**Rapport Machine Learning Julien MOLINIER**

Régression Logistique :

Tout d'abord on remarque que le temps d'exécution du programme de régression logistique est nettement plus court que le programme avec le boosting.

Avec les paramètres par défaut : (sur 1 cœur CPU et L2 penalty)

- Si on prend les classes 4 et 8 on obtient la matrice de confusion suivante :

$$\begin{pmatrix} 978 & 4 \\ 12 & 962 \end{pmatrix}$$

Accuracy : 0.9918

- Si on prend les classes 0 et 1 on obtient la matrice suivante :

$$\begin{pmatrix} 979 & 1 \\ 0 & 1135 \end{pmatrix}$$

Accuracy : 0.9995

On remarque ici que le programme ne se trompe qu'une seule fois. On peut expliquer ceci par la grande différence entre les 2 classes qui permet au programme de bien les différencier lors de l'apprentissage.

- Si on prend les classes 4 et 9 :

$$\begin{pmatrix} 951 & 31 \\ 41 & 968 \end{pmatrix}$$

Accuracy : 0.9638

Ici on voit que les erreurs sont plus nombreuses (41+31) car les classes 4 et 9 sont bien plus ressemblantes.

Boosting avec des filtres de Haar :

Par souci de temps d'exécution long le programme ne prend que les classes 4 et 8 et ne travaille que sur la moitié de la base de données des images mnist. De plus on ne garde que les filtres de Haar type-2-x et type-2-y. Ensuite on ne conserve que ceux donc la diagonale est inférieure à la racine carrée de 30. Pour finir on ne conserve que 1/8 des filtres. Finalement on obtient 15446 filtres.

Concernant le classifieur AdaBoost, les paramètres sont : n_estimators=50, learning_rate=1.0, algorithm='SAMME.R'.

On obtient, en 30 minutes environ, la matrice de confusion suivante :

$$\begin{pmatrix} 499 & 1 \\ 3 & 475 \end{pmatrix}$$

Accuracy : 0.9959