



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΗΜΜΥ. ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ

Εργασία 3

Image segmentation

Συντάκτης:
Χρήστος Χουτουρίδης [8997]
cchoutou@ece.auth.gr

Διδάσκοντες:
Α. Ντελόπουλος
antelopo@ece.auth.gr

Δ. Αλέτρας
lnmpikas@ece.auth.gr

5 Ιουλίου 2025

1. ΕΙΣΑΓΩΓΗ

Η παρούσα εργασία αφορά την υλοποίηση και αξιολόγηση τεχνικών τμηματοποίησης εικόνας (*image segmentation*) με βάση φασματικούς αλγορίθμους (*spectral clustering*) και τη μέθοδο *normalized cuts*. Στόχος είναι η κατανόηση και εφαρμογή γραφοκεντρικών μοντέλων στην επεξεργασία εικόνας, μέσω της μετατροπής της εικόνας σε γράφο και της χρήσης των ιδιοδιανυσμάτων του αντίστοιχου γραφικού Λαπλασιανού για την ανάδειξη δομής και περιοχών ενδιαφέροντος. Η εργασία οργανώνεται σε τρία βασικά μέρη: (α) αναπαράσταση εικόνας ως γράφου, (β) φασματικό ομαδοποίηση με *spectral clustering*, και (γ) τμηματοποίηση εικόνας μέσω της *normalized cuts*, σε μη αναδρομική και αναδρομική μορφή.

Η προσέγγιση βασίζεται στην έννοια του *affinity matrix*, όπου κάθε pixel της εικόνας θεωρείται κόμβος γράφου και οι ακμές μεταξύ των κόμβων αποτυπώνουν την ομοιότητά τους. Οι τεχνικές *spectral clustering* και *normalized cuts* εκμεταλλεύονται τη φασματική ανάλυση του γραφήματος για να επιτύχουν διαχωρισμό των pixels σε ομάδες, με βάση τη δομή συνδεσιμότητας και τα χαρακτηριστικά της εικόνας. Η εργασία καλύπτει την υλοποίηση αυτών των μεθόδων από το μηδέν, τον πειραματισμό πάνω σε προκαθορισμένα και ρεαλιστικά δεδομένα, και την αξιολόγηση των αποτελεσμάτων μέσα από συγκριτικές απεικονίσεις και σχολιασμό.

1.1. Παραδοτέα

Τα παραδοτέα της εργασίας αποτελούνται από:

- Την παρούσα αναφορά.
- Τα python αρχεία με τον κώδικα της εργασίας και το αρχείο `requirements.txt` με τις απαιτήσεις της python στον κατάλογο `scripts/`.
- Το [σύνδεσμο](#) με το αποθετήριο που περιέχει όλο το project με τον κώδικα σε python της αναφοράς και τα παραδοτέα.

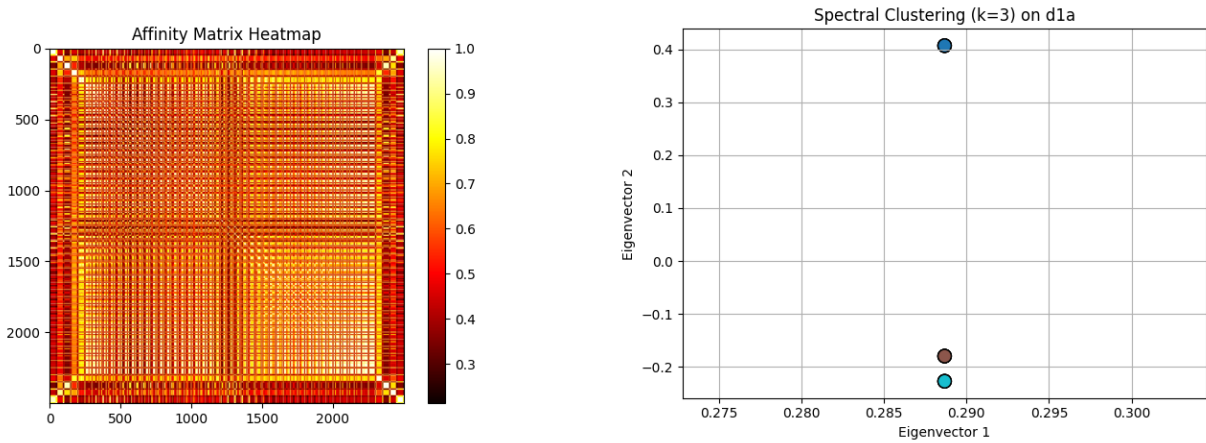
2. ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΟΡΓΑΝΩΣΗ ΚΩΔΙΚΑ

Πριν προχωρήσουμε στην ανάλυση των αλγορίθμων και στην επίδειξη της λειτουργικότητάς τους, αξίζει να περιγράψουμε συνοπτικά την προγραμματιστική προσέγγιση που ακολουθήθηκε κατά την ανάπτυξη του παρόντος project. Για την υλοποίηση χρησιμοποιήθηκε μια μεθοδολογία κοντά στο *Test-Driven Development (TDD)*, με στόχο την προοδευτική ενσωμάτωση λειτουργικότητας με παράλληλη διατήρηση της σταθερότητας του κώδικα. Κάθε αρχείο Python που περιλαμβάνει την υλοποίηση ενός αλγορίθμου, συνοδεύεται από ενσωματωμένες συναρτήσεις ελέγχου (*test functions*), οι οποίες μπορούν να κληθούν μέσα από το μπλοκ `if __name__ == '__main__':`. Τα tests αυτά συμβάλλουν καθοριστικά στον άμεσο εντοπισμό λαθών, στην απομόνωση δυσλειτουργιών, και γενικότερα στην αξιοπιστία του συστήματος κατά την ανάπτυξη. Επιπλέον, η ύπαρξή τους προσφέρει ένα βασικό επίπεδο επιβεβαίωσης ότι η συμπεριφορά των αλγορίθμων είναι ορθή ως προς τις υποθέσεις της εκφώνησης, και διατηρείται σταθερή κατά την τροποποίηση ή προσθήκη νέου κώδικα.

Για την οπτική επαλήθευση των αποτελεσμάτων, τα tests παράγουν γραφήματα με χρήση του `matplotlib`, αξιοποιώντας ως backend το `TkAgg`. Αν το `TkAgg` είναι διαθέσιμο στο εκτελούμενο σύστημα, ο χρήστης μπορεί να επιλέξει να δει τα γραφήματα περνώντας ως ορίσμα `True` στη συνάρτηση ελέγχου. Ακολουθεί ενδεικτικό παράδειγμα χρήσης:

```
# If you have TkAgg you can pass plot=True, otherwise pass False
for k in [2, 3, 4]:
    _test_n_cuts(k, True)
```

Στην εικόνα 1 φαίνονται στιγμιότυπα από τέτοια plots.



Σχήμα 1: Ενδεικτικά plots από το αρχείο `image_to_graph` και το `spectral_clustering`.

3. ΕΙΚΟΝΕΣ ΩΣ ΓΡΑΦΟΙ

Το πρώτο βήμα της υλοποίησης αφορά τη μετατροπή μιας εικόνας σε γράφο, έτσι ώστε οι αλγόριθμοι φασματικής ανάλυσης που ακολουθούν να μπορούν να εφαρμοστούν. Η συνάρτηση `image_to_graph(img_array: np.ndarray) -> np.ndarray` λαμβάνει ως είσοδο μια εικόνα διαστάσεων $[M, N, C]$ με τιμές κανονικοποιημένες στο διάστημα $[0, 1]$, και επιστρέφει τον πλήρως συνδεδεμένο affinity πίνακα του γράφου. Η βασική υπόθεση είναι ότι κάθε εικονοστοιχείο (pixel) αντιστοιχεί σε έναν κόμβο του γράφου και ότι κάθε ζεύγος κόμβων συνδέεται με ακμή βάσει της χρωματικής τους ομοιότητας. Το βάρος της ακμής μεταξύ δύο κόμβων ορίζεται ως:

$$A(i, j) = \frac{1}{e^{d(i, j)}}$$

όπου $d(i, j)$ είναι η Ευκλείδεια απόσταση των διανυσμάτων RGB των δύο pixels. Ο μετασχηματισμός αυτός επιλέχθηκε γιατί προσφέρει εκθετική απόσβεση των αποστάσεων, περιορίζοντας την επίδραση απομακρυσμένων κόμβων και διατηρώντας ισχυρότερες τις συσχετίσεις μεταξύ παρόμοιων pixels. Η συνάρτηση περιλαμβάνει ρητό έλεγχο για τον τύπο των τιμών της εικόνας (`np.floating`), ενώ η διαγώνιος του πίνακα μηδενίζεται τεχνητά (`np.fill_diagonal(distances, 1e-10)`) ώστε να αποφευχθεί αριθμητική αστάθεια κατά την αντιστροφή των αποστάσεων.

Μέσα στο ίδιο αρχείο `image_to_graph.py`, περιλαμβάνονται δύο συναρτήσεις ελέγχου: `_test_1()` και `_test_2()`. Η `_test_1()` δημιουργεί μια τυχαία εικόνα 4x4 με RGB κανάλια, υπολογίζει τον πίνακα affinity και εμφανίζει βασικά χαρακτηριστικά του, όπως το σχήμα, την ελάχιστη/μέγιστη τιμή και τη συμμετρία του. Η `_test_2()` εφαρμόζει τη συνάρτηση στην πραγματική εικόνα `d2b` που παρέχεται στο `.mat` αρχείο και εκτελεί τους ίδιους ελέγχους. Ακολουθεί ενδεικτική έξοδος από την `_test_2()`:

```
=== Test 2 ===
Input image shape: (50, 50, 3)
dtype: float64
Affinity matrix shape: (2500, 2500)
Is symmetric: True
Max value: 1.0
Min value: 0.2129168031816433
```

Τα αποτελέσματα επιβεβαιώνουν ότι ο πίνακας είναι τετραγωνικός, συμμετρικός και έχει την αναμενόμενη πυκνή δομή (καθώς κάθε κόμβος συνδέεται με όλους τους υπόλοιπους).

4. Spectral Clustering

Ο αλγόριθμος spectral clustering βασίζεται στην ιδέα ότι η φασματική ανάλυση του Laplacian πίνακα ενός γράφου μπορεί να αποκαλύψει τη δομή των clusters που ενυπάρχουν στα δεδομένα. Δεδομένου ενός affinity matrix A , υπολογίζεται ο βαθμωτός πίνακας D , με $D(i, i) = \sum_j A(i, j)$. Σε αυτή την υλοποίηση μπήκαμε στην λογική να υλοποιήσουμε και μια έκδοση στην οποία ο Laplacian κανονικοποιείται. Στην unnormalized έκδοχή, ο Laplacian ορίζεται ως $L = D - A$, ενώ στη normalized μορφή ως:

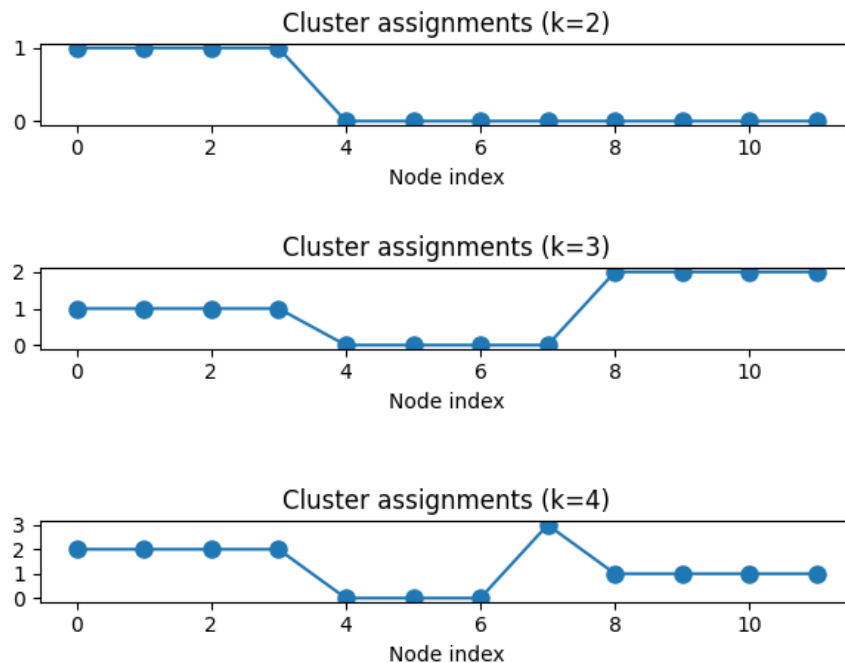
$$L_{\text{sym}} = I - D^{-1/2} A D^{-1/2}$$

Η ανάλυση βασίζεται στην επίλυση ιδιοτιμών του πίνακα L (ή του L_{sym}) και τη χρήση των k ιδιοδιανυσμάτων που αντιστοιχούν στις μικρότερες ιδιοτιμές, για την αναπαράσταση των κόμβων στον νέο embedding χώρο.

Στον embedding χώρο εφαρμόζεται ο αλγόριθμος k -means, ο οποίος ομαδοποιεί τα σημεία σε k clusters. Η υλοποίηση περιλαμβάνεται στη συνάρτηση `spectral_clustering(affinity_mat, k, normalized=False)` και υποστηρίζει τόσο normalized όσο και unnormalized έκδοχή, με χρήση της παραμέτρου `normalized`. Η επιλογή eigenvectors γίνεται με τη χρήση της `scipy.sparse.linalg.eigs`, ώστε να είναι αποδοτική ακόμη και για πίνακες μεγάλης διαστατικότητας.

Demo 1

Για την επίδειξη της λειτουργίας του αλγορίθμου χρησιμοποιήθηκε ο προκατασκευασμένος affinity πίνακας `d1a`, ο οποίος έχει διαστάσεις 12×12 και χρησιμοποιείται ως είσοδος για τρεις επαναλήψεις με $k = 2, 3$, και 4 . Τα αποτελέσματα παρουσιάζονται ως γραφήματα ετικετών κατά node index και παρουσιάζονται στην εικόνα 2 Σε

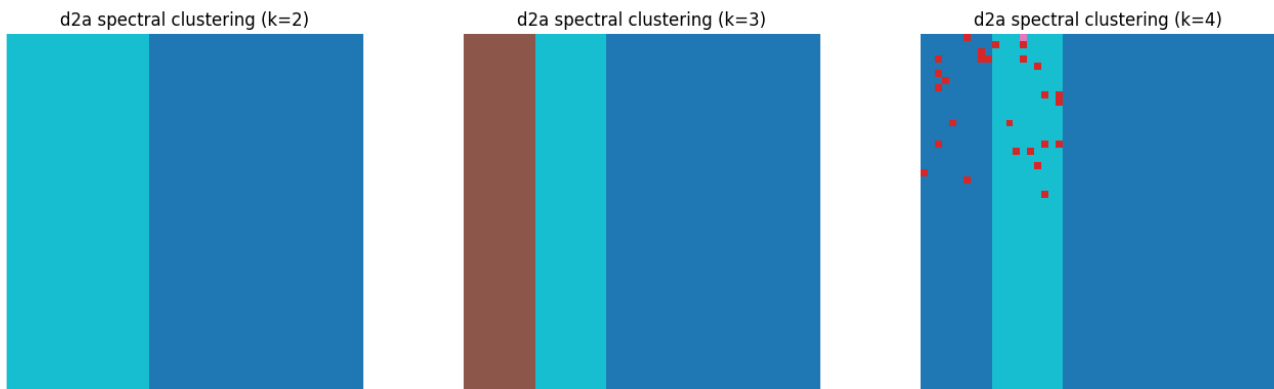


Σχήμα 2: Ενδεικτικά plots από το demo1.

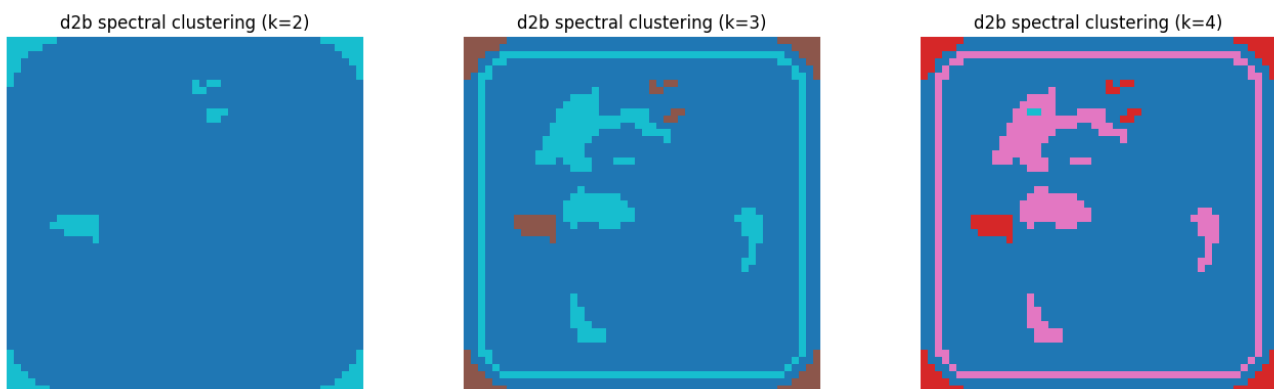
όλες τις περιπτώσεις, ο αλγόριθμος εντοπίζει ξεκάθαρες ομάδες, με σχεδόν πλήρη συμφωνία με τη δομή του πίνακα.

Demo 2

Η συνάρτηση εφαρμόστηκε στις εικόνες *d2a* και *d2b*, αφού πρώτα μετατράπηκαν σε γράφους με τη χρήση της *image_to_graph*. Για κάθε εικόνα εφαρμόστηκε τόσο η unnormalized όσο και η normalized εκδοχή του spectral clustering, και για $k = 2, 3, 4$. Τα αποτελέσματα της **μη κανονικοποιημένης** έκδοσης που ζητείται από την εκφώνηση παρουσιάζονται στις εικόνες **3, 4**



Σχήμα 3: Plots της *d2a* για unnormalized Laplacian.



Σχήμα 4: Plots της *d2b* για unnormalized Laplacian.

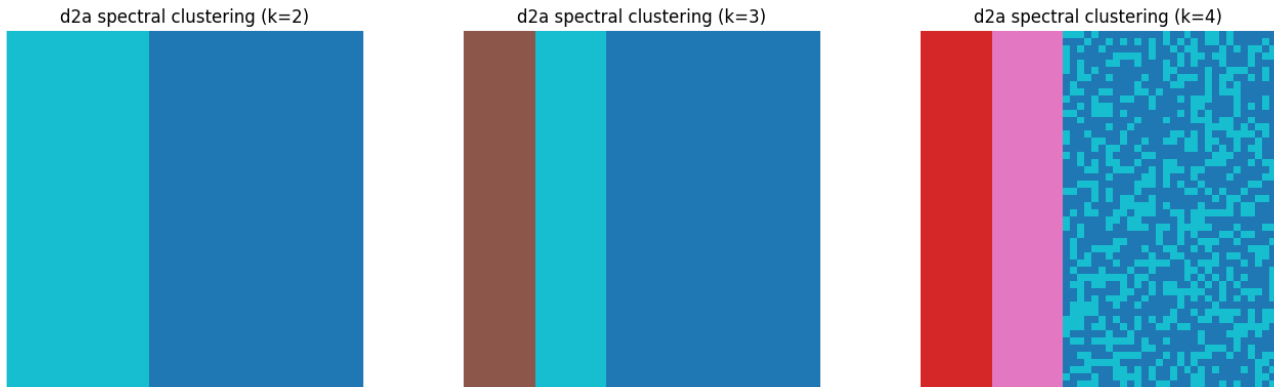
Αντίστοιχα τα αποτελέσματα της **κανονικοποιημένης** έκδοσης παρουσιάζονται στις εικόνες **5,6** και επιβεβαιώνουν την ευαισθησία της μεθόδου στην επιλογή του τύπου Laplacian.

Σύγκριση με scikit-learn

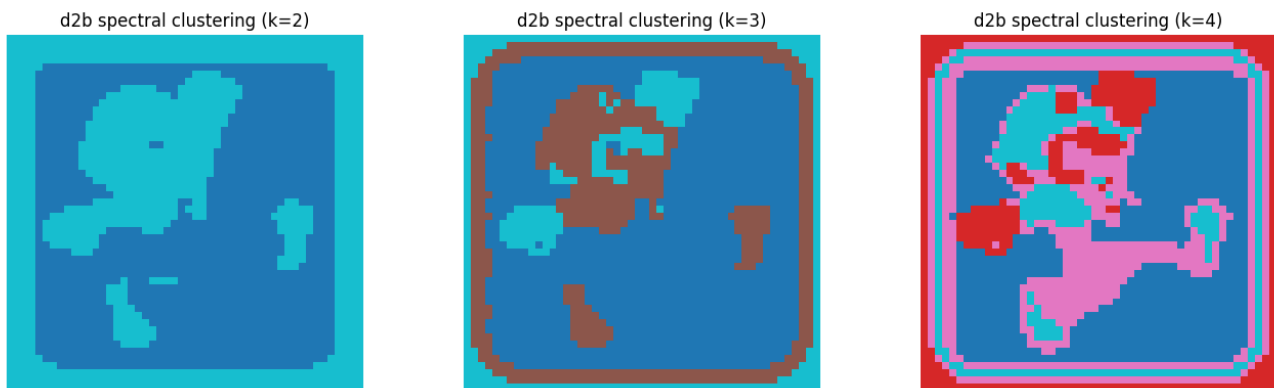
Για λόγους επαλήθευσης, συγκρίθηκε η υλοποίηση με την ενσωματωμένη *sklearn.cluster.spectral_clustering*, η οποία χρησιμοποιεί normalized Laplacian. Για $k = 3$, η συμφωνία μεταξύ των δύο μεθόδων ήταν απόλυτη (Adjusted Rand Index = 1.0000), ενώ για $k = 2$ και 4 υπήρξαν μικρές αποκλίσεις λόγω διαφορετικής ομαδοποίησης από τον *k-means*. Η συμφωνία υπολογίστηκε με τη μετρική *Adjusted Rand Index (ARI)* και τα αποτελέσματα καταγράφηκαν στην έξοδο της κονσόλας.

Γενικά παρατηρούμε ότι υπάρχει αρκετή σχέση μεταξύ τους της normalized εκδοσης και της reference υλοποίησης του *sklearn*. Θεωρούμε πως οι διαφορές μπορεί να οφείλονται στο:

- Αν περιλαμβάνεται το trivial eigenvector (π.χ. σταθερή τιμή),
- Αν *eigs()* δίνει ίδια σειρά ιδιοτιμών σε διαφορετικά runs,
- ή αν το floating point jitter που επηρεάζει το *k-means* όταν τα σημεία συμπίπτουν.



Σχήμα 5: Plots της d2a για normalized Laplacian.



Σχήμα 6: Plots της d2b για normalized Laplacian.

```
=== Comparing with sklearn spectral_clustering (k=2) ===
```

```
Labels (own):      [1 1 1 1 0 0 0 0 0 0 0 0]
```

```
Labels (sklearn): [0 0 0 0 1 1 1 1 0 0 0 0]
```

```
Adjusted Rand Index (ARI): 0.0294
```

```
=== Comparing with sklearn spectral_clustering (k=3) ===
```

```
Labels (own):      [2 2 2 2 0 0 0 0 1 1 1 1]
```

```
Labels (sklearn): [0 0 0 0 2 2 2 2 1 1 1 1]
```

```
Adjusted Rand Index (ARI): 1.0000
```

```
=== Comparing with sklearn spectral_clustering (k=4) ===
```

```
Labels (own):      [2 2 1 1 0 0 0 0 3 3 1 3]
```

```
Labels (sklearn): [3 3 0 3 2 2 2 2 1 1 1 1]
```

```
Adjusted Rand Index (ARI): 0.6379
```

5. Normalized Cuts

Η μέθοδος *normalized cuts* αποτελεί μια εναλλακτική προσέγγιση φασματικής ομαδοποίησης, η οποία διαφέρει από το κλασικό spectral clustering ως προς τη διατύπωση του φασματικού προβλήματος. Σε αυτή την περίπτωση επιλύεται το γενικευμένο ιδιοπρόβλημα $Lx = \lambda Dx$, όπου L είναι ο μη κανονικοποιημένος Laplacian $L = D - A$ και D ο βαθμωτός πίνακας. Η μέθοδος βασίζεται στην υπόθεση ότι ένα «καλό» διαμέρισμα του γράφου μεγιστοποιεί την εσωτερική συνοχή (association) και ελαχιστοποιεί τη διασύνδεση μεταξύ των ομάδων (cut).

Η βασική μετρική που υιοθετείται είναι η $Ncut$:

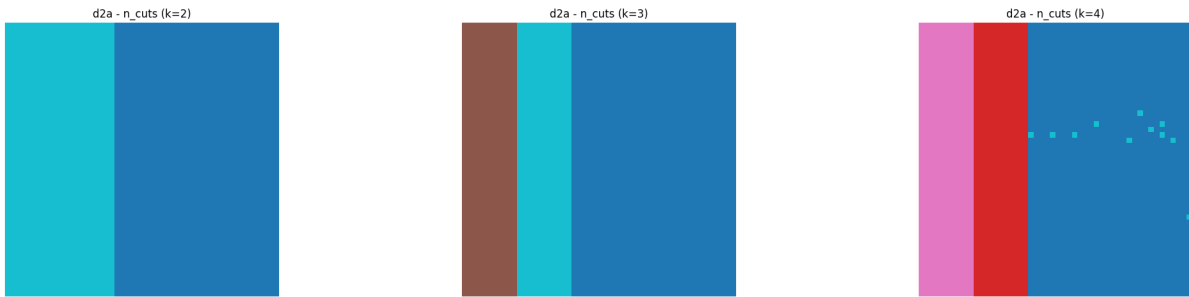
$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

όπου $cut(A, B)$ είναι το συνολικό βάρος των ακμών μεταξύ A και B , και $assoc(A, V)$ το άθροισμα των ακμών που ξεκινούν από το A προς όλους τους κόμβους. Μικρές τιμές της $Ncut$ υποδεικνύουν καλή διάσπαση.

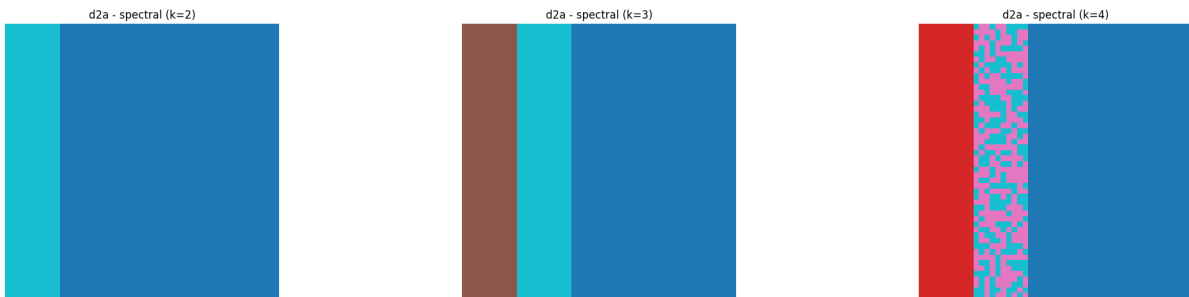
Η συνάρτηση n_cuts επιλύει το παραπάνω πρόβλημα για k ομάδες με φασματική ανάλυση και clustering στο φασματικό embedding με $k-means$. Η αναδρομική της μορφή, $n_cuts_recursive$, εφαρμόζει επαναληπτικά διχοτομήσεις, εφόσον πληρούνται κριτήρια μεγέθους ομάδας και κόστους διάσπασης.

Demo 3a

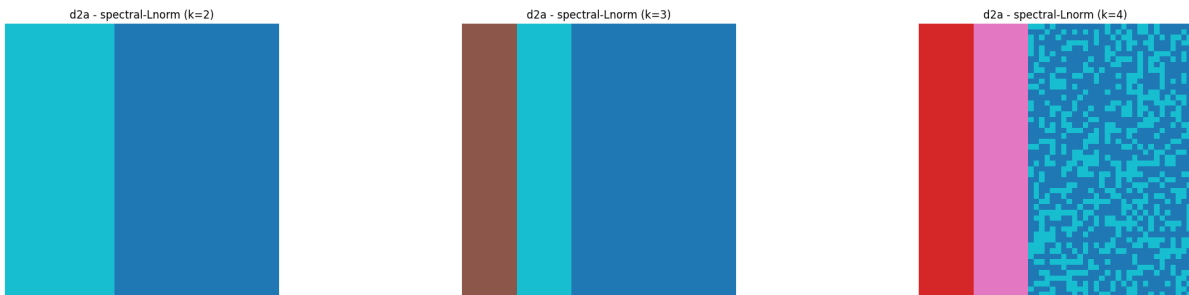
Στο πρώτο μέρος της επίδειξης εφαρμόστηκε η μη αναδρομική έκδοση της *normalized cuts* με $k = 2, 3, 4$ για τις εικόνες $d2a$ και $d2b$. Τα αποτελέσματα συγκρίθηκαν με εκείνα του spectral clustering, τόσο με την ζητηθείσα έκδοση όσο και με την *normalized* στον Laplacian. Στις εικόνες 7,8,9 παρουσιάζονται στιγμιότυπα των αποτελεσμάτων για την $d2a$, και από τις 3 μεθόδους. Αντίστοιχα στις εικόνες 10,11,12 παρουσιάζονται στιγμιότυπα των αποτελεσμάτων για την $d2b$, και από τις 3 μεθόδους.



Σχήμα 7: Αποτελέσματα μη αναδρομικού *normalized cuts* για $d2a$.

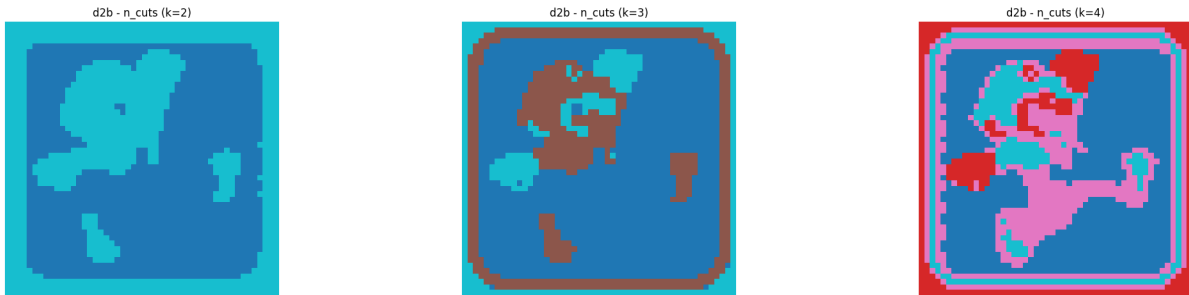


Σχήμα 8: Αποτελέσματα *spectral clustering* για $d2a$.

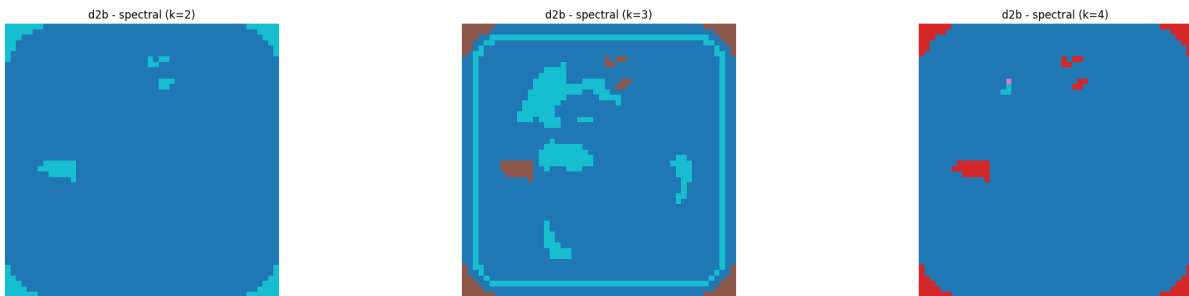


Σχήμα 9: Αποτελέσματα *spectral clustering* με *normalized Laplacian* για $d2a$.

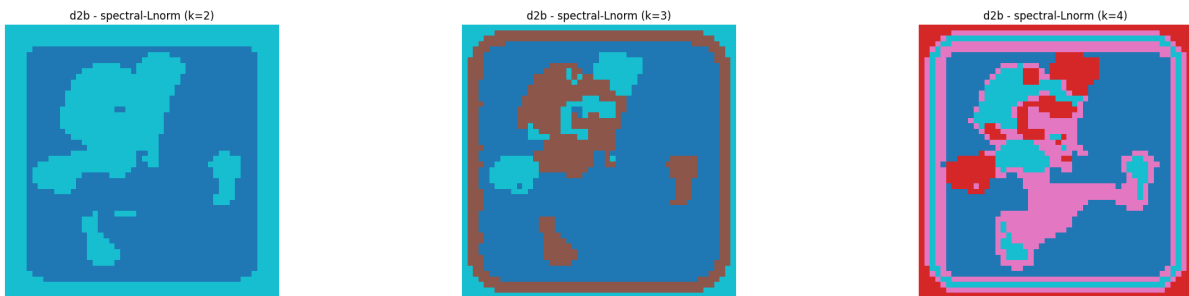
Από τα γραφήματα γίνεται κατανοητό ότι ενώ η *normalized_cuts* παρουσιάζει πολύ καλύτερη συμπεριφορά από την αρχική *spectral clustering*, η κανονικοποιημένη έκδοση της *spectral clustering*, έχει αντίστοιχες επιδόσεις.



Σχήμα 10: Αποτελέσματα μη αναδρομικού *normalized cuts* για *d2b*.



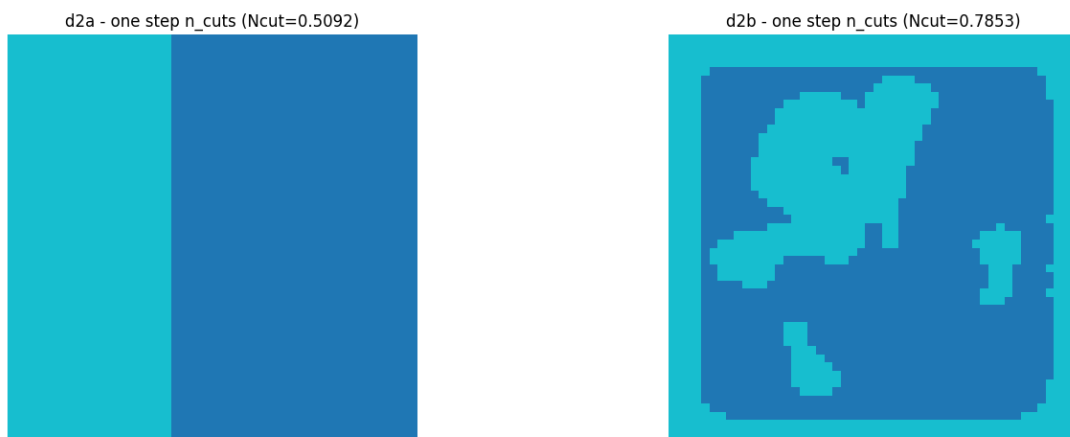
Σχήμα 11: Αποτελέσματα *spectral clustering* για *d2b*.



Σχήμα 12: Αποτελέσματα *spectral clustering* με *normalized Laplacian* για *d2b*.

Demo 3b

Για να αξιολογηθεί η ποιότητα ενός μόνο βήματος διάσπασης, εφαρμόστηκε η *n_cuts* με $k = 2$ και υπολογίστηκε η τιμή της *Ncut* με τη βοήθεια της *calculate_n_cut_value*. Στην εικόνα 13 φαίνεται η διχοτόμηση των *d2a* και *d2b*.



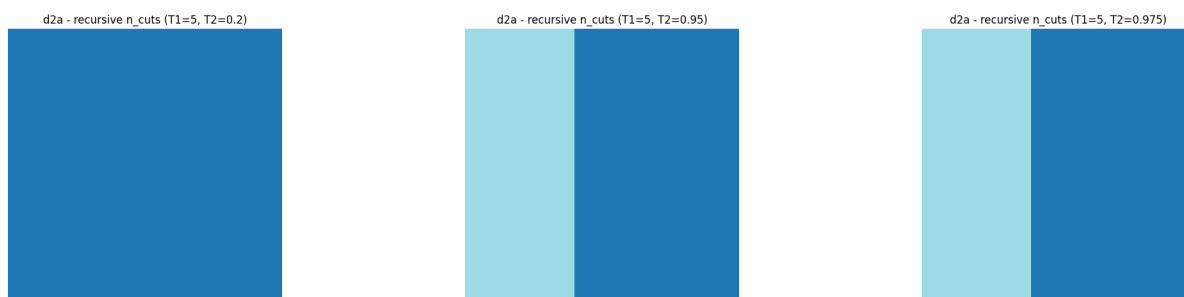
Σχήμα 13: Ένα βήμα *normalized cut* σε *d2a* και *d2b*.


```
=== Image d2a ===
Ncut value: 0.5092
```

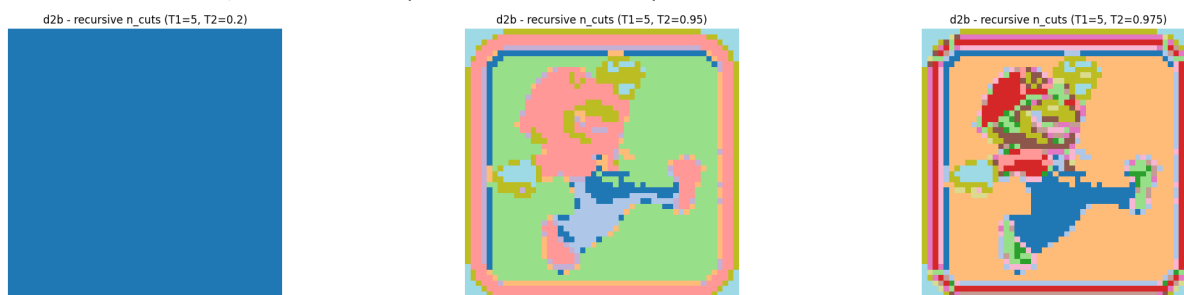
```
=== Image d2b ===
Ncut value: 0.7853
```

Demo 3c

Το τελικό σχέλος της επίδειξης αφορά την αναδρομική εκδοχή του *normalized cuts*. Η συνάρτηση *n_cuts_recursive* δέχεται κατώφλια $T1$ (ελάχιστο μέγεθος ομάδας) και $T2$ (μέγιστη αποδεκτή τιμή Ncut). Τα πειράματα εκτελέστηκαν με $T1 = 5$ και τρεις τιμές του $T2$: 0.20, 0.95 και 0.975, καθώς η δοθείσα τιμή 0.2 ήταν πολύ "αυστηρή" στον τερματισμό του αλγορίθμου. Η εικόνες 14,15 παρουσιάζουν τα αποτελέσματα για την εικόνες *d2b* και *d2b* αντίστοιχα.



Σχήμα 14: Αναδρομικό *normalized cuts* με διαφορετικά thresholds σε *d2b*.



Σχήμα 15: Αναδρομικό *normalized cuts* με διαφορετικά thresholds σε *d2b*.

Παρατηρήσεις για τα thresholds

Η τιμή του $T2$ επηρεάζει άμεσα το πόσο «εύκολα» συνεχίζεται η αναδρομή. Χαμηλές τιμές (π.χ. 0.2) καθιστούν τη μέθοδο αυστηρή, οδηγώντας σε ελάχιστες διασπάσεις. Μεγαλύτερες τιμές όπως 0.95 ή 0.975 παράγουν πιο λεπτομερή διαμερίσματα με περισσότερες ομάδες. Το κατώφλι $T1$ ελέγχει την ελάχιστη επιτρεπτή διάσταση ομάδας και αποτρέπει τον υπερκατακερματισμό. Στην πράξη, οι τιμές $T1 = 5$ και $T2 = 0.95$ έδωσαν αξιόπιστα αποτελέσματα και στις δύο εικόνες.

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο πλαίσιο της παρούσας εργασίας υλοποιήθηκαν εξ αρχής και αξιολογήθηκαν δύο τεχνικές φασματικής τμηματοποίησης εικόνας: το spectral clustering και η μέθοδος *normalized cuts*, τόσο στη βασική όσο και στην αναδρομική της μορφή. Για την εφαρμογή τους αναπτύχθηκε ενδιάμεση αναπαράσταση της εικόνας ως γράφου μέσω affinity matrix, ενώ παράλληλα υλοποιήθηκε διαδικασία φασματικής ανάλυσης με χρήση γενικευμένων ιδιοτιμών και ομαδοποίηση μέσω *k-means*. Η δομή του project υποστήριξε τη σταδιακή ανάπτυξη με έμφαση

στον έλεγχο ορθότητας μέσω ενσωματωμένων tests, τα οποία συνοδεύτηκαν από γραφική αποτύπωση των αποτελεσμάτων.

Από τα πειραματικά αποτελέσματα προκύπτει ότι οι δύο μέθοδοι παρουσιάζουν παρόμοια προσέγγιση, αλλά διαφέρουν στις λεπτομέρειες υλοποίησης και συμπεριφοράς. Το spectral clustering αποδείχθηκε σταθερό και αποδοτικό σε εικόνες με σαφώς διαχωρισμένα clusters, ενώ η μέθοδος normalized cuts προσφέρει επιπλέον έλεγχο μέσω των παραμέτρων $T1$ και $T2$, επιτρέποντας πιο προσαρμοστική τμηματοποίηση. Η συμφωνία με τη βιβλιοθήκη *scikit-learn* κρίθηκε ικανοποιητική, ειδικά όταν χρησιμοποιείται normalized Laplacian. Συνολικά, η εργασία ανέδειξε τη δύναμη των φασματικών μεθόδων στην τμηματοποίηση εικόνας, αλλά και τη σημασία του σωστού χειρισμού των παραμέτρων και της ανάλυσης του γράφου.