MEM704 - Μηχανική Μάθηση
1η Εργαστηριακή Άσκηση
Ο αλγόριθμος PageRank και η ανάλυση SVD

Παράδοση: 7/03/2024, 18:00,
Εξέταση: 8/3/2024 στο Εργαστήριο

Στη άσκηση αυτή θα υλοποιήσουμε τον αλγόριθμο PageRank για την εκτίμηση της σημαντικότητας ιστοσελίδων στο διαδίκτυο. Στην κέντρο του αλγορίθμου βρίσκεται η μέθοδος των δυνάμεων για την εύρεση της κυρίαρχης ιδιοτιμής ενός κατάλληλου πίνακα. Επίσης θα χρησιμοποιήσουμε την ανάλυσης SVD για την ανακατασκευή μιας εικόνας.

1 Μέθοδος των δυνάμεων

Έστω $A \in \mathbb{R}^{n,n}$. Η μέθοδος των δυνάμεων για την προσέγγιση της κυρίαρχης ιδιοτιμής λ και του αντίστοιχου κυρίαρχου ιδιοδιανύσματος x, δίνεται από τον παρακάτω ψευδο-αλγόριθμο

Μέθοδος των δυνάμεων για $Ax = \lambda x$

Διαλέγουμε τυχαίο $x_0 \in \mathbb{R}^n$, $x_0 = x_0/\|x_0\|_1$
Διαλέγουμε k_{max} - μέγιστο αριθμό επαναλήψεων
Διαλέγουμε $\epsilon \sim 10^{-6}$ - σφάλμα
$k = 0,\ d_k = 1$
While $d_k > \epsilon$ and $k < k_{max}$ do
  $x_k = Ax_{k-1}$
  $x_k = x_k/\|x_k\|$
  $d_k = \|x_k - x_{k-1}\|$
$\lambda = \frac{x_k^T A x_k}{x_k^T x_k}$

Να γραφτεί ένας κώδικας Python που να υλοποιεί τον παραπάνω αλγόριθμο. Μπορείται να χρησιμοποιήσεται την βιβλιοθήκη Numpy και ειδικότερα το πακέτο linalg. Για την επιλογή του x_0 μπορείται να χρησιμοποιήσεται το πακέτο random της Numpy. Ως νόρμα μπορείτε να χρησιμοποιήσετε μια εκ των $\|\cdot\|_1,\ \|\cdot\|_2,\ \|\cdot\|_\infty$. Δοκιμάστε τον κώδικά σας με ένα απλό ($n \geq 3$) πίνακα για τον οποίο ξέρετε όλες τις ιδιοτιμές και τα αντίστοιχα ιδιοδιανύσματα του. Επαληθεύεστε αριθμητικά την θεωρητική ταχύτητα σύγκλισης της μεθόδου συγκρίνοντας το λόγο $|\lambda_2/\lambda_1|$ με τους διαδοχικούς λόγους d_{k+1}/d_k. Στη συνέχεια, δοκιμάστε τον κωδικά σας για τον τριδιαγώνιο πίνακα $T = triad[-1, 2, -1]$ (2 στη κύρια διαγώνιο και -1 στη 1η υπερδιαγώνιο και 1η υποδιαγώνιο), για διάφορες τιμές του n. Ο T είναι θετικά ορισμένος και οι ιδιοτιμές του είναι $\lambda_k = 2 - 2\cos\left(\frac{k\pi}{n+1}\right)\ k = 1, \ldots n$.

2 Ο αλγόριθμος PageRank

Ο αλγόριθμος αυτός αποτελεί τον πυρήνα της μηχανής αναζήτησης Google και δίνει μια εκτίμηση για το πόσο *σημαντική* είναι μια σελίδα του διαδικτύου. Προτάθηκε από τους S. Brin και L.Page το 1998, στην εργασία με τίτλο : *The anatomy of a large-scale hypertextual Web search engine*, Computer Networks and ISDN Systems. 30 (1–7): 107–117. Η μέθοδος βασίζεται στη εύρεση του κυρίαρχου ιδιοδιανύσματος ενός πίνακα τύπου Markov. Οι πίνακες Markov είναι τετραγωνικοί, τα στοιχεία τους εκφράζουν πιθανότητες και το άθροισμα των στοιχείων κάθε στήλης είναι 1. Τα χαρακτηριστικά αυτά έχουν σαν αποτέλεσμα το 1 να είναι η μεγαλύτερη ιδιοτιμή με πολλαπλότητα 1 και τα στοιχεία του αντίστοιχου ιδιοδιανύσματος αθροίζουν σε 1. Οι υπόλοιπες ιδιοτιμές είναι σε απόλυτη τιμή μικρότερες του 1.

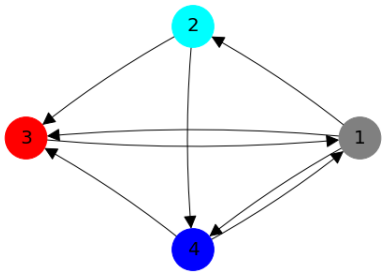
Έστω ότι έχουμε N ιστοσελίδες μεταξύ των οποίων υπάρχουν πολλαπλοί σύνδεσμοι. Ο πίνακας Markov της μηχανής αναζήτησης έχει την μορφή

$$M = d \cdot A + \frac{1-d}{N} B, \quad B = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

όπου $d \sim 0.85$, N ο αριθμός ιστοσελίδων και A ο πίνακας με στοιχεία a_{ij}

$$a_{ij} = \begin{cases} \frac{1}{L(j)} & \text{αν υπάρχει σύνδεσμος από την σελίδα } j \text{ στη σελίδα } i \\ 0 & \text{διαφορετικά} \end{cases}$$

όπου $L(j)$ είναι όλοι οι εξερχόμενοι σύνδεσμοι από την σελίδα j. Για παράδειγμα, θεωρούμε τον κατευθυνόμενο γράφο(Directed Graph) της εικόνας, κάθε κόμβος του οποίου αντιπροσωπεύει μια ιστοσελίδα με τους αντίστοιχους συνδέσμους.



και $L(1) = 3,\ L(2) = 2,\ L(3) = 1,\ L(4) = 2$. Ο πίνακας A σε αυτή την περίπτωση είναι

$$A = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

Εφαρμόζοντας την μέθοδο των δυνάμεων στον πίνακα M έχουμε την εξής προσέγγιση του κυρίαρχου ιδιοδιανύσματος και της ιδιοτιμής

$$x \approx (0.387, 0.129, 0.29, 0.193),\ \lambda \approx 0.9999999999656655, \quad \sum_{i=1}^{4} x_i = 1$$

Να γραφτεί ένας κώδικας Python ο οποίος θα διαβάζει τα στοιχεία του γράφου, θα φτιάχνει τον πίνακα M και θα υπολογίζει με την μέθοδο των δυνάμεων το κυρίαρχο ιδιοδιάνυσμα και την αντίστοιχη ιδιοτιμή. Το αρχικό διάνυσμα το διαλέγουμε ως $x_{0,i} = \frac{1}{N},\ i = 1, \ldots, N$.

Η κάθε γραμμή του αρχείου(graph0.txt) που περιγράφει τον γράφο είναι της μορφής: $n_i\ n_j$ πού σημαίνει ότι υπάρχει ένας σύνδεσμος από τον κόμβο n_i στο κόμβο n_j, π.χ το αρχείο που περιγράφει τον παραπάνω γράφο είναι

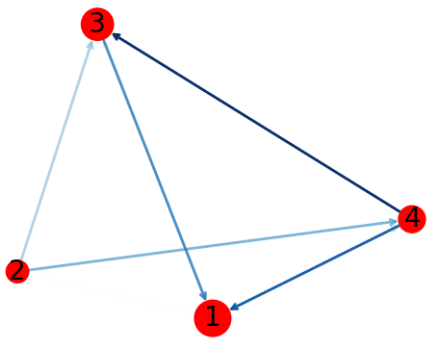
1 2
1 3
1 4
2 3
2 4
3 1
4 1
4 3

Δοκιμάστε τον κώδικα σας με αυτόν το γράφο και αυτούς των αρχείων graph1.txt, graph2.txt και βεβαιωθείτε ότι δουλεύει σωστά ελέγχοντας ότι για το κυρίαρχο ιδιοζεύγος ισχύουν: $\lambda \approx 1,\ \sum_{i=1}^{N} |x_i| \approx 1$. Το προγραμμά σας στο τέλος θα τυπώνει τους κόμβους με φθίνουσα σειρά ως προς την σημαντικότητας τους.

3 Η βιβλιοθήκη NetworkX της Python

Στη Python υπάρχει η βιβλιοθήκη *NetworkX* με την οποία μπορούμε με μεγάλη ευκολία να κατασκευάσουμε, χειριστούμε και επεξεργαστούμε γράφους. Στη βιβλιοθήκη, υπάρχουν επίσης και διάφοροι αλγόριθμοι μεταξύ των οποίων και ο PageRank. Εγκαταστήστε πρώτα την βιβλιοθήκη στο περιβάλλον Python που δουλεύετε. Στη συνέχεια φτιάξτε ένα κατευθυνόμενο γράφο(Directed Graph) G, καλέστε τον αλγόριθμο PageRank: networkx.pagerank(G, d) και συγκρίνεται με τα δικά σας αποτελέσματα.

Με την βιβλιοθήκη αυτή μπορούμε επίσης να κάνουμε εύκολα γραφική αναπαράσταση του γράφου:



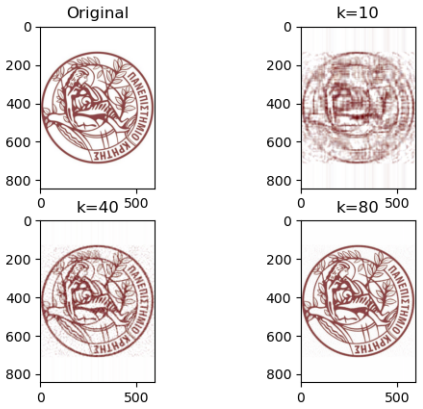
Το μέγεθος των κόμβων του γράφου στη εικόνα είναι ανάλογο της σημαντικότητάς του.

4 Ανάλυση SVD, Ανακατασκευή εικόνας

Δοσμένης μιας εικόνας θα χρησιμοποιήσουμε την ανάλυση σε ιδιάζουσες τιμές(SVD) για να ανακατασκευάσουμε την εικόνα χρησιμοποιώντας όσο το δυνατό μικρότερο όγκο πληροφορίας. Η ιδέα βασίζεται στο ότι αν ένας πίνακας $A \in \mathbb{R}^{m,n}$ έχει ανάλυση SVD $A = U\Sigma V^T$ τότε ο πίνακας A_k με

$$A_k = \sigma_1 u_1 v_1^T + \cdots + \sigma_k u_k v_k^T, \quad k = 1, \ldots, r, \quad r = \text{rank}(A),$$

είναι ο πίνακας τάξης $\text{rank}(A_k) = k$ που έχει την μικρότερη δυνατή απόσταση από τον A ως προς την νόρμα $\|\cdot\|_2$. Να γραφτεί ένας κώδικας Python ο οποίος θα διαβάζει μια εικόνα (A), θα υπολογίζει την ανάλυση SVD και θα κατασκευάζει τον πίνακα A_k για διάφορες τιμές του k. Ένας τρόπος για να εισάγεται την εικόνα στον κωδικά σας είναι μέσω της βιβλιοθήκης Matplotlib και της κλάσης image.imread. Στη συνέχεια μπορείτε να χρησιμοποιήσετε την κλάση linalg.svd της Numpy για το υπολογισμό της ανάλυσης SVD της εικόνας. Για διάφορες τιμές του k υπολογίστε το σφάλμα $\epsilon_k = \|A - A_k\|_2$. Με την χρήση της Matplotlib κάντε το γράφημα της ακολουθίας (k, ϵ_k) και των ιδάζουσων τιμών σ_i. Τέλος σε ένα κοινό γράφημα τοποθετήστε πρώτα την αρχική εικόνα και στη συνέχεια την ανακατασκευασμένη εικόνα A_k για ενδεικτικές τιμές του k, π.χ.



Δοκιμάστε τον κωδικά σας με τις εικόνες $uoc\_logo.png$ και $python\_logo.png$ που σας δίνονται.

5 Παράδοση - Εξέταση

Για κάθε μέρος της άσκησης να φτιάξετε διαφορετικό κώδικα Python με όνομα π.χ: $\{math, tem, ph\}XXXX\_Lab1\{a, b, c, d\}.py$ όπου $XXXX$ είναι ο αριθμός μητρώου σας. Επίσης στις πρώτες γραμμές του κάθε προγράμματος θα υπάρχουν σαν σχόλιο τα στοιχεία σας: όνομα, επώνυμο και ΑΜ. Θα στείλετε την άσκηση μέσω της σελίδας του μαθήματος στο UoC-eLearn με την μορφή ενός αρχείου με όνομα $\{math, tem, ph\}XXXX\_LAB1.zip$ το οποίο θσ περιέχει όλους του κώδικες, το αργότερο μέχρι 18:00, Πέμπτη 7 Μαρτίου. Εκπρόθεσμες ασκήσεις δεν θα βαθμολογηθούν. Εργαστείται ατομικά. Κώδικες που είναι προϊόν αντιγραφής θα μηδενίζονται.