REPORT N. 5

Diploma Thesis

EFSTRATIOS KARKANIS
Π19064

Δεδομένα

Τα δεδομένα που χρησιμοποιήθηκαν για την έρευνα προέρχονται από ένα project που ονομάζεται cabspotting project. Συγκεκριμένα, σε αυτό το σύνολο δεδομένων περιέχονται περίπου 11.000.000 δεδομένα GPS διαφόρων taxi (Yellow Cab Vehicle) στην περιοχή του San Francisco, California. Η όλη δειγματοληψία των δεδομένων συνέβη τον Μάιο του 2008.

Αξίζει να σημειωθεί ότι το dataset αποτελείται από πολλά αρχεία. Κάθε ένα εξ' αυτών περιλαμβάνει την τροχιά ενός taxi. Ωστόσο, για να παράγουμε πιο αξιόπιστα αποτελέσματα, οι τροχιές αυτές έχουν «σπάσει» με βάση τον χρόνο. Δηλαδή, εάν υπάρχουν διαδοχικά σημεία εντός της ίδιας τροχιάς που απέχουν μεταξύ τους χρονική απόσταση παραπάνω από 90 δευτερόλεπτα, τότε η τροχιά σπάει σε υποτροχιές. Αυτή η διαδικασία γίνεται για όλες τις τροχιές, μέχρι τελικά όλες οι υποτροχιές που θα παραχτούν να πληρούν την παραπάνω συνθήκη. Τέλος, ο λόγος που επιλέγουμε σαν ανώτατο χρονικό όριο τα 90 δευτερόλεπτα είναι ότι το Valhalla API που εκτελεί την διαδικασία Map Matching, ακόμα και τότε παράγει αξιόπιστα αποτελέσματα, ενώ επίσης η πλειοψηφία των δεδομένων μεταξύ τους είχαν απόκλιση 90 δευτερόλεπτα.

Για τους σκοπούς της έρευνάς μας χρησιμοποιούμε τα δεδομένα μίας μόνο εβδομάδας (περίπου 3.300.000 εγγραφές), καθώς θεωρείται ότι για την πρόβλεψη κυκλοφοριακής ροής σε οδικά δίκτυα δεν κρίνονται σημαντικά τα δεδομένα που βρίσκονται πολύ πίσω στο χρόνο, παρά μόνο τα πιο πρόσφατα. Μπορούμε να υποθέσουμε λοιπόν ότι στο San Francisco, τα δεδομένα της μίας εβδομάδας που επιλέχθηκαν είναι τα πλέον πρόσφατα δεδομένα κίνησης taxi και με βάση αυτά μπορούμε να κάνουμε μία πρόβλεψη κυκλοφοριακής ροής για το μέλλον.

Διαδικασία Map Matching

Για τους σκοπούς του Map Matching, χρησιμοποιήθηκε το Valhalla API, ένα Open Source λογισμικό που επιτρέπει διάφορες ενέργειες πάνω σε γεωγραφικά δεδομένα. Συγκεκριμένα, για την έρευνά μας χρησιμοποιήθηκε το endpoint **/trace_attributes** του API. Η συγκεκριμένη υπηρεσία δοθέντος των GPS συντεταγμένων (διατεταγμένες ως προς το χρόνο) μίας ολόκληρης τροχιάς, επιστρέφει το ακριβές μονοπάτι που ακολουθήθηκε από την συγκεκριμένη τροχιά. Πιο αναλυτικά, το μονοπάτι είναι μία σειρά από OSM IDs. Τα αναγνωριστικά αυτά δεν είναι τυχαία, αλλά ανακτώνται από την επίσημη βάση που εμπεριέχει τα Open Street Map (OSM) δεδομένα. Αυτό σημαίνει ότι υπάρχει δυνατότητα γραφικής απεικόνισης του μονοπατιού που εκτέλεσε η τροχιά πάνω στο χάρτη.

Το μόνο αρνητικό αυτής της διαδικασίας Map Matching είναι ο χρόνος. Συγκεκριμένα, για 3.300.000 εκατομμύρια δεδομένα GPS, χρειάστηκαν περίπου 12 ώρες για να ολοκληρωθεί η διαδικασία. Προφανώς, ο χρόνος αυτός εξαρτάται και από το Hardware του μηχανήματος, όπως επίσης και από την ποιότητα σύνδεσης στο διαδίκτυο.

Κώδικας Strict Path Query (SPQ)

Ένα από τα πιο σημαντικά κομμάτια αυτής της πτυχιακής εργασίας είναι και η υλοποίηση της συνάρτησης που θα υλοποιεί τα SPQ queries. Υπενθυμίζεται ότι ένα τέτοιο query θα βρίσκει όλες τις τροχιές που ακολουθούν επακριβώς ένα συγκεκριμένο μονοπάτι π οποιουδήποτε μήκους και εντός ενός συγκεκριμένου χρονικού διαστήματος. Η έναρξη αυτού του διαστήματος υποδηλώνει τον χρόνο που η τροχιά εισέρχεται στο συγκεκριμένο μονοπάτι, ενώ η λήξη του διαστήματος υποδηλώνει τον ανεκτό χρόνο που η εκάστοτε τροχιά θα πρέπει να έχει εξέλθει από αυτό.

Πρέπει να σημειωθεί ότι όλος ο κώδικας έχει γραφεί σε python, ενώ η συνολική διάρκεια εκτέλεσης ενός οποιουδήποτε ερωτήματος SPQ διαρκεί περίπου 300 – 500 ms . Οι συγκεκριμένοι χρόνοι είναι ρεαλιστικοί και προκύπτουν από ένα dataset με μέγεθος 3.300.000 εγγραφών. Με άλλα λόγια, η συγκεκριμένη χρονική απόδοση είναι πάρα πολύ ικανοποιητική! Η ταχύτητα εκτέλεσης ενός SPQ σε python είναι **κατά πολύ μεγαλύτερη από την ταχύτητα εκτέλεσης των SPQs που έχουν υλοποιηθεί σε προηγούμενη έρευνα**. Επομένως, δεν κρίνεται απαραίτητή η χρήση μίας βάσης PostgreSQL για την παρούσα φάση.

Ακολουθεί τώρα ο ψευδοκώδικας που περιγράφει την συνάρτηση SPQ:

ΑΛΓΟΡΙΘΜΟΣ SPQ

Είσοδος: ένα μονοπάτι π (1), χρόνος time_enter (2), χρόνος time_leave (3).

Επιπλέον, η συνάρτηση προσπελάβνει κάθε φορά έναν πίνακα visited_segments που περιέχει τα δεδομένα. Ο συγκεκριμένος πίνακας περιέχει την ακόλουθη πληροφορία:

- Taxi ID: το αναγνωριστικό της τροχιάς
- Traj ID: το αναγνωριστικό της υποτροχιάς της τροχιάς Taxi ID
- OSM Way ID: το αναγνωριστικό της ακμής
- Start Time: χρόνος που το όχημα εισήλθε στην ακμή
- End Time: χρόνος που το όχημα εξήλθε από την ακμή

0. Υπολόγισε το μήκος του μονοπατιού π και αποθήκευσέ το σε μία μεταβλητή path_length
1. Φτιάξε μία κενή λίστα trajectories
2. Βρες όλες τις εγγραφές του πίνακα visited_segments που έχουν Start Time >= time_enter και End Time <= time_leave, και αποθήκευσέ το σε μία μεταβλητή με όνομα examined_data.
3. Βρες όλα τα αναγνωριστικά των γραμμών που περιέχουν σαν OSM Way ID την πρώτη ακμή στο μονοπάτι π και αποθήκευσέ τα σε μία λίστα needed_indexes.
4. Για κάθε στοιχείο index στη λίστα needed_indexes επανέλαβε:
   4.1 Βρες το Taxi ID του στοιχείου index και αποθήκευσέ το σε μία μεταβλητή taxi_id
   4.2 Βρες το Traj ID του στοιχείου index και αποθήκευσέ το σε μία μεταβλητή traj_id
   4.3 Όρισε την τιμής μίας νέας μεταβλητής inter = 1
   4.4 Από I = 1 έως path_length επανέλαβε:

4.4.1 Έλεγξε εάν η γραμμή με αναγνωριστικό index+i περιέχει σαν Taxi ID == taxi_id ΚΑΙ Traj ID == traj_id ΚΑΙ OSM Way ID την επόμενη σε σειρά ακμή στο μονοπάτι π.

4.4.2 Εάν ισχύει η παραπάνω συνθήκη αύξησε τον μετρητή inter κατά 1

4.5 Τέλος εσωτερικού βρόγχου

4.6 Εάν path_length == inter, πρόσθεσε το ζευγάρι (taxi_id,traj_id) στη λίστα trajectories

4.7 Τέλος εξωτερικού βρόγχου

5. Διέγραψε τα διπλότυπα ζευγάρια (εάν υπάρχουν) από τη λίστα trajectories και επέστρεψε το μήκος της

ΤΕΛΟΣ ΑΛΓΟΡΙΘΜΟΥ

Δημιουργία Time Series Dataset

Το τελικό βήμα αυτής της φάσης είναι η κατασκευή του time series dataframe. Οι στήλες του dataset θα αντιπροσωπεύουν τον χρόνο, ενώ οι γραμμές θα αντιπροσωπεύουν διάφορα μονοπάτια με μήκος >=2.

Όσον αφορά τον χρόνο, εφόσον γίνεται χρήση δεδομένων τροχιών μίας εβδομάδας, μπορούμε να διασπάσουμε αυτό το διάστημα ανά ώρα, δημιουργώντας έτσι 24*7 - 1 στήλες. Κάθε στήλη θα περιλαμβάνει ουσιαστικά ένα διάστημα μίας ώρας (π.χ 2008-05-23 00:01:00 – 2008-05-23 00:02:00).

Από την άλλη, για το θέμα με τα μονοπάτια, μπορούμε να παράγουμε τυχαία μονοπάτια αποτελούμενα από διαδοχικές ακμές OSM Way Id. Εφόσον η διαδικασία Map Matching επιστρέφει το ακριβές μονοπάτι που ακολούθησε η τροχιά, τότε η εύρεση τυχαίων συνεχόμενων μονοπατιών τυχαίου μήκους μπορεί να γίνει απευθείας από τα Map Matched δεδομένα.

Τέλος, όσο περισσότερα μονοπάτια χρησιμοποιήσουμε, τόσο περισσότερη πιθανότητα υπάρχει να καλυφθεί ολόκληρο το οδικό δίκτυο. Εάν επιλέξουμε τελικά να χρησιμοποιήσουμε n στο πλήθος μονοπάτια, τότε θα χρειαστούν n*(24*7 – 1) κλήσεις της συνάρτησης SPQ. Συγκεκριμένα, για 504 μονοπάτια χρειάστηκε συνολικός χρόνος εκτέλεσης 20 λεπτών, δηλαδή 504*(24*7 – 1) κλήσεις.