COMP 0083: ATML: Introduction to Convex Optimization **2021/2022**

Coursework

Due Date: **11th Jan 2022** *Instructor: Saverio Salzo*

1: Multiple Choice

Q1: b) ϕ is increasing

Q2: a) $2A^*(Ax-b)$

Q3: a) $f^*(u) = g^*(u) + 2u$

Q4: c) $\bar{u} = (K + \lambda n \mathrm{Id})^{-1} y$

2: Theory

P1

1.

$$\begin{aligned} f(x) &= e^x \\ f^*(u) &= \sup_{x\in\mathbb{R}} [xu - e^x] \end{aligned}$$

If $u < 0$, x can be made infinitely negative to give a supremum of ∞. If $u > 0$, the maximum is found by differentiation to give the stationary point, this gives a maximum of $u\ln u - u$. If $u = 0$, we can take the limit 0^+ of the $u > 0$ case, to give 0. Hence,

$$f^*(u) = \begin{cases} \infty & u < 0 \\ 0 & u = 0 \\ u\ln u - u & u > 0 \end{cases}$$

2.

$$\begin{aligned} f(x) &= \iota_{[-1,0]}(x) \\ f^*(u) &= \sup_{x\in\mathbb{R}} \left[xu - \iota_{[-1,0]}(x)\right] \end{aligned}$$

For $u \leq 0$ this is maximised by $x = -1$, for $u \geq 0$ this is maximised by $x = 0$. Hence,

$$f^*(u) = \begin{cases} -u & u \leq 0 \\ 0 & u > 0 \end{cases}$$

3.

$$\begin{aligned} f(x) &= \iota_{\{a\}}(x) \\ f^*(u) &= \sup_x \left[\langle x, u\rangle - \iota_{\{a\}}(x)\right] \end{aligned}$$

It is straightforward to see that this is maximised by $x = a$ $\forall u$. Hence,

$$f^*(u) = \langle a, u\rangle$$

P2

1. We are given that

$$f((1-\lambda)x + \lambda y) \leq f(x)^{1-\lambda} f(y)^{\lambda} \tag{1.1}$$

$$\begin{aligned} \log(f(x)^{1-\lambda} f(y)^{\lambda}) &= (1-\lambda)\log(f(x)) + \lambda\log(f(y)) \\ &\Leftrightarrow \log(f((1-\lambda)x + \lambda y)) \leq (1-\lambda)\log(f(x)) + \lambda\log(f(y)) \\ &\Leftrightarrow f \text{ is logarithmically convex} \end{aligned}$$

Where in getting to the second line, we used the fact that the log is monotonic increasing and the inequality 1.1.

2. Given $\ln(f(x))$ is convex, and using the hint that $\phi(g(x))$ is convex if g is convex and ϕ is increasing and convex. We choose $\phi(x) = e^x$ and $g(x) = \ln(f(x))$. It is known that e^x is convex and increasing and f is logarithmically convex $\Rightarrow \exp(\ln(f(x))) = f(x)$ is convex.

3. *Propositon*: If f is logarithmically convex, then $f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \prod_{i=1}^{n} f(x_i)^{\lambda_i}$ $\forall x_i \in \mathsf{X}$, $(\lambda_i)_{1\leq i\leq n} \in \mathbb{R}^n_+$ such that $\sum_{i=1}^{n} \lambda_i = 1$

We show it's true for $n = 1$: $\lambda = 1 \Rightarrow f(\lambda x) = f(x)^{\lambda}$. proposition is true for $n = 1$

We assume true for $n = k$.

Induction hypothesis: $f\left(\sum_{i=1}^{k} \lambda_i x_i\right) \leq \prod_{i=1}^{k} f(x_i)^{\lambda_i} \Leftrightarrow \log f\left(\sum_{i=1}^{k} \lambda_i x_i\right) \leq \sum_{i=1}^{k} \lambda_i \log(f(x_i))$

Induction step: For brevity we write $\phi(x) = \log(f(x))$. We assume that $\lambda_{k+1} \neq 1$, as if $\lambda_{k+1} = 1$ then the proposition is trivially true.

$$\begin{aligned}
\phi\left(\sum_{i=1}^{k+1}\lambda_i x_i\right) &= \phi\left((1-\lambda_{k+1})\left(\sum_{i=1}^{k}\mu_i x_i\right)+\lambda_{k+1}x_{k+1}\right) \text{ where } \mu_i = \frac{\lambda_i}{1-\lambda_{k+1}} \\
&\leq (1-\lambda_{k+1})\phi\left(\sum_{i=1}^{k}\mu_i x_i\right)+\lambda_{k+1}\phi(x_{k+1}) \text{ By convexity of } \phi \\
&\leq \sum_{i=1}^{k}\lambda_i\phi(x_i)
\end{aligned}$$

Where in the last step we can use the induction hypothesis, because $\sum_{i=1}^{k}\mu_i = 1$ by construction. Hence,

$$\begin{aligned}
\log f\left(\sum_{i=1}^{k+1}\lambda_i x_i\right) &\leq \sum_{i=1}^{k+1}\lambda_i \log(f(x_i)) \\
\Leftrightarrow f\left(\sum_{i=1}^{k+1}\lambda_i x_i\right) &\leq \prod_{i=1}^{k+1} f(x_i)^{\lambda_i}
\end{aligned}$$

This is therefore true $\forall n$

4. We prove $\sum_{i=1}^{n}\lambda_i x_i \geq \prod_{i=1}^{n} x_i^{\lambda_i}$ for all $x_i \in \mathbb{R}_{++}$ for $\lambda_i \in [0,1]$ and $\sum_{i=1}^{n}\lambda_i = 1$ $\forall n \in \mathbb{N}$ by induction.

$n = 1$ case: This is trivially true

Induction hypothesis:

$$\begin{aligned}
\sum_{i=1}^{k}\lambda_i x_i &\geq \prod_{i=1}^{k} x_i^{\lambda_i} \\
\Leftrightarrow \log\left(\sum_{i=1}^{k}\lambda_i x_i\right) &\geq \sum_{i=1}^{k}\lambda_i \log(x_i)
\end{aligned}$$

Induction step:

$$\begin{aligned}
\log\left(\sum_{i=1}^{k+1}\lambda_i x_i\right) &= \log\left((1-\lambda_{k+1})\left(\sum_{i=1}^{k}\mu_i x_i\right)+\lambda_{k+1}x_{k+1}\right) \text{ where } \mu_i = \frac{\lambda_i}{1-\lambda_{k+1}} \\
&\geq (1-\lambda_{k+1})\log\left(\sum_{i=1}^{k}\mu_i x_i\right)+\lambda_{k+1}\log(x_{k+1}) \text{ By concavity of log} \\
&\geq \sum_{i=1}^{k}\lambda_i \log(x_i)
\end{aligned}$$

Where in the last step we can use the induction hypothesis, because $\sum_{i=1}^{k}\mu_i = 1$ by construction. Hence,

$$\begin{aligned} \log\left(\sum_{i=1}^{k+1}\lambda_i x_i\right) &\geq \sum_{i=1}^{k+1}\lambda_i \log(x_i) \\ \Leftrightarrow \sum_{i=1}^{k+1}\lambda_i x_i &\geq \prod_{i=1}^{k+1} x_i^{\lambda_i} \end{aligned}$$

This is therefore true $\forall n$

P3

Prove that the maximum of a convex function f on polytope $\mathcal{C} = \text{co}(a_1, \ldots, a_m)$ is attained at one of the vertices.

We take some $x \in \mathcal{C}$, we can represent it as a convex combination of the vertices, $x = \sum_i \lambda_i a_i$ with for $\lambda_i \in [0,1]$ and $\sum_i \lambda_i = 1$.

$$\begin{aligned} f(x) = f(\sum_i \lambda_i a_i) &\leq \sum_i \lambda_i f(a_i) \text{ by convexity} \\ &\leq \sum_i \lambda_i f(c_0) \text{ where } f(c_0) = \max_{1\leq i\leq m}[f(c_i)] \\ &= f(c_0) \\ \Rightarrow f(x) &\leq f(c_0)\ \forall x \end{aligned}$$

Proving the maximum is attained on one of the vertices.

P4

A function $f(x,y)$ is jointly convex if

$f(\lambda x_1 + (1-\lambda)x_2, \lambda y_1 + (1-\lambda)y_2) \leq \lambda f(x_1, y_1) + (1-\lambda) f(x_2, y_2)$

We prove this for $f(x,y) = \|x-y\|^2$.

$$\begin{aligned} f(\lambda x_1 + (1-\lambda)x_2, \lambda y_1 - (1-\lambda)y_2) &= \|\lambda x_1 + (1-\lambda)x_2 - \lambda y_1 + (1-\lambda)y_2\|^2 \\ &= \|\lambda(x_1 - y_1) + (1-\lambda)(x_2 - y_2)\|^2 \\ \text{(triangle inequality)}\ &\leq \lambda\|x_1 - y_1\|^2 + (1-\lambda)\|x_2 - y_2\|^2 \\ &= \lambda f(x_1, y_1) + (1-\lambda) f(x_2, y_2) \end{aligned}$$

P5

Minimal sufficient conditions for the existence of minimizers for a convex function $f : \mathcal{X} \to]-\infty, +\infty]$: f must be proper, closed and coercive.

Minimal sufficient conditions for the uniqueness of a minimizer for a convex function $f : \mathcal{X} \to]-\infty, +\infty]$: f must be proper and strictly convex.

P6

1. Primal optimization problem where $A : \mathbb{R}^d \to \mathbb{R}^n$:

$$\min_{Ax=b} \frac{1}{2}\|x\|^2$$

To find the dual problem we rewrite the primal problem as

$$\min_{x \in \mathbb{R}^d} \left[\frac{1}{2}\|x\|^2 + \iota_{\{b\}}(Ax) \right]$$

This is in the form

$$\min_{x \in \mathbb{R}^d} [f(x) + g(Ax)]$$

With $f(x) = \frac{1}{2}\|x\|^2$ and $g(u) = \iota_{\{b\}}(u)$. We find the dual functions:

$$\begin{aligned} f^*(u) &= \sup_{x \in \mathbb{R}^d} \left[\langle x, u \rangle - \frac{1}{2}\|x\|^2 \right] \\ &= \sup \left[-\frac{1}{2}\|x - u\|^2 + \frac{1}{2}\|u\|^2 \right] \\ &= \frac{1}{2}\|u\|^2 \end{aligned}$$

The dual problem takes the form

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

$g^*(u) = \langle u, b \rangle$ as found in the previous section. So the dual problem is

$$\begin{aligned} &\min_{u \in \mathbb{R}^n} \left[\frac{1}{2}\|A^*u\|^2 + \langle u, b \rangle \right] \\ =&\min_{u \in \mathbb{R}^n} \left[\frac{1}{2}\langle AA^*u, u \rangle + \langle u, b \rangle \right] \end{aligned}$$

2. Strong duality:

We would like to minimise $f(x) = \frac{1}{2}\|x\|^2$ given the constraint $Ax = b$. Slater's condition states that strong duality holds if $\exists x^* \in \mathrm{dom}(f)$ that satisfies all constraints. Since $\mathrm{dom}(f) = \mathbb{R}^d$ and $A : \mathbb{R}^d \to \mathbb{R}^n$, if $\exists x^*$ such that $Ax^* = b$, then strong duality holds.

Conversely, as the dual of the dual problem is the primal, one can check for strong duality by applying Slater's condition to the dual problem: We would like to minimise $g^*(u)$ with $u \in \mathbb{R}^n$ unconstrained. Slater's condition is clearly satisfied, hence strong duality holds.

3. The KKT conditions:

Let $\hat{x}$ and $\hat{u}$ be minimizers of the primal and dual problems respectively. Then the KKT conditions are:

1. $\hat{x} \in \partial f^*(-A^*\hat{u})$ and $A\hat{x} \in \partial g^*(\hat{u})$
2. $-A^*\hat{u} \in \partial f(\hat{x})$ and $\hat{u} \in \partial g(A\hat{x})$

1. and 2. are equivalent. We write out all the functions that need to be considered below:

$$\begin{aligned} f &= \frac{1}{2}\|\cdot\|_2^2 & g &= \iota_{\{b\}}(\cdot) \\ f^* &= \frac{1}{2}\|\cdot\|_2^2 & g^* &= \langle \cdot, b \rangle \end{aligned}$$

We now define the subdifferentials:

$$\begin{aligned} \partial f(x) &= x \\ \partial f^*(x) &= x \end{aligned}$$

$$\begin{aligned} \partial g(u) &= \begin{cases} \mathbb{R}^n & u = b \\ 0 & u \neq b \end{cases} \\ \partial g^*(u) &:= \{x \in \mathbb{R}^n | \forall y \in \mathbb{R}^n, \langle y, b \rangle \geq \langle u, b \rangle + \langle x, y - u \rangle\} \\ &= \{x \in \mathbb{R}^n | \forall y \in \mathbb{R}^n, \langle b - x, y - u \rangle \geq 0\} \\ &= b \end{aligned}$$

The KKT conditions are then:

1. $\hat{x} = -A^*\hat{u}$ and $A\hat{x} = b$
2. $-A^*\hat{u} = \hat{x}$ and $\hat{u} \in \mathbb{R}^n$

These are equivalent.

4. Rate of convergence of the primal iterates from the application of FISTA on the dual problem:

$f^*(-A^*u) = \frac{1}{2}\langle AA^*u, u\rangle$ is L-Lipschitz smooth with $L = \|AA^*\|$, (the largest aigenvalue of AA^*). FISTA is applied by iteratively updating u_k according to

$$\begin{aligned} u_{k+1} &= \operatorname{prox}_g\left(y_k - \gamma \nabla f(-A^* y_k)\right) \\ y_{k+1} &= \alpha_{k+1} + \frac{t_k - 1}{t_{k+1}}(u_{k+1} - u_k) \end{aligned}$$

with $t_0 = 1$ and $t_k = (1 + \sqrt{1 + 4t_{k-1}})/2$, and $\gamma \leq 1/L$. Defining $\psi(u) = g^*(u) + f^*(-A^*u)$, we have the Theorem: for k iterations of FISTA $\forall k \in \mathbb{N}$

$$\begin{aligned} \psi(u_k) - \min\psi &\leq \frac{\|y_0 - \hat{u}\|^2}{2\gamma t_{k-1}^2} \\ &\leq \frac{L\|y_0 - \hat{u}\|^2}{2t_{k-1}^2} \end{aligned}$$

Where we set $\gamma = 1/L$ in the second line. Moreover, if $t_k = (1 + \sqrt{1 + 4t_{k-1}})/2$, it can be shown (notes of Lecture 9) that $\frac{1}{t_{k-1}} \leq \frac{2}{k+1}$, therefore

$$\psi(u_k) - \min\psi \leq \frac{4L\|y_0 - \hat{u}\|^2}{2(k+1)^2} \tag{1.2}$$

We now define: $\phi(x) = f(x) + g(Ax)$, and $\hat{x}$ and $\hat{u}$ to be minimizers of ϕ and ψ respectively. $f(x) = \frac{1}{2}\|x\|^2$ is μ-strongly convex with $\mu = 1$. This can easily be seen by noting that $(\nabla f(x) - \nabla f(y))^\intercal(x - y) = \|x - y\|$. If the KKT conditions are satisfied, we have the following theorem:

$$\frac{\mu}{2}\|x - \hat{x}\|^2 \leq \psi(u) - \psi(\hat{u}) \tag{1.3}$$

Combing equations 1.2 and 1.3, we get

$$\|x_k - \hat{x}\| \leq \frac{2\|y_0 - \hat{u}\|\sqrt{\|AA^*\|}}{k+1}$$

The primal iterates converge at a rate of O($1/k$)

3: LASSO Problem

The problem

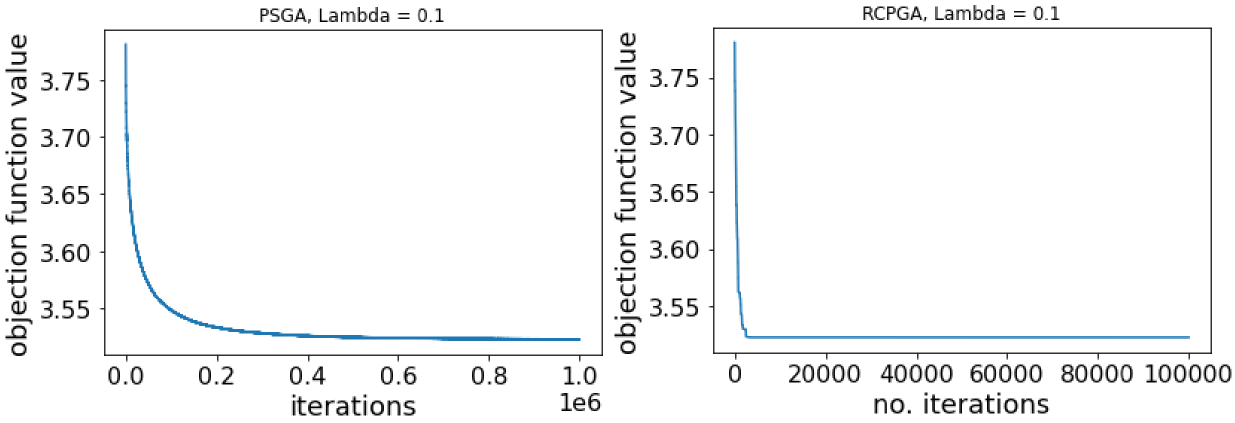

Figure 1.1. Left: objective value plot for PSGA, Right: objective value plot for RCPGA

1. PSGA was implemented using the soft thresholding operator acting component wise. This can be done as the 1-norm is separable in the components of its argument. The update of the νth component of x can be written:

$$x_\nu^{k+1} = \text{soft}_{\gamma_k\lambda}(x_\nu^k - \gamma_k(\langle a^{i_k}, x\rangle - y_{i_k})a_\nu^{i_k})$$

2. RCPGA was implemented

3. $\lambda = 0.1$ was chosen, and the objective function values vs number of iterations was plotted for each of the algorithms. The plots are shown in figure 1.1. The RCPGA algorithm converged much faster to the sparse solution. The PSGA algorithm had much slower convergence, and even after millions of iterations did not converge to the true sparse solution. This is likely due to γ_k decaying too quickly.

The objective values over the sequence of ergodic means $\left(\bar{x}^k = \left(\sum_{i=0}^k \gamma_i\right)^{-1} \sum_{i=0}^k \gamma_i x_i\right)$ was plotted against iteration number for the PSGA algorithm. The results are shown in figure 1.2, along with a plot of the objective function values not using the ergodic means for a comparison.

To assist with convergence using the PSGA method, the γ_k were multiplied by a constant θ. The experiment was repeated using different values of $\theta \in [0, 1]$. The results are plotted in figure 1.3. Smaller values of theta give slower convergence, this is understandable as smaller theta means smaller step sizes. As a result, none of the plots converged to the true sparse solution even after millions of iterations. The rate of decay of γ may be too fast, so the experiment for PSGA was again repeated using a γ that decayed with the cube root of k instead of the square root, again this did not converge to the true sparse solution even after millions of iterations.

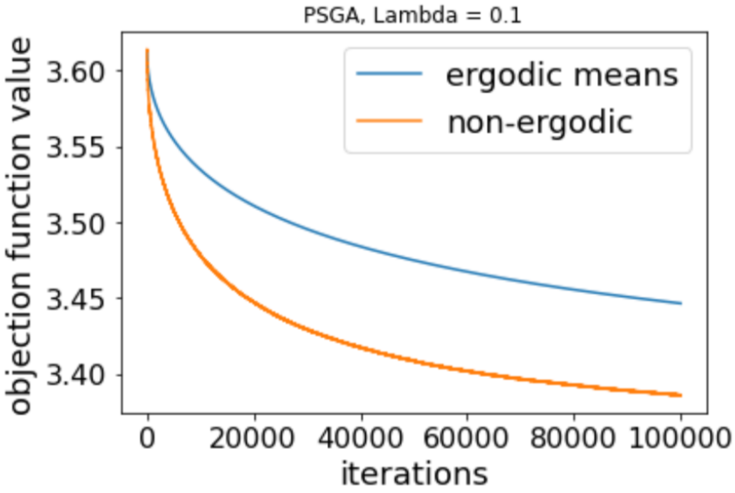

Figure 1.2. PSGA plot of the objective function value over the ergodic means

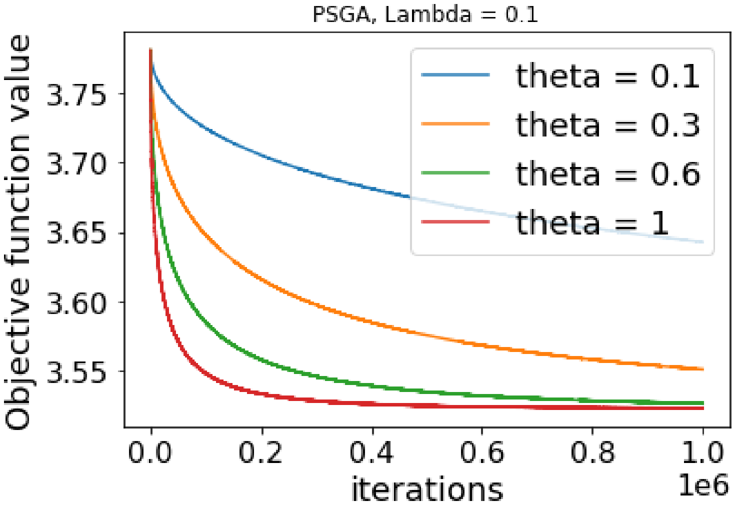

Figure 1.3. PSGA with step size multiplied by a constant θ

4: Binary Classification SVM

1. The dual problem:

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

FISTA is applied by

$$\begin{aligned}\alpha_{k+1} &= \operatorname{prox}_g\left(y_k - \gamma\nabla f(y_k)\right)\\ y_{k+1} &= \alpha_{k+1} + \frac{t_k-1}{t_{k+1}}(\alpha_{k+1}-\alpha_k)\end{aligned}$$

with $t_0 = 1$ and $t_k = (1+\sqrt{1+4t_{k-1}})/2$, and $\gamma \leq 1/L$, where L is the Lipschitz constant of f.

$$\begin{aligned}f(\alpha) &= \frac{1}{2}\langle K_y,\alpha\rangle - \langle \mathbf{1}_n,\alpha\rangle\\ \text{and } g(\alpha) &= \sum_{i=1}^{n}\iota_{[0,1/\lambda n]}(\alpha_i)\end{aligned}$$

The Lipschitz constant is found by observing:

$$\begin{aligned}f(\alpha) &= K_y\alpha - \mathbf{1}_n\\ \Leftrightarrow \|f(x)-f(y)\|_2 &= \|K_y(x-y)\|_2\\ &\leq \|K_y\|_F\|x-y\|_2\end{aligned}$$

L is chosen to be the smaller of $\|K_y\|_F$ or the largest eigenvalue of K_y. (where $\|\cdot\|_F$ is the Frobenius norm.

As $g(\alpha)$ is separable in the components of the argument, the proximal operator of g can be written

$$\operatorname{prox}_g(c) = \Big(\operatorname{prox}_{\iota_{[0,1/\lambda n]}}(c_1),\ldots,\operatorname{prox}_{\iota_{[0,1/\lambda n]}}(c_n)\Big)$$

The proximal operator for an indicator function of set $\mathcal{C}$ is simply the projection operator into that set.

$$\operatorname{prox}_{\iota_{[0,1/\lambda n]}}(c_i) = \begin{cases} c_i & c_i \in [0,1/\lambda n]\\ 0 & c_i < 0\\ 1/\lambda n & c_i > 1/\lambda n\end{cases}$$

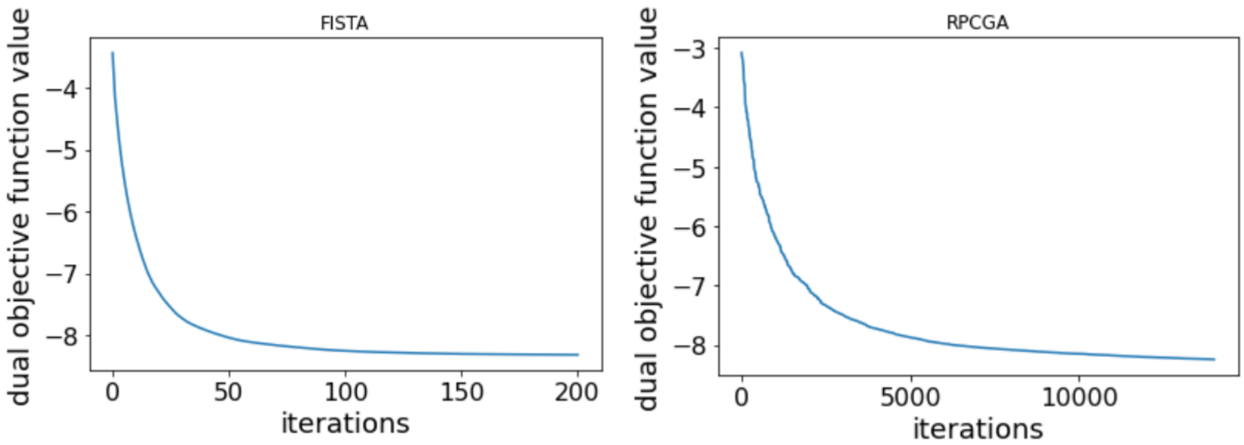

Figure 1.4. Dual objective function value against iterations for FISTA (left) and RPGCA (right) with $\sigma = 0.5$ and $\lambda = 0.1$

The FISTA algorithm was implemented for $\sigma = 0.5$ and $\lambda = 0.1$, plot of iterations vs dual objective function value is shown in figure 1.4.

2. The RPGCA algorithm is given below, it is identical to PSGA except that a random component 'i' of the argument is updated at every step. The update of the random component 'i' is given by

$$\alpha_i^{k+1} = \text{prox}_{g_i}\left(\alpha_i^k - \gamma_i \nabla_i f(\alpha^k)\right)$$

Where $\gamma_i < 2/L_i$, where L_i is the elementwise Lipschitz constant. I did not know how to calculate the elementwise Lipschitz constants for the problem, so I heuristically chose a value of *gamma* to use for all elements. The same f and g are used as defined previously. The RPGCA algorithm was implemented for $\sigma = 0.5$ and $\lambda = 0.1$, a plot of iterations vs dual objective function value is shown in figure 1.4.

3. The decision boundary along with the two classes is shown with contour plots for the resultant α found by both the FISTA and RPGCA algorithms in figur 1.5. In both cases $\sigma = 0.5$ and $\lambda = 0.1$.

4. FISTA converges much faster than RPCGA, even when γ (the step size) was chosen to be 3 times larger for RPCGA, this is understandable as only one component of α is updated in each iteration of RPCGA. Both algorithms result in near identical looking contour plots. Their accuracy was tested on a training set of 10000 points (generated in the same way as the original 200 points), in both cases the accuracy was 100%.

The entire above process was repeated, but instead I used a different proximal operator. We define:

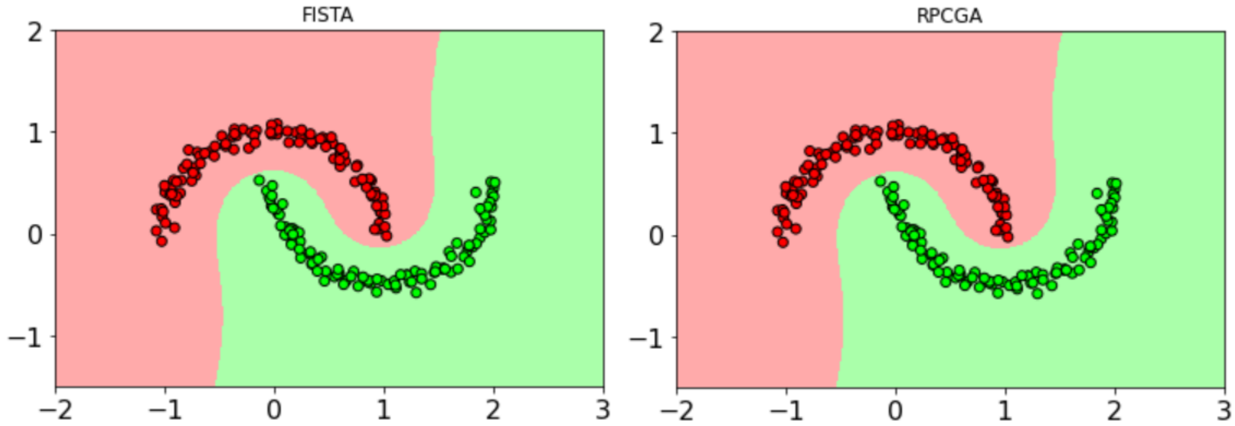

Figure 1.5. Contour plots for FISTA (left) and RPCGA (right)

$$\begin{aligned} f(\alpha) &= \frac{1}{2}\langle K_y, \alpha\rangle \\ \text{and } g(\alpha) &= \sum_{i=1}^{n} \iota_{[0,1/\lambda n]}(\alpha_i) - \langle \mathbf{1}_n, \alpha\rangle \\ &= \sum_{i=1}^{n} \big(\iota_{[0,1/\lambda n]}(\alpha_i) - \alpha_i\big) \end{aligned}$$

Again, $g(\alpha)$ is separable in the components of α. The corresponding elementwise proximal operator can be found:

$$\begin{aligned} \operatorname{prox}_{g_i}(\alpha_i) &= \operatorname*{argmin}_{y\in\mathbb{R}} \left[\frac{1}{2}\|\alpha_i - y\|^2 + \iota_{[0,1/\lambda n]} - y\right] \\ &= \operatorname*{argmin}_{y\in[0,1/n\lambda]} \left[\frac{1}{2}\|\alpha_i - y\|^2 - y\right] \\ &= \begin{cases} \alpha_i + 1 & \alpha_i + 1 \in [0, 1/\lambda n] \\ 0 & \alpha_i + 1 < 0 \\ 1/\lambda n & \alpha_i + 1 > 1/\lambda n \end{cases} \end{aligned}$$

The contour plots found from FISTA and RPCGA using this newly defined proximal operator are shown on the next page. Interestingly, the resulting decision boundary is differently shaped to the previously found boundary.

Appendix

The code for questions 3 and 4 is included in the ipynb files.

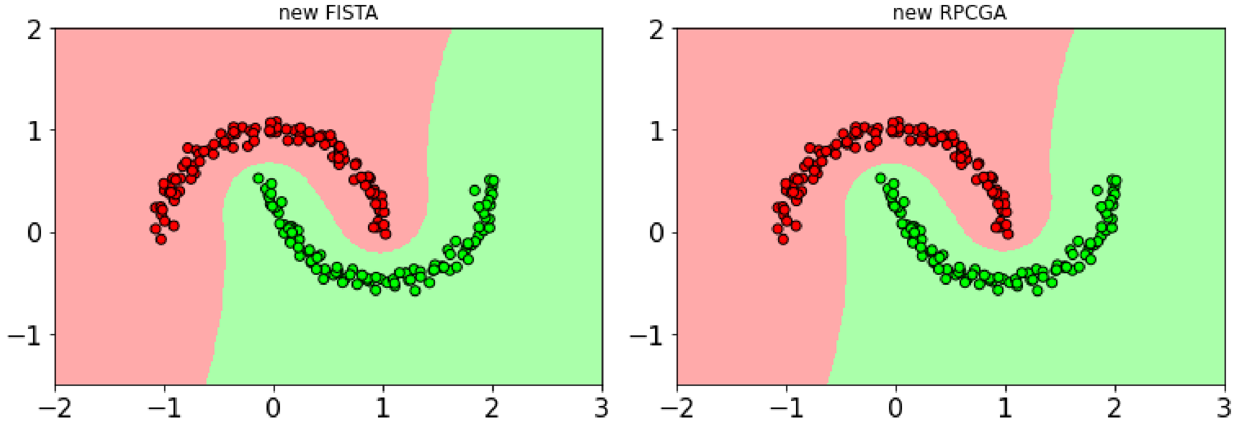

Figure 1.6. Caption