Exercice de programmation 2 : Classification (Régression logistique)

Dans cet exercice, vous allez mettre en œuvre la régression logistique et l'appliquer à un ensemble de données.

Logistic Regression

Dans cette partie de l'exercice, vous allez construire un modèle de régression logistique pour prédire si un étudiant sera admis dans une université.

Supposons que vous soyez l'administrateur d'un département universitaire et que vous souhaitiez déterminer les chances d'admission de chaque candidat en fonction de ses résultats à deux examens. Vous disposez de données historiques sur les candidats précédents que vous pouvez utiliser comme ensemble d'apprentissage pour la régression logistique. Pour chaque exemple d'entraînement, vous disposez des résultats du candidat à deux examens et de la décision d'admission.

Votre tâche consiste à construire un modèle de classification qui estime la probabilité d'admission d'un candidat en fonction des notes obtenues à ces deux examens.

Avant de commencer à mettre en œuvre un algorithme d'apprentissage, il est toujours bon de de visualiser les données si possible.

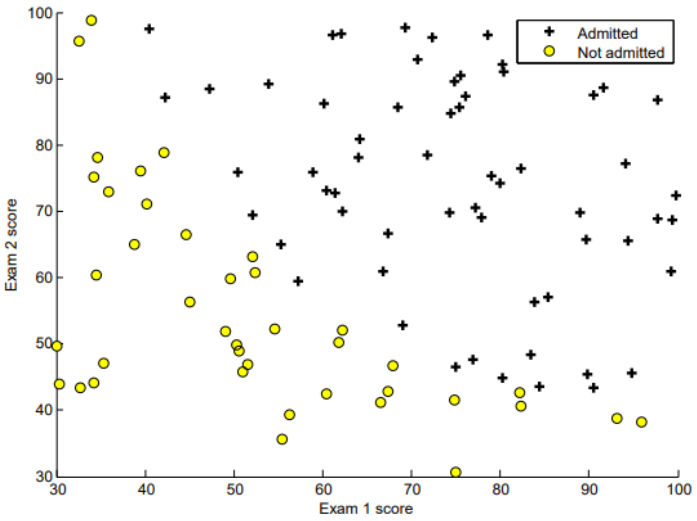

Figure 1: Scatter plot of training data

Implementation

Avant de commencer avec la fonction de coût réelle, rappelez-vous que l'hypothèse de régression logistique est définie comme suit :

$$h_\theta(x) = g(\theta^T x),$$

où la fonction g est la fonction sigmoïde. La fonction sigmoïde est définie comme suit :

$$g(z) = \frac{1}{1 + e^{-z}}.$$

Votre première étape consiste à mettre en œuvre cette fonction. Tester cette fonction avec des valeurs différentes. Pour les grandes valeurs positives de x, la sigmoïde devrait être proche de 1, tandis que pour les grandes valeurs négatives, la sigmoïde devrait être proche de 0. L'évaluation de sigmoïde(0) devrait vous donner exactement 0,5.

Vous allez maintenant implémenter la fonction de coût et le gradient pour la régression logistique.

Rappelez-vous que la fonction de coût dans la régression logistique est

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[-y^{(i)} \log(h_\theta(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right],$$

et le gradient du coût est un vecteur de même longueur que θ dont le jème élément (pour j = 0, 1, . . . , n) est défini comme suit

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Notez que si ce gradient semble identique au gradient de la régression linéaire, la formule est en fait différente car la régression linéaire et la régression logistique ont des définitions différentes de hθ(x).

Nous initialisons les paramètres initiaux à 0 et le taux d'apprentissage alpha à 0,01.

Lorsque vous effectuez une descente de gradient pour apprendre à minimiser la fonction de coût J(θ), il est utile de surveiller la convergence en calculant le coût. Dans cette section, vous allez implémenter une fonction pour calculer J(θ) afin de pouvoir vérifier la convergence de votre implémentation de la descente par gradient.

Appelez votre fonction de coût en utilisant les paramètres optimaux de θ. Vous devriez voir que le coût est d'environ 0.203.

Cette valeur θ finale sera ensuite utilisée pour tracer la frontière de décision sur les données de formation, ce qui donnera une figure similaire à la figure 2.

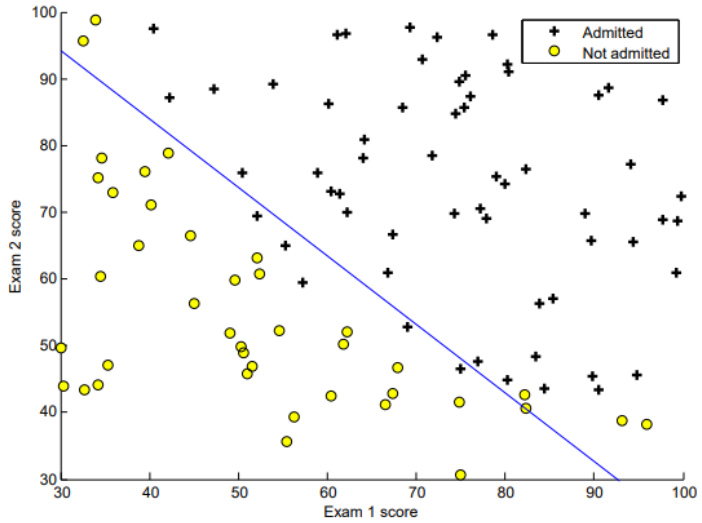

Figure 2: Training data with decision boundary

Après avoir appris les paramètres, vous pouvez utiliser le modèle pour prédire si un étudiant particulier sera admis. Pour un étudiant dont la note à l'examen 1 est de 45 et la note à l'examen 2 est de 85, vous devez vous attendre à une probabilité d'admission de 0,776.