac{1}{x}, x - y \rangle = 0$$

Thus, $\forall x, y \in dom f$, $\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0$ so $f(x) = -\log(x)$ is convex. □

2.2.3

Given Jensen's inequality in 2.2.1 for a convex function $f(x)$:

$$f\left(\sum_{i=1}^{n} \lambda_i x_i \right) \leq \sum_{i=1}^{n} \lambda_i f(x_i)$$

and having proved in 2.2.2 that $f(x) = -\log(x)$ is convex, we can write:

$$-\log \left(\sum_{i=1}^{n} \lambda_i x_i \right) \leq -\sum_{i=1}^{n} \lambda_i \log(x_i)$$

Rearranging:

$$\sum_{i=1}^{n} \lambda_i x_i \geq \exp \left(\sum_{i=1}^{n} \log(x_i^{\lambda_i}) \right)$$

Choosing $\lambda_i = \frac{1}{n}$:

$$\frac{1}{n} \sum_{i=1}^{n} x_i \geq \exp \left(\sum_{i=1}^{n} \log(x_i^{\frac{1}{n}}) \right)$$

The sum of logarithms is the logarithm of the products:

$$\frac{1}{n}\sum_{i=1}^{n} x_i \geq \exp\left(\log((x_1 \cdots x_n)^{\frac{1}{n}})\right)$$

Thus we get our inequality:

$$\frac{1}{n}\sum_{i=1}^{n} x_i \geq \sqrt[n]{x_1 \cdots x_n}$$

□

2.3

Given a polytope $C = co(a_1, ..., a_m)$ in X, we know that any point $x \in C$ can be expressed:

$$x = \sum_{i=1}^{m} \lambda_i a_i$$

where $\sum_{i=1}^{m} \lambda_i = 1$. Knowing that f is a convex function on C, we can write Jensen's inequality for :

$$f\left(\sum_{i=1}^{m} \lambda_i a_i\right) \leq \sum_{i=1}^{m} \lambda_i f(a_i)$$

where $\sum_{i=1}^{m} \lambda_i = 1$.

Because $\sum_{i=1}^{m} \lambda_i f(a_i)$ is a weighted average of $f(a_i)$'s we know that:

$$\sum_{i=1}^{m} \lambda_i f(a_i) \leq \max_{\{a_i\}_{i=1}^{m}} f(a_i)$$

the weighted average will always be less than or equal to the maximum $f(a_i)$ value. Thus, we know that:

$$f\left(\sum_{i=1}^{m} \lambda_i a_i\right) \leq \max_{\{a_i\}_{i=1}^{m}} f(a_i)$$

So the maximum of the convex function f on C is attained at one of the vertices $a_1, ..., a_m$. □

2.4

To prove that the function $f(x, y) = \|x - 2y\|^2$ is convex, we will show that the Hessian is a positive semi-definite matrix. The Hessian is defined as:

$$\nabla^2 f(x, y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial^2 x} & \frac{\partial f(x,y)}{\partial y \partial x} \\ \frac{\partial f(x,y)}{\partial x \partial y} & \frac{\partial f(x,y)}{\partial^2 y} \end{bmatrix}$$

Calculating each term:

$$\nabla^2 f(x, y) = \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix}$$

we see that the Hessian is positive semi-definite and so the function is jointly convex.

2.5

The conditions for the existence of minimizers is that f is closed and coercive. The conditions for the uniqueness of minimizers is that f is strictly convex. Thus for the existence and uniqueness of minimizers for a convex function f, we require that f is closed, coercive, and strictly convex.

2.6

We are considering the optimisation problem:

$$\min_{\|\mathbf{Ax}-\mathbf{b}\|_\infty \leq \epsilon} \frac{1}{2}\|\mathbf{x}\|^2$$

where $\epsilon > 0$, $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{b} \in \mathbb{R}^{n\times 1}$, and $\mathbf{x} \in \mathbb{R}^{d\times 1}$.

2.6.1

To compute the dual problem, we begin by reformulating the problem as:

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{Ax})$$

where $f(x) = \frac{1}{2}\|\mathbf{x}\|^2$ and $g(\mathbf{Ax}) = i_{[0,1]}\left(\frac{1}{\epsilon}\|\mathbf{Ax}-\mathbf{b}\|_\infty\right)$.
Thus we have our primal problem:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}\|^2 + i_{[0,1]}\left(\frac{1}{\epsilon}\|\mathbf{Ax}-\mathbf{b}\|_\infty\right)$$

We know from the Fenchel-Rockafellar duality theory that the dual problem is formulated as:

$$\min_{\mathbf{u}} g^*(\mathbf{u}) + f^*(-\mathbf{A}^*\mathbf{u})$$

where $\mathbf{u} \in \mathbb{R}^{n\times 1}$, $\mathbf{A}^* \in \mathbb{R}^{d\times n}$ the transpose of $\mathbf{A}$, and g^* and f^* are the Fenchel conjugates of g and f respectively.
The Fenchel conjugates are:

$$g^*(\mathbf{Ax}) = i^*_{[0,1]}\left(\frac{1}{\epsilon}\|\mathbf{Ax}-\mathbf{b}\|_\infty\right) = \frac{1}{\epsilon}\|\mathbf{Ax}-\mathbf{b}\|_1$$

$$f^*(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$$

We can substitute:

$$\min_{\mathbf{u}} \frac{1}{\epsilon}\|\mathbf{u}-\mathbf{b}\|_1 + \frac{1}{2}\|-\mathbf{A}^*\mathbf{u}\|^2$$

Simplifying, we have our dual problem:

$$\min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^*\mathbf{AA}^*\mathbf{u} + \frac{1}{\epsilon}\|\mathbf{u}-\mathbf{b}\|_1$$

2.6.2

To determine if strong duality holds, we use the qualification condition:

$$\mathbf{0}_n \in int\big(dom(g) - \mathbf{A}dom(f)\big)$$

that $\mathbf{0}_n = \mathbf{0} \in \mathbb{R}^{n\times 1}$ is in the interior of the intersection of $dom(g)$, the domain of g, with $\mathbf{A}dom(f)$, $\mathbf{A}$ applied to the domain of f.

We know that for $f(x) = \frac{1}{2}\|\mathbf{x}\|^2$:

$$\mathbf{0}_d \in int\big(dom(f)\big)$$

where $\mathbf{0}_d = \mathbf{0} \in \mathbb{R}^{d\times 1}$. Thus, knowing that $\mathbf{A} \in \mathbb{R}^{n\times d}$ is a linear operator:

$$\mathbf{0}_n \in int\big(\mathbf{A}dom(f)\big)$$

Moreover, for $g(\mathbf{u}) = i_{[0,1]}\left(\frac{1}{\epsilon}\|\mathbf{u} - \mathbf{b}\|_\infty\right)$:

$$\mathbf{0}_n \in dom(g)$$

if $\|\mathbf{b}\|_\infty \leq \epsilon$. Therefore:

$$\mathbf{0}_n \in int\big(dom(g)\big)$$

if $\|\mathbf{b}\|_\infty < \epsilon$.

Under the condition that $\|\mathbf{b}\|_\infty < \epsilon$, strong duality holds. This is because it will be the case that $\mathbf{0}_n \in int\big(\mathbf{A}dom(f)\big)$ and $\mathbf{0}_n \in int\big(dom(g)\big)$ and so the same will hold for their intersection, $\mathbf{0}_n \in int\big(dom(g) - \mathbf{A}dom(f)\big)$ as required.

2.6.3

The KKT conditions:

$$\mathbf{x} \in \partial f^*(-\mathbf{A}^*\mathbf{u}) \text{ and } \mathbf{A}\mathbf{x} \in \partial g^*(\mathbf{u})$$

where ∂f^* and ∂g^* are the subgradients of f^* and g^* respectively.

We know that:

$$\partial f^*(-\mathbf{A}^*\mathbf{u}) = \partial\frac{1}{2}\|-\mathbf{A}^*\mathbf{u}\|^2 = -\mathbf{A}^*\mathbf{u}$$

Moreover:

$$\partial g^*(\mathbf{u}) = \partial\frac{1}{\epsilon}\|\mathbf{u} - \mathbf{b}\|_1$$

and so

$$(\partial\|\mathbf{u} - \mathbf{b}\|_1)_i = \begin{cases} \frac{1}{\epsilon}, & \text{if } |u_i - b_i| > 0 \\ -\frac{1}{\epsilon}, & \text{if } |u_i - b_i| < 0 \\ [-\frac{1}{\epsilon}, \frac{1}{\epsilon}], & \text{if } |u_i - b| = 0 \end{cases}$$

where $(\partial\|\mathbf{u} - \mathbf{b}\|_1)_i$ is the i^{th} element of $\partial\|\mathbf{u} - \mathbf{b}\|_1 \in \mathbb{R}^{n\times 1}$

Thus, our KKT conditions state that:

$$\mathbf{x} = -\mathbf{A}^*\mathbf{u}$$

and

$$(\mathbf{A}\mathbf{x})_i \in \begin{cases} \frac{1}{\epsilon}, & \text{if } |u_i - b_i| > 0 \\ -\frac{1}{\epsilon}, & \text{if } |u_i - b_i| < 0 \\ [-\frac{1}{\epsilon}, \frac{1}{\epsilon}], & \text{if } |u_i - b| = 0 \end{cases}$$

where $(\mathbf{A}\mathbf{x})_i$ is the i^{th} element of $\mathbf{A}\mathbf{x} \in \mathbb{R}^{n\times 1}$

2.6.4

To derive a rate of convergence on the primal iterates from the applications of FISTA (Fast Iterative Shrinkage Threshold Algorithm) on the dual problem when strong duality holds, the distance to the primal solution is bounded by the dual objective values:

$$\frac{2}{\mu}\|\mathbf{x}_k - \bar{\mathbf{x}}\|^2 \leq \Psi(\mathbf{u}_k) - \Psi(\bar{\mathbf{u}})$$

where $\bar{\mathbf{x}} \in \mathbb{R}^{d\times 1}$ and $\bar{\mathbf{u}} \in \mathbb{R}^{n\times 1}$ are minimizers for the primal and dual solution respectively, $\mathbf{x}_k \in \mathbb{R}^{d\times 1}$ and $\mathbf{u}_k \in \mathbb{R}^{n\times 1}$ are respectively the primal and dual solutions for the k^{th} step of FISTA, and μ is the coefficient of strong convexity for f. For our problem, $\mu = 1$. Moreover, $\Psi(\mathbf{u}) = \frac{1}{2}\mathbf{u}^*\mathbf{A}\mathbf{A}^*\mathbf{u} + \frac{1}{\epsilon}\|\mathbf{u} - \mathbf{b}\|_1$, our dual objective.

We also know for FISTA that (proof in lecture):

$$\Psi(\mathbf{u}_k) - \Psi(\bar{\mathbf{u}}) \leq \frac{\|\mathbf{u}_0 - \bar{\mathbf{u}}\|^2}{2\gamma\left(t_{k-1}\right)^2}$$

where $\mathbf{u}_0 \in \mathbb{R}^{n\times 1}$ is the initial starting point for FISTA, and $0 < \gamma \leq \frac{1}{L}$ and t_{k-1} are both scalars dictating step sizes in FISTA (t_{k-1} is for step $k-1$). L is the Lipschitz smoothness of f. For our problem, $L = 1$.

Thus:

$$\|\mathbf{x}_k - \bar{\mathbf{x}}\|^2 \leq \frac{\|\mathbf{u}_0 - \bar{\mathbf{u}}\|^2}{4\gamma\left(t_{k-1}\right)^2}$$

Simplifying:

$$\|\mathbf{x}_k - \bar{\mathbf{x}}\| \leq \frac{\|\mathbf{u}_0 - \bar{\mathbf{u}}\|}{2\sqrt{\gamma}t_{k-1}}$$

Two choices for t_k in FISTA are:

- $t_k = \frac{1+\sqrt{1+4(t_{k-1})^2}}{2}$ where $\frac{1}{t_{k-1}} \leq \frac{2}{k+1}$
- $t_k = \frac{k+a}{a}$ with $a \geq 2$ where $\frac{1}{t_{k-1}} \leq \frac{a}{k+1}$

where $t_0 = 1$

In both cases the rate of convergence of the primal iterates when applying FISTA on the dual problem:

$$\|\mathbf{x}_k - \bar{\mathbf{x}}\| \leq \frac{a\|\mathbf{u}_0 - \bar{\mathbf{u}}\|}{\sqrt{\gamma}(k+1)}$$

where $a = 2$ if $t_k = \frac{1+\sqrt{1+4(t_{k-1})^2}}{2}$. In other words, the rate is $\mathcal{O}(\frac{1}{k})$.

Part 3: Solving the Lasso problem

Our problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2n}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2+\lambda\|x\|_1$$

where $\mathbf{A}\in\mathbb{R}^{n\times d}$, $\mathbf{x}\in\mathbb{R}^{d\times 1}$, $\mathbf{y}\in\mathbb{R}^{n\times 1}$, n is the number of data points, and d is the number of dimensions.

Equivalently, our problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2n}\sum_{i=1}^{n}\left(\left\langle \mathbf{a}^i,\mathbf{x}\right\rangle - y_i\right)^2+\lambda\|x\|_1$$

where $\mathbf{a}^i\in\mathbb{R}^{1\times d}$ is the i^{th} row of $\mathbf{A}$, $i=1,...,n$.

3.1

The Proximal Stochastic Gradient Algorithm:

$$\mathbf{x}^{k+1} = prox_{\gamma_k\lambda\|\cdot\|_1}\left(\mathbf{x}^k-\gamma_k\left(\left\langle \mathbf{a}^{i_k},\mathbf{x}^k\right\rangle - y_{i_k}\right)\mathbf{a}^{i_k}\right)$$

where:

$$\gamma_k = \frac{n}{\|A\|^2\sqrt{k+1}}$$

and

$$prox_{\gamma_k\lambda\|\cdot\|_1}(x) = soft_{\gamma_k\lambda}(x) = \begin{cases} 0, & \text{if } |x|\le\gamma_k\lambda \\ x-\gamma_k\lambda, & \text{if } x>\gamma_k\lambda \\ x+\gamma_k\lambda, & \text{if } x<-\gamma_k\lambda \end{cases}$$

and i_k is sampled uniformly from $\{1,...,n\}$ at each step k.

3.2

The Randomized Coordinate Proximal Gradient Algorithm:

$$x_j^{k+1} = \begin{cases} soft_{\gamma_j\lambda}(x_j^k-\frac{\gamma_j}{n}\left\langle a_j,\mathbf{A}\mathbf{x}^k-y\right\rangle), & \text{if } j=j_k \\ x_j^k, & \text{otherwise} \end{cases}$$

where we define $\mathbf{a}_j\in\mathbb{R}^{n\times 1}$ as the j^{th} column of $\mathbf{A}$, $i=1,...,d$.

$$\gamma_j = \frac{n}{\|\mathbf{a}_j\|^2}$$

and j_k is sampled uniformly from $\{1,...,d\}$ at each step k.

3.3

The initial vector passed to the algorithms:

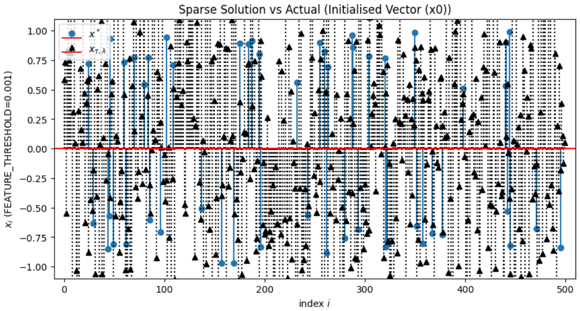

Figure 1: x_0 vs Sparse Vector

Plotting the objective function values vs the number of iterations for both the algorithms:

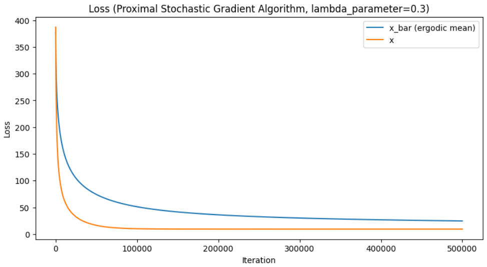

Figure 2: PSGA Loss

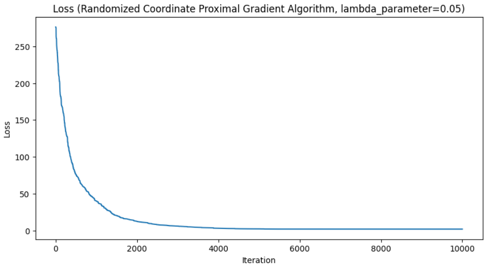

Figure 3: RCPGA Loss

The corresponding solution compared to the actual sparse vector for both the algorithms:

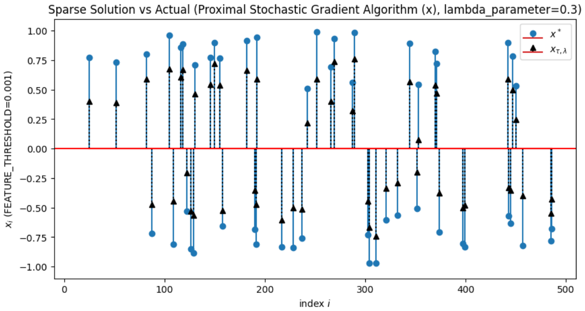

Figure 4: PSGA Solution vs Sparse Vector

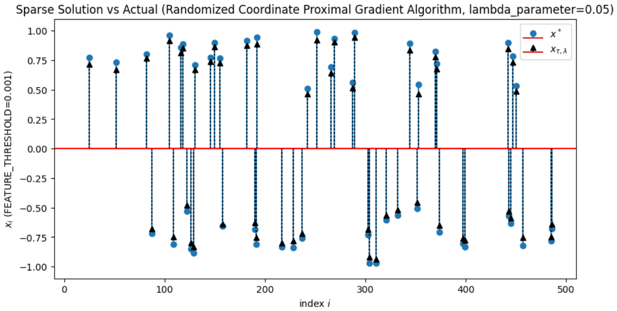

Figure 5: RCPGA Solution vs Sparse Vector

Part 4: Support Vector Machines

We are given the primal problem:

$$\min_{\mathbf{w}\in\mathcal{H}} \frac{1}{\lambda n}\sum_{i=1}^{n}(1-y_i\langle\mathbf{w},\Lambda(\mathbf{x}_i)\rangle)_+ + \frac{1}{2}\|\mathbf{w}\|^2$$

where $\mathcal{H}$ is a Hilbert space.
The primal problem can be expressed the form:

$$\min_{w\in\mathcal{H}} f(\mathbf{w}) + g(\mathbf{A}\mathbf{w})$$

where:

$$f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$$

and

$$g(\mathbf{A}\mathbf{w}) = \frac{1}{\lambda n}\sum_{i=1}^{n}(1-y_i\langle\mathbf{w},\Lambda(\mathbf{x}_i)\rangle)_+$$

where $\mathbf{A}_i = \Lambda(\mathbf{x}_i) \in \mathcal{H}$ our Hilbert space.
This corresponds to the dual problem of the form:

$$\min_{\mathbf{u}\in\mathbb{R}^n} f^*(-\mathbf{A}^*\mathbf{u}) + g^*(\mathbf{u})$$

where:

$$f^*(-\mathbf{A}^*\mathbf{u}) = \frac{1}{2}\|-\mathbf{A}^*\mathbf{u}\|^2 = \frac{1}{2}\mathbf{u}^*\mathbf{A}\mathbf{A}^*\mathbf{u}$$

and using the kernel defined by $\mathcal{H}$, we can substitute $\mathbf{A}\mathbf{A}^*$ with the gram matrix $\mathbf{K}$:

$$f^*(-\mathbf{A}^*\mathbf{u}) = \frac{1}{2}\mathbf{u}^*\mathbf{K}\mathbf{u}$$

Moreover,

$$g^*(\mathbf{u}) = -\langle\mathbf{y},\mathbf{u}\rangle + \sum_{i=1}^{n} i_{\left[0,\frac{1}{\lambda n}\right]}(\mathbf{u}_i)$$

Defining $\alpha_i = u_i y_i$ and know that $y_i y_i = 1$ because $y_i \in \{-1, 1\}$ we can see that:

$$f^*(-\mathbf{A}^*\mathbf{u}) = \frac{1}{2}\alpha^*\mathbf{K}_y\alpha$$

and

$$g^*(\mathbf{u}) = -\langle\mathbf{1}_n,\alpha\rangle + \sum_{i=1}^{n} i_{\left[0,\frac{1}{\lambda n}\right]}(\alpha_i)$$

where $(\mathbf{K}_y)_{i,j} = y_i\mathbf{K}_{i,j}y_j$.
Combining, we have our dual problem:

$$\min_{\alpha\in\mathbb{R}^n} \frac{1}{2}\alpha^*\mathbf{K}_y\alpha - \langle\mathbf{1}_n,\alpha\rangle + \sum_{i=1}^{n} i_{\left[0,\frac{1}{\lambda n}\right]}(\alpha_i)$$

4.1

The Fast Iterative Shrinkage Threshold Algorithm (FISTA):

$$\alpha_{k+1} = prox_{\gamma g^*(\cdot)}\bigg(\nu_k + \gamma \mathbf{A}_y \nabla f^*(-\mathbf{A}_y^* \alpha_k)\bigg)$$

$$\nu_{k+1} = \alpha_k + \frac{t_k - 1}{t_{k+1}}(\alpha_{k+1} - \alpha_k)$$

where $t_k = \frac{1+\sqrt{1+4(t_{k-1})^2}}{2}$ or $t_k = \frac{k+a}{a}$ with $a \geq 2$. For our implementation we chose the first definition for t_k.

For our dual problem:

$$\nabla f^*(-\mathbf{A}_y^* \alpha_k)) = -\mathbf{A}_y^* \alpha_k$$

and

$$prox_{\gamma g^*(\cdot)}(\omega) = \begin{cases} 0, & \text{if } \omega < 0 \\ \omega, & \text{if } 0 \leq \omega \leq \frac{1}{\gamma n} \\ \frac{1}{\gamma n}, & \text{if } \frac{1}{\gamma n} < \omega \end{cases}$$

With FISTA implemented, we can plot the dual objective function:

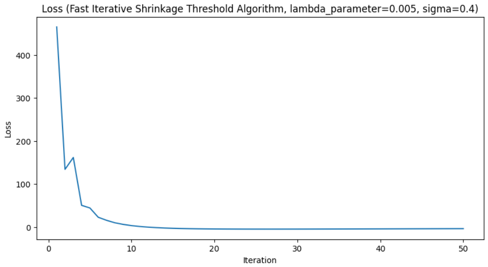

Figure 6: FISTA Loss

4.2

We can also implement the Randomised Coordinate Projected Gradient Algorithm on the dual problem and plot the dual objective function:

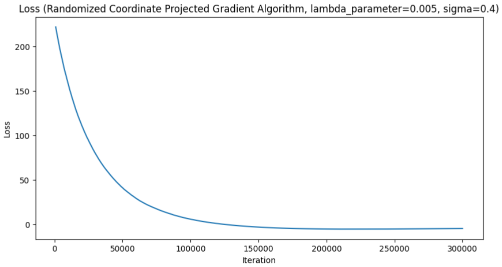

Figure 7: RCPGA Loss

4.3

We can plot the decision boundary for the randomly initialised α vector as well as the two classes:

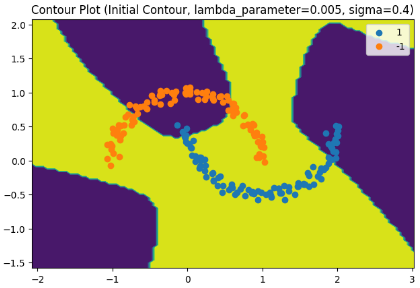

Figure 8: Initial Contour Plot

Similarly for each algorithm, we can plot the decision boundaries from the learned α:

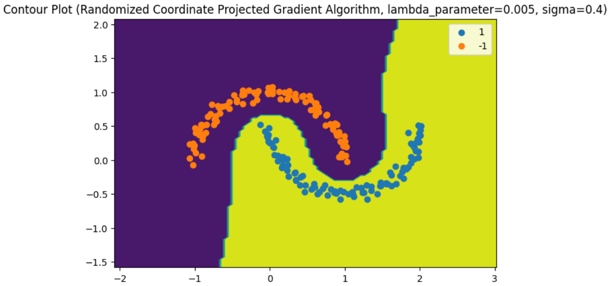

Figure 9: RCPGA Contour Plot

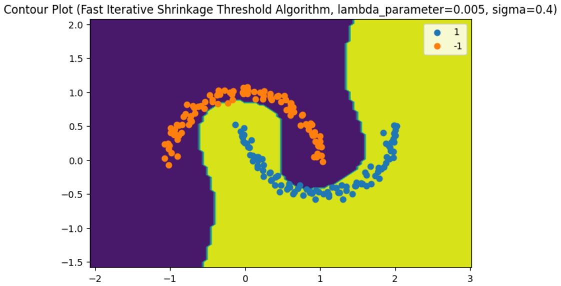

Figure 10: FISTA Contour Plot

4.4

Comparing the two algorithms, we can see that both are able to converge to α vectors producing decision boundaries that successfully separate the two training points for the two classes. However, the FISTA decision boundary is quite close to the points for both classes (i.e. the decision boundary is not as curves as for RCPGA). Thus, the quality of the solution is not as nice as the RCPGA where there is ample buffer surrounding the decision boundary for both classes. This would seem that RCPGA can produce more reasonable results, however considering the loss function plots for both algorithms, we can see that this comes at a tradeoff of computational resources/speed. FISTA is able to converge in around 50 iterations whereas RCPGA required on the order of 3×10^5 iterations. Although each FISTA step involves two steps (to calculate α_{k+1} and ν_{k+1}), this is still a much faster algorithm than RCGPA. Thus, depending on the practical use case, if there is a requirement on the algorithm's speed, then FISTA may be more appropriate. On the other hand, if we have more resources available, then RCGPA may be more appropriate as it seems to produce more reasonable/accurate results.