Question 1:

Apply randomized block coordinate descent to your problem by considering a full batch of your dataset, and compare it with gradient descent. Is the use of coordinate descent beneficial on your problem?

The coordinate descent method implemented here selects, at iteration k, a single coordinate $j_k \in \{1, \ldots, d\}$ and for a predetermined stepsize $\alpha_k > 0$, performs the following gradient update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla_{j_k} f(\mathbf{x}_k) \mathbf{e}_{j_k},$$

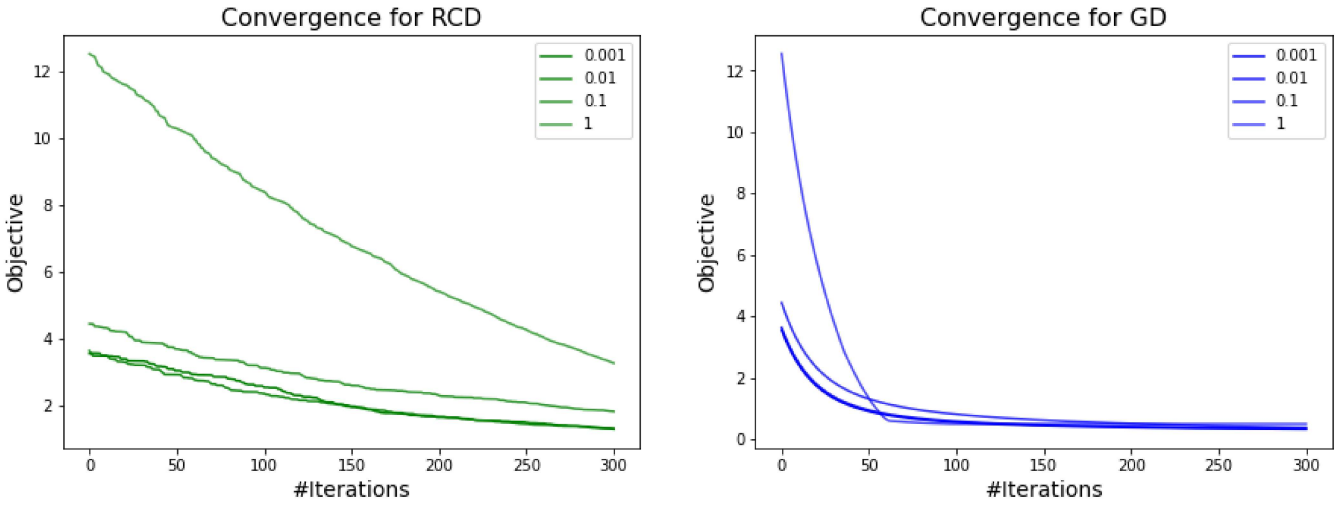
where ∇_{j_k} is shorthand for $\nabla_{j_k} f(\mathbf{x}) := [\nabla f(\mathbf{x})]_{j_k}$, $x \in \mathbb{R}^d$ and $\mathbf{e}_{j_k}$ canonical basis vector for coordinate j_k in $\mathbb{R}^d$.

We implement RCD with regularisation parameters

$$\lambda \in \{0.001, 0.01, 0.1, 1\}$$

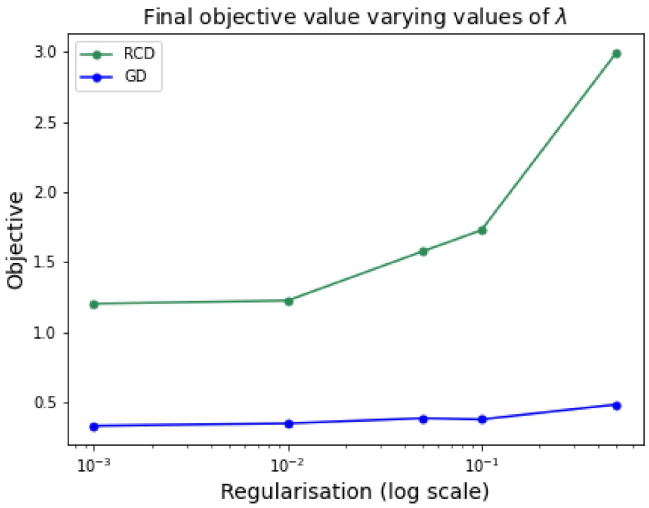
and run each iteration 300 times for a fixed step size $\alpha = 0.015$ which is optimal (as determined in Part 1).

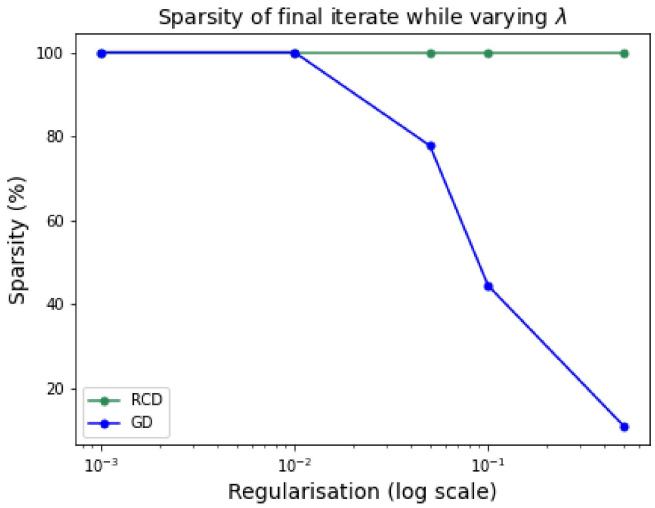
The results are plotted below:



From the above plots we see that even for a number of different regularisation parameters, GD performs better than RCD. Note that GD is simply RCD, but with a single block. The objective value for GD converges quickly to 0 whereas RCD has much slower convergence. This makes sense since less information is incorporated at each step of the algorithm when only a single coordinate is updated.

Consider also the final iterates of GD vs. those of RCD for different values of λ and we note that GD consistenly performs better than RCD.





The regularisation parameter enforces sparsity, and has a greater effect on gradient descent by forcing it to converge faster.

Question 2: Stochastic RCD

Combine randomized block coordinate descent with stochastic gradient (i.e. the method from Part 3). Do you observe a benefit from using coordinates together with stochastic gradient?

The coordinate descent method (RCD) may be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla_{j_k} f(\mathbf{x}_k) \mathbf{e}_{j_k},$$

and stochastic gradient (SG) as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f_{i_k}(\mathbf{x}_k),$$

We combine the two methods SG and RCD to obtain Randomised Coordinate Stochastic Gradient Coordinate (RCSG) which may be expressed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla_{j_k} f_{i_K}(\mathbf{x}_k) \mathbf{e}_{j_k},$$

where $j_k \in \{1, \ldots, d\}$ and $i_k \in \{1, \ldots, n\}$

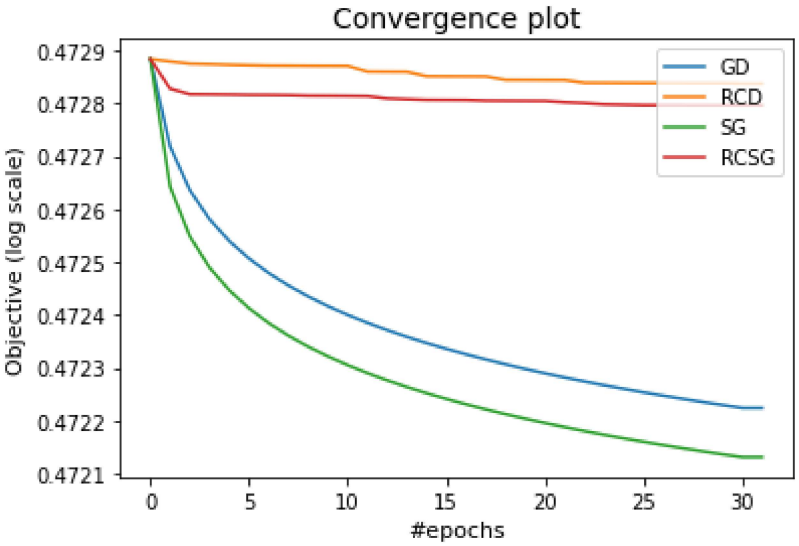
```
Epoch 49
(51, 2)
```

We now have 4 methods of interest to be compared, namely

- Gradient descent
- Randomised coordinate descent
- Stochastic Gradient
- Randomised Coordinate Stochastic Gradient

We run a single iteration of 30 epochs using the same stepsize $\frac{1}{L_{\max}}$ for every coordinate.

```
METHOD  FINAL OBJECTIVE
GD:     0.472224
RCD:    0.472838
SG:     0.472131
RCSG:   0.472797
```



The above figure shows that vanilla SG still outperforms all the other methods. Both variants of RCD perform worse than their full-coordinate counterparts, which makes sense in that at every epoch the same number of accesses to the gradient is performed, but less coordinates are used to calculate an update and thus the algorithm has less information at its disposal, resulting in steps which are less optimal.