Your name: (Rohan Chandra)

Answer the following questions. Latex your solutions into this document, and submit a pdf on Elms.

1. Question 1
   (a) Unwrapped ADMM

Solution: Unwrapped ADMM:

$$\min \frac{1}{2}||x||^2 + C\sum_i h(y_i)$$

such that $y - Ax = 0$

   (b) Augmented langrangian(scaled):

Solution:

$$L_\tau = \frac{1}{2}||x||^2 + C\sum_i h(y_i) + \sum_i \frac{\tau}{2}||y_i - A_i x + \lambda_i||^2$$

   (c) X update:

Solution:

$$x^{k+1} = \text{argmin}_x \frac{1}{2}||x||^2 + \frac{\tau}{2}||y - Ax + \lambda||^2$$

$$(I + \tau A^T A)x^{k+1} = \tau A^T(y^{k+1} + \lambda^k)$$

   (d) Y-update

Solution:

$${y_i}^{k+1} = \text{prox}_h(A_i x^k - {\lambda_i}^k, C/\tau)$$

   (e) λ-update

Solution:

$${\lambda_i}^{k+1} = {\lambda_i}^k + {y_i}^{k+1} - Ax^{k+1}$$

   (f) Value for N

Solution: For $\tau = 1$, $C = 10$ and a tolerance, $\epsilon = 1e^{-3}$,
$N = 1800$.

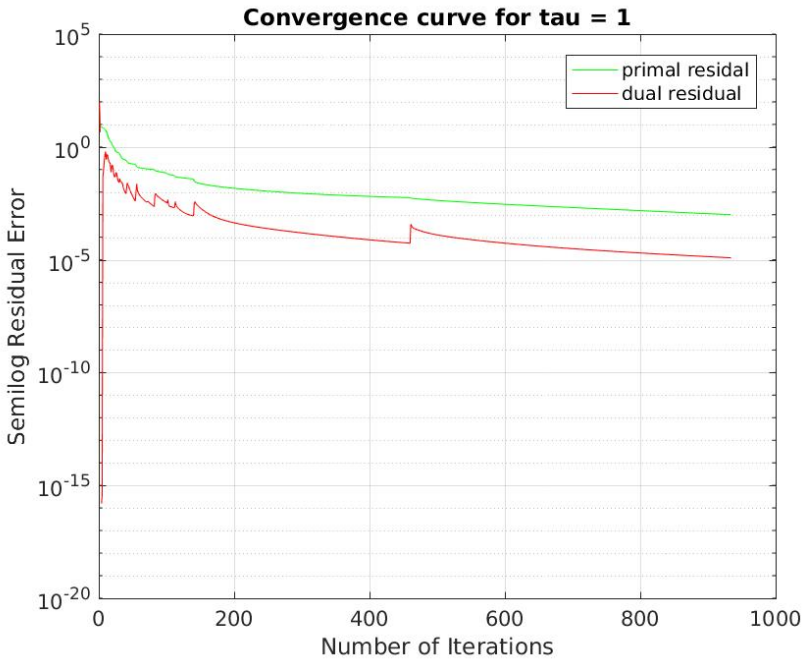

Figure 1: Unwrapped ADMM residual plot

2. Question 2

   (a) Which Code?

Solution: According to wall clock time, SVM does tend to work faster than ADMM for 100 data vectors with 20 features each.

Additionally, for $C = 10$, $\epsilon = 1e^{-3}$, $\tau = 1$, $N = 100$ and $Nfeatures = 20$, I get an accuracy of around 85% with unwrapped ADMM and around 80% accuracy with MATLAB's SVM implementation. Hence, using unwrapped ADMM makes more sense.

I would in any case rather prefer the ADMM simply since ADMM is relatively new with a fun learning curve and allows for improvements to future editions of the algorithm.