



ΕΘΝΙΚΟ ΚΑΙ
ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

Τεχνητή Νοημοσύνη **2**

1η Εργασία

Ιωάννης Δαλιάνης 1115201700027

Τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ
Οκτώβριος 2020

Περιεχόμενα

	Σελίδα
1 Θεωρητική Άσκηση	2
2 Προγραμματιστική Άσκηση	4
3 Προγραμματιστική Άσκηση	5

1 Θεωρητική Άσκηση

Θα υπολογιστούν τα partial derivatives της ridge regression loss function:

$$J(\mathbf{w}) = MSE(\mathbf{w}) + a \frac{1}{2} \sum_{i=1}^n w_i^2 (\Sigma \chi.1)$$

Θέλουμε δηλαδή να υπολογίσουμε το gradient της Σχέσης 1 :

$$\nabla J(\mathbf{w})$$

Αρχικά υπολογίζουμε τα partial derivatives κάθε μέλους της πρόσθεσης στη Σχέση 1. Όσον αφορά το $MSE(\mathbf{w})$, γνωρίζουμε από τη θεωρία ότι:

$$\frac{\partial}{\partial w_j} MSE(\mathbf{w}) = \frac{2}{m} \sum_{i=1}^m (\mathbf{w} \cdot \mathbf{x}^{(i)} - \mathbf{y}^{(i)}) x_j^{(i)}$$

Έτσι, το gradient, αν υπολογίσουμε κάθε partial derivative ως προς w παίρνουμε:

$$\nabla_w MSE(\mathbf{w}) = \frac{2}{m} (\mathbf{X}^T (\mathbf{X} \mathbf{w} - \mathbf{y})) (\Sigma \chi.2)$$

Για το δεύτερο μέλος της Σχέσης 1:

$$\frac{\partial}{\partial w_j} (a \frac{1}{2} \sum_{i=1}^n w_i^2) = a \frac{1}{2} \cdot \frac{\partial}{\partial w_j} (\sum_{i=1}^n w_i^2) = a \frac{1}{2} \cdot \sum_{i=1}^n \frac{\partial w_i^2}{\partial w_j}$$

Συνεπώς το **gradient** του δεύτερου μέλους είναι:

$$[a \frac{1}{2} \cdot \sum_{i=1}^n \frac{\partial w_i^2}{\partial w_1}, a \frac{1}{2} \cdot \sum_{i=1}^n \frac{\partial w_i^2}{\partial w_2}, \dots, a \frac{1}{2} \cdot \sum_{i=1}^n \frac{\partial w_i^2}{\partial w_n}] (\Sigma\chi.3)$$

Έστω ότι ο αριθμός των **features** είναι $n=2$, Τότε το **Regularization term** θα είναι:

$$a \frac{1}{2} \cdot \sum_{i=1}^n w_i^2 = a \frac{1}{2} \cdot (w_1^2 + w_2^2)$$

Ενδεικτικά υπολογίζουμε το **partial derivative** $\frac{\partial}{\partial w_1}$:

$$a \frac{1}{2} \cdot \frac{\partial}{\partial w_1} (w_1^2 + w_2^2) = a \frac{1}{2} \cdot \left(\frac{\partial w_1^2}{\partial w_1} + \frac{\partial w_2^2}{\partial w_1} \right) = a \frac{1}{2} \cdot 2 \cdot w_1 + 0 = a \cdot w_1$$

Έτσι η Σχέση 3 γίνεται:

$$[a \cdot w_1, a \cdot w_2, \dots, a \cdot w_n] = a \cdot \mathbf{w} (\Sigma\chi.4)$$

Έτσι από τις Σχέσεις 2 και 4 έχουμε το **gradient** της **ridge regression loss function**:

$$\nabla J(\mathbf{w}) = \frac{2}{m} \cdot (\mathbf{X}^T \cdot (\mathbf{X}\mathbf{w} - \mathbf{y})) + a \cdot \mathbf{w}$$

2 Προγραμματιστική Άσκηση

Τα αρχεία για την εκτέλεση των αλγορίθμων γίνονται **mount** μέσω του **google drive**. Πετάμε τη στήλη **'Unnamed: 0'**, καθώς δε περιέχει καμία χρήσιμη πληροφορία. Η συνάρτηση **scaleData** παίρνει σαν ορίσματα μία συνάρτηση που εφαρμόζει κάποιου τύπου **scaling** και κάποια δεδομένα πάνω στα οποία την εφαρμόζει. Η συνάρτηση **MakeSets** χωρίζει τα δεδομένα σε **train** και **set**, κάνοντας και **encode** στα **labels**. Εφαρμόζεται στα δεδομένα πρώτα η **Ridge Regression** από την **sklearn** βιβλιοθήκη και εμφανίζονται τα αποτελέσματα. Οι συναρτήσεις για **gradient descent** Υλοποιούνται σε μία κλάση που διαχωρίζει τις περιπτώσεις ανάλογα με τα ορίσματα που δίνονται (**None** ή **'GD'**, **'Mini Batch'**, **'SGD'**). Εφαρμόζονται **plot** και **score functions** για να ελεγχθεί η αποδοτικότητα των αλγορίθμων. Το **R2 Score** χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός μοντέλου γραμμικής παλινδρόμησης. Μετράει την απόκλιση ανέμεσα στα **actual** και τα **predicted** δεδομένα. Τέλειο σκορ είναι το **1.0** αν τα δεδομένα είναι ακριβώς ίδια, ενώ μπορεί να είναι και αρνητικό αν το μοντέλο δε κάνει ιδιαίτερα καλή δουλειά στην πρόβλεψη των **labels**. Από τα **scores** φαίνεται ότι καλύτερη επίδοση πάνω στα συγκεκριμένα δεδομένα έχει το **mini batch gradient descent**. Κάποιες συναρτήσεις-μέλη στην κλάση των **descents** φτιάχτηκαν για να γίνει προσπάθεια να χρησιμοποιηθεί η κλάση σα μέλος στη συνάρτηση **learning_curve** της **sklearn**. Για τον έλεγχο **overfitting underfitting** χρησιμοποιήθηκε μέθοδος από: εδώ. Σε αυτές τις καμπύλες, **Overfitting** έχουμε όταν το **error** στα **training** δεδομένα είναι πολύ χαμηλό, αλλά τότε το **error** στα **test** δεδομένα είναι πολύ μεγάλο. **Underfitting** έχουμε όταν το **error** και στα **training** και στα **test** δεδομένα είναι πολύ μεγάλο.

3 Προγραμματιστική Άσκηση

Από το `dataset` διαγράφονται οι στήλες `'flag'`, `'date'`, `'user'`, `'Unnamed: 0'`, `'id'` καθώς δε χρησιμεύουν για την αξιολόγηση των Tweets. Με τη συνάρτηση `clean-Text` καθαρίζουμε τα tweets από urls, από μεγάλα κενά, από usernames(@) κ.λ.π. Αφαιρούνται και όλοι οι αριθμοί καθώς η ύπαρξη αριθμών σε ένα tweet σε καμία περίπτωση δε παίζει ρόλο στο ύφος του tweet. Αφαιρούνται επίσης και τα `stop-words`. Τα text δεδομένα των tweets μετατρέπονται σε **Bag of Words** αριθμητικά δεδομένα για να μπορούν να χρησιμοποιηθούν από τις συναρτήσεις. Χρησιμοποιείται η `logisticRegr` της `sklearn` βιβλιοθήκης, τα ορίσματα της οποίας καθορίζονται από τη μέθοδο **Grid Search** πάνω σε ένα δείγμα από τα δεδομένα. Με τα **Best parameters** που αξιολογούνται πετυχαίνουμε **Precision Score**, **Recall Score** και **F-Measure Score** της τάξης του 74.6. Στο **Grid Search** κάποιες παράμετροι δε δίνονται διότι ορισμένες δε γίνεται να δίνονται ταυτόχρονα με κάποιες άλλες, π.χ. το `l1 penalty` με ορισμένες παραμέτρους `solver`.