ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

Αναφορά στο μάθημα “Τεχνολογίες Γραφημάτων και Εφαρμογές”

Ιάσονας Αλέξανδρος Καραφωτιάς, ΑΜ: 218142

Διδάσκων: Μιχαήλ Δημήτρης

21/01/2024

Υλοποίηση αλγορίθμου:

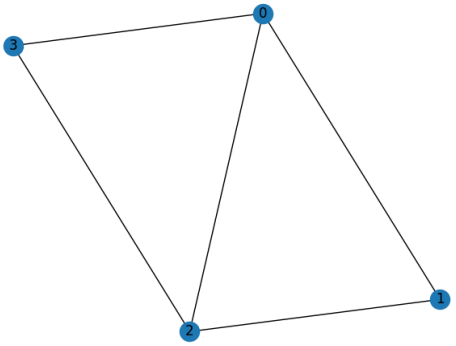
```python
def stochastic_gradient_descent(graph, rank, epsilon=1e-3, max_iter=1000, lambda_reg=0.1, clip_value=1e5):
    # Create a mapping from node labels to indices
    node_indices = {node: idx for idx, node in enumerate(graph.nodes())}
    # Initialize Z with small random values to prevent overflow
    Z = np.random.rand(graph.number_of_nodes(), rank) * 0.01
    # Repeat until convergence or maximum iterations
    for it in range(max_iter):
        Z_prev = Z.copy()
        # Update Zi for each node i
        for i in graph.nodes():
            i_idx = node_indices[i]  # Convert node label to index
            Zi = Z_prev[i_idx, :]
            grad_sum = np.zeros(rank)
            for j in graph.neighbors(i):
                j_idx = node_indices[j]  # Convert node label to index
                Zj = Z_prev[j_idx, :]
                Yij = graph[i][j]['weight'] if 'weight' in graph[i][j] else 1
                grad = (Yij - np.dot(Zi, Zj)) * Zj
                grad_sum += grad

            # Clip the gradient to prevent overflow (this is essential to prevent overflow)
            grad_sum = np.clip(grad_sum, -clip_value, clip_value)
            # Learning rate: decreases with iteration
            eta = 1.0 / np.sqrt(it + 1)
            # Update rule for Zi
            Z[i_idx, :] = Zi + eta * (grad_sum - lambda_reg * Zi)

        # Check for convergence
        norm_diff = np.linalg.norm(Z - Z_prev, 'fro')
        if norm_diff < epsilon:
            break
    return Z
```

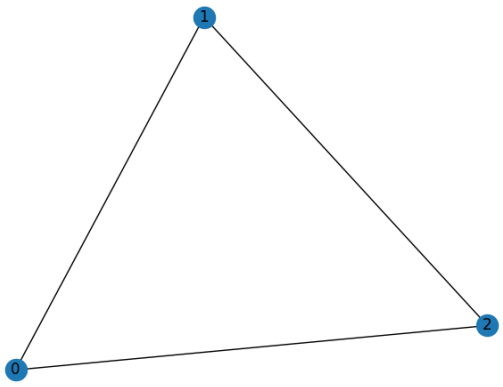
Έλεγχος σε μικρά γραφήματα (Testing.py):

Graph 1:



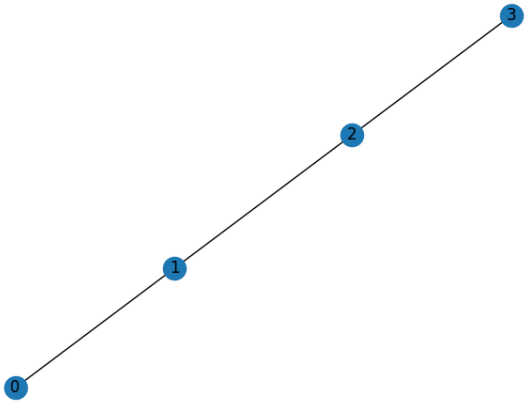

```
Run 1:
[[0.6822992  0.65942272]
 [0.68868883 0.65168736]
 [0.69428032 0.64725919]
 [0.67076771 0.67072481]]
Run 2:
[[0.71930064 0.61858362]
 [0.68918178 0.65369716]
 [0.72163047 0.61585208]
 [0.75087364 0.58166698]]
Run 3:
[[0.93824833 0.13600602]
 [0.93756777 0.15408654]
 [0.93890935 0.13012192]
 [0.94282988 0.11579171]]
Run 4:
[[0.45134401 0.83445794]
 [0.41910685 0.85186274]
 [0.45242691 0.83386097]
 [0.4909991  0.81296508]]
Run 5:
[[0.64322444 0.69729536]
 [0.61852944 0.71988693]
 [0.63886211 0.70139099]
 [0.67459842 0.66862591]]
Best Threshold: 0.8979591836734693
Best Similarity: 75.00%
```

Graph 2:




```
Epoch 31, Norm Difference: 0.001059274
073127
Epoch 32, Norm Difference: 0.000997279
071982
Convergence reached.
Run 1:
[[0.50613631 0.80288206]
 [0.52181678 0.7919877 ]
 [0.53586767 0.78328447]]
Run 2:
[[0.77663543 0.54498706]
 [0.77287749 0.54995847]
 [0.76076795 0.56737637]]
Run 3:
[[0.75594965 0.57387637]
 [0.74361037 0.58871604]
 [0.73403506 0.60154672]]
Run 4:
[[0.53623642 0.7823686 ]
 [0.5439088  0.7774769 ]
 [0.51640195 0.79642896]]
Run 5:
[[0.34382305 0.8842965 ]
 [0.37161783 0.87335697]
 [0.34959208 0.88178642]]
Best Threshold: 0.0
Best Similarity: 66.67%
```

Graph 3:




```
Run 1:
[[0.70462001 0.63722008]
 [0.68739774 0.65284634]
 [0.63931316 0.70461372]
 [0.57719686 0.75362379]]
Run 2:
[[0.87176284 0.39030621]
 [0.83315311 0.44674017]
 [0.80482336 0.51287117]
 [0.78375634 0.52397221]]
Run 3:
[[0.75704172 0.57344559]
 [0.70676238 0.63641597]
 [0.62931322 0.71406958]
 [0.56987976 0.75802448]]
Run 4:
[[0.91905959 0.23280434]
 [0.90457119 0.29418416]
 [0.87043294 0.38116534]
 [0.84203071 0.43830149]]
Run 5:
[[0.44649614 0.83832563]
 [0.5337252  0.7892208 ]
 [0.63656462 0.70843671]
 [0.69551546 0.64538054]]
Best Threshold: 0.8979591836734693
Best Similarity: 68.75%
```

Έλεγχος στο dataset Gene Fusion (Testing.py):

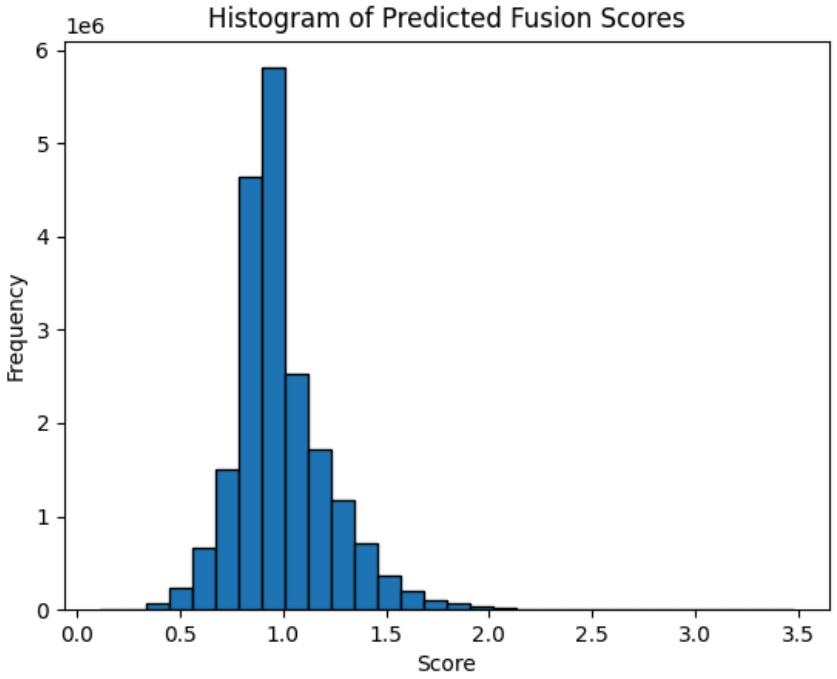
http://konect.cc/networks/gene_fusion/

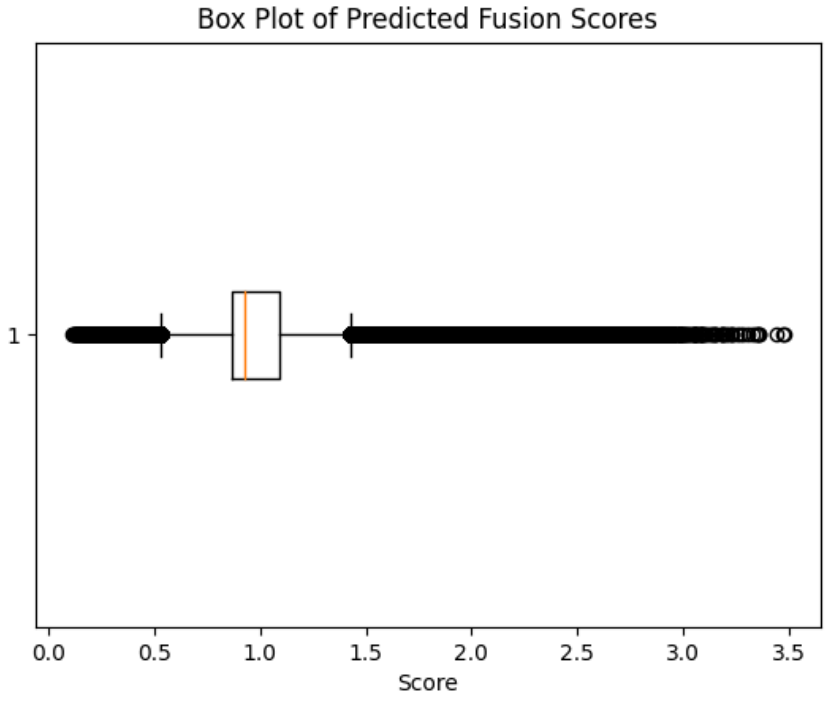
```
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [ 1.59334003e+03  1.59334003e+03  1.59334003e+03]
 [ 1.59334003e+03  1.59334003e+03  1.59334003e+03]
 [ 1.59334003e+03  1.59334003e+03  1.59334003e+03]
 [ 1.59334003e+03  1.59334003e+03  1.59334003e+03]
 [ 1.59334003e+03  1.59334003e+03  1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [-1.59334003e+03 -1.59334003e+03 -1.59334003e+03]
 [ 1.59334003e+03  1.59334003e+03  1.59334003e+03]]
Difference: 175.80102388780332, Similarity: 0.6399958065906417
Best Threshold: 1.0, Best Similarity: 0.6705145080315438
```

Ανάλυση αποτελεσμάτων (ScoreAnalysis.py):

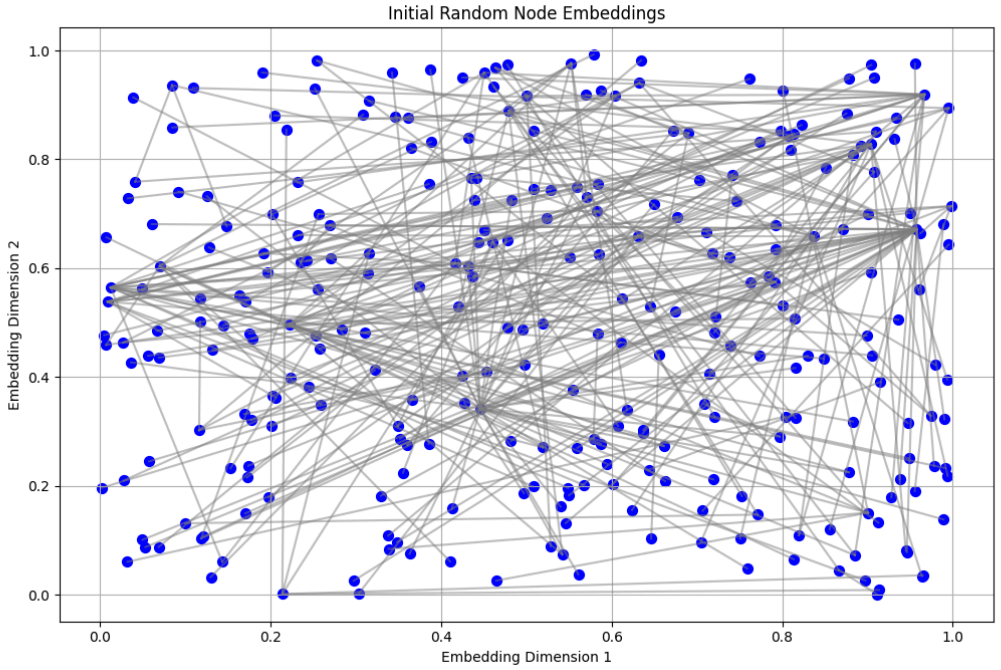
```
Statistical Summary:
Minimum Score: -0.053630973032419024
Maximum Score: 3.422011755636744
Mean Score: 0.9906931304621905
Median Score: 0.9281340494122686
Standard Deviation: 0.23792216963519922
```
]

Με βάση τα παραπάνω, φαίνεται ότι οι τιμές είναι μετρίως κατανεμημένες. Ο μέσος όρος και ο διάμεσος είναι κοντά ο ένας στον άλλο, κάτι που υποδηλώνει ότι υπάρχει μια συμμετρική κατανομή των βαθμολογιών γύρω από τη μέση τιμή.





Διαγραμματική αναπαράσταση των κόμβων και ακμών:




```python
G = nx.convert_node_labels_to_integers(G)

# Parameters
r = 2
epsilon = 0.001
lambda_reg = 0.01

Z = stochastic_gradient_descent(G, r, epsilon, lambda_reg)

# Visualization
if Z.shape[1] == 2:
    plt.figure(figsize=(12, 8))
    plt.scatter(Z[:, 0], Z[:, 1], s=50, c='blue')

    # Draw lines between connected nodes
    for i, j in G.edges():
        plt.plot([Z[i, 0], Z[j, 0]], [Z[i, 1], Z[j, 1]], 'grey', alpha=0.5)

    plt.xlabel('Embedding Dimension 1')
    plt.ylabel('Embedding Dimension 2')
    plt.title('Node Embeddings with Connections')
    plt.show()
```

Εφαρμογή: Πρόβλεψη σύντηξης γονιδίων κατά την ανάπτυξη καρκίνου

Dataset: http://konect.cc/networks/gene_fusion/

Το παραπάνω dataset περιλαμβάνει πληροφορίες σχετικά με ανθρώπινα γονίδια τα οποία έχει παρατηρηθεί ότι ενώνονται μεταξύ τους όταν παρουσιάζεται καρκίνος. Συγκεκριμένα, τα γονίδια αντιπροσωπεύονται ως κόμβοι και οι συνδέσεις μεταξύ τους υποδεικνύουν παρατηρούμενες συγχωνεύσεις γονιδίων στην εμφάνιση καρκίνου.

Ο συγκεκριμένος αλγόριθμος μπορεί ενδεχομένως να εφαρμοστεί στην εκπαίδευση ενός μοντέλου για την ανακάλυψη πιθανών συγχωνεύσεων γονιδίων που δεν έχουν παρατηρηθεί πριν ώστε να προβλέψουμε την πιθανότητα σύντηξης μεταξύ οποιωνδήποτε δύο γονιδίων, κάτι που μπορεί να είναι χρήσιμο για την έρευνα για τον καρκίνο, ειδικά για τον εντοπισμό νέων γονιδιακών αλληλεπιδράσεων που μπορεί να οδηγήσουν στην ανάπτυξη καρκίνου. Επομένως μπορεί να συμβάλλει στην έγκαιρη ανίχνευση του καρκίνου, την ανάπτυξη στοχευμένων γονιδιακών θεραπειών και την κατηγοριοποίηση γονιδίων βάσει της πιθανότητας τους να σχετίζονται με καρκίνο.

Ενδεικτικά αποτελέσματα (Training.py):

```
(VideoTraining) alexj@alexj-B560M-DS3H:~/PycharmProjects/Graphs$ python Training.py
Best parameters: {'rank': 2, 'lambda_reg': 0.01}
(VideoTraining) alexj@alexj-B560M-DS3H:~/PycharmProjects/Graphs$ python Training.py
Best parameters: {'rank': 10, 'lambda_reg': 0.01}
Final Model Evaluation:
Accuracy: 0.6904761904761905
Precision: 1.0
Recall: 0.6904761904761905
F1 Score: 0.8169014084507042
(VideoTraining) alexj@alexj-B560M-DS3H:~/PycharmProjects/Graphs$ python Training.py
Best parameters: {'rank': 2, 'lambda_reg': 0.01}
Final Model Evaluation:
Accuracy: 0.6785714285714286
Precision: 1.0
Recall: 0.6785714285714286
F1 Score: 0.8085106382978724
(VideoTraining) alexj@alexj-B560M-DS3H:~/PycharmProjects/Graphs$ python Training.py
Best parameters: {'rank': 2, 'lambda_reg': 0.01}
Final Model Evaluation:
Accuracy: 0.6428571428571429
Precision: 1.0
Recall: 0.6428571428571429
F1 Score: 0.782608695652174
(VideoTraining) alexj@alexj-B560M-DS3H:~/PycharmProjects/Graphs$
```

Συμπέρασμα:

Το μοντέλο μας φαίνεται να έχει σταθερά σχετικά καλή απόδοση. Η απόδοση εν είναι εξαιρετικά υψηλή, κάτι το οποίο θεωρούμε αναμενόμενο δεδομένου του μικρού μεγέθους δεδομένων στα οποία εκπαιδεύτηκε το μοντέλο και του σύντομου χρόνου εκπαίδευσης. Ωστόσο το μοντέλο παρουσιάζει σταθερά την ίδια απόδοση, κάτι το οποίο υποδηλώνει ότι η μεθοδολογία βρίσκεται στη σωστή κατεύθυνση και ενδεχομένως έχει περιθώρια βελτίωσης.

Θεωρούμε ότι αν η εκπαίδευση επεκταθεί σε μεγαλύτερα dataset και ενδεχομένως με επιπλέον βελτιστοποιήσεις η μεθοδολογία του μοντέλου μας μπορεί βελτιωθεί και να έχει πράγματι πρακτική εφαρμογή στην μελέτη του καρκίνου και των γονιδιακών μεταλλάξεων.