Project Algorithms 3

Σκορδούλης Κωνσταντίνος 1115 2016 00155

Main Programs: predict.py new_representation.py

Compilation: Υπάρχει διαθέσιμο **Makefile**, οπότε με την εντολή **make**, γίνεται γρήγορα το compilation (για το ./cluster της εργασίας 2)

ΣΗΜΑΝΤΙΚΟ: Τα προγράμματα υλοποιήθηκαν σε Python ➔ **(Python3.7.2)**. Χρησιμοποιήθηκαν οι βιβλιοθήκες **keras, panda, numpy**.

ΣΗΜΑΝΤΙΚΟ2: Παραθέτω και 2 source files, **cluster.cpp**(main) και **Update.cpp**(source file). Και τα 2 αφορούν κώδικα της 2^{ης} εργασίας.

- Στο **cluster.cpp**, άλλαξα πως διαβάζει το input , και αφαίρεσα έλεγχο για διπλότυπα διανύσματα (διανύσματα που έχουν ίδιες συντεταγμένες).
- Τώρα για το **Update.cpp**, απλά μου ξέφυγε μια **assert()**, για το k (number of clusters)**.**
- Δεν χρειάζεται κάτι, απλά **copy paste** στο φάκελο εργασία2.

Predict.py:

Καταρχάς διαβάζω το αρχείο που μου δίνεται αγνοώντας την 1^{η} στήλη(timestamp). Διαβάζω και το **actual.csv** (οι πραγματικές τιμές/πραγματικά διανύσματα). Τέλος υπολογίζω τα σφάλματα **MAE**, **MAPE**, **MSE** και γράφω τα αποτελέσματα στο **predicted.csv**. Κάποιες παρατηρήσεις:

1. Στους τύπους, υπολογίζουμε **actual_value – predicted_value**. **Value** δεν θεωρώ τις τιμές των συντεταγμένων, αλλά τα **διανύσματα**. Οπότε για να υπολογίσω σωστά το παραπάνω, χρησιμοποιώ **Manhattan_Distance()**.
2. Τώρα για το **MAPE,** όπου έχουμε (**actual_value – predicted_value**) / **actual_value,** προκύπτει (αριθμός/διάνυμσα) != αριθμός. Μπορούμε όπως να θεωρήσουμε ότι στο παρανομαστή έχουμε **actual_value – 0 ➔ πραγματικό – μηδενικό διάνυσμα**. Γρήγορα βλέπουμε ότι ουσιαστικά έχουμε **sum(actual_value.coordinates)**. Τελικά για το MAPE έχουμε **Manhattan_Distance(actual,predicted)/ sum(actual.coordinates)**

New Representation.py

Καταρχάς διαβάζω το input και φορτώνω το μοντέλο που μας δίνεται. Το μοντέλο που θέλω να δημιουργήσω θα έχει **1 μόνο layer:**

- **Dense(64, input_shape=(128,), activation='softmax')**.
- Δηλαδή παίρνει σαν είσοδο διάνυσμα με 128 συντεταγμένες και βγάζει ως output διάνυσμα **με 64 συντεταγμένες.**

- Επειδή είναι το τελευταίο layer, χρησιμοποιώ **activation function** και συγκεκριμένα την **softmax ➔ classification into multiple classes**. Δηλαδή μας εξυπηρετεί γιατί θα χρειαστούμε να κάνουμε **clustering** πάνω σε αυτά.
- Φορτώνω τα βάρη (1ου layer) του pre-trained μοντέλου μας, στο καινούργιο μου μοντέλο και εκτελώ την predict.
- Τέλος φτιάχνω το **new_representation.csv,** εισάγοντας τα αποτελέσματα μου (από το predict()) και προσθέτοντας και τα **timestamps** (ως item_id).

Σχολιασμός Αποτελεσμάτων

Καταρχάς ο πιο αποτελεσματικός αλγόριθμος για **clustering** (με βάση το **Silhouette**) είναι ο **211** ή καλύτερα **Initialization++ ➔ Lloyd's Assignment ➔ PAM ala Lloyd** (για in-depth analysis, δες προηγούμενο pdf).

Σύγκρινα το clustering των δεδομένων μου με **d = 128** (nn_representation.csv) και **d = 64** (new_representation.csv) όπου d-> dimension, τόσο για **k = 4** όσο και για **k = 12** (k = πλήθος clusters).

Παρατήρησα λοιπόν, για k = 4 και για k = 12, το **d = 64** είχε καλύτερα αποτελέσματα Silhouette (**stotal)**, από ότι για d = 128.

Αυτό το αποτέλεσμα είναι λογικό για τους εξής λόγους:

- Σκοπός του προβλήματος μας είναι να **κωδικοποιήσουμε/συμπτύξουμε** την πληροφορία/διάνυσμα μας, για να αντιμετωπίσουμε το **<u>Curse of Dimensionality</u>**.
- Με απλά λόγια:
  - Όσες περισσότερες διαστάσεις έχουν τα αντικείμενα μας, τόσο πιο <u>**αραιή(sparse)** είναι η κατανομή τους στο χώρο</u>.
  - Επιπλέον, τα αντικείμενα γίνονται <u>περισσότερο **διακριτά**</u> (καθώς έχουν περισσότερα χαρακτηριστικά που να τα διαφοροποιούν μεταξύ τους).
  - Άρα <u>**δεν μπορεί να γίνει ικανοποιητικό clustering**</u> με τους συνηθισμένους τρόπους, για αυτό προσπαθούμε να κωδικοποιήσουμε/μικρύνουμε τη διάσταση της πληροφορίας.
- Επομένως επιδιώκουμε μείωση/κωδικοποίηση των διαστάσεων μας ➔ καλύτερο Clustering
- <u>**Τα αποτελέσματα μας συμβαδίζουν με τη θεωρία!**</u>

Τέλος, είχα <u>καλύτερο Silhouette(**stotal**) με (**k=4**)</u> συγκριτικά με (k=12) , όπου k το πλήθος των clusters.