CS590-MLG

Soham Mukherjee

Problem-1

1.1. ① If we are agnostic to the number of clusters define Eigengap heuristic as,

$|\lambda_k - \lambda_{k+1}|$ where $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{|V|}$ are eigen values of L_{sym}.

Choose a k that has large $|\lambda_k - \lambda_{k+1}|$.
This would imply there are k-many communities.

② If exactly k is known then we can perform hierarchical clustering.

② If we know exactly there are k-clusters we would like to take first k-eigenvectors.

But consider the case where they are k-perfectly disconnected components. As a result 0 will have a multiplicity of k. In this case the eigenspace is spanned by k-cluster indicator vectors.

1.2. Let us choose a partition of the vertices $V \in G$.

WLOG,

$$V_1 = \{1, 2, \ldots, n\}$$
$$V_2 = \{n+1, \ldots, 2n\}$$
$$\vdots$$
$$V_r = \{(r-1)n+1, \ldots, rn\}$$
$$\vdots$$
$$V_k = \{(k-1)n+1, \ldots, kn\}.$$

Let us define,

$$M = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ n \\ n+1 \\ \vdots \\ \\ \\ (k-1)+1 \\ \\ nk \end{array} \begin{array}{c} \overset{1\,2\,\ldots\,n}{} \\ \left[\begin{array}{cccc} 0\,p\ldots p & a\,a\ldots a & a\ldots a & \ldots\ldots a\ldots a \\ p\,0\ldots p & a\,a\ldots a & a\ldots a & \ldots\ldots a\ldots a \\ p\ldots 0 & a\,a\ldots a & a\ldots a & \ldots\ldots a\ldots a \\ a\ldots a & 0\,p\ldots p & a\ldots a & \ldots\ldots a\ldots a \\ a\ldots a & p\,0\ldots p & a\ldots a & a\ldots a \\ \vdots & & & \\ & & & \\ \vdots & & & \\ a\ldots a & a\ldots a & a\ldots a & \ldots \\ a\ldots a & a\ldots a & a\ldots a & \ldots \end{array}\right. \\ \overset{(k-n)+1\ldots nk}{} \\ \left.\begin{array}{c} a\ldots a \\ a\ldots a \\ a\ldots a \\ a\ldots a \\ a\ldots a \\ \\ \\ \\ 0\,p\ldots p \\ p\,0\ldots p \\ p\,p\,0\ldots 0 \\ p\,p\ldots 0 \end{array}\right] \end{array}$$

Then, $A_{ij} = 1$ with probability M_{ij} and $A_{ij} = A_{ji}$.

$$E(A) = \left[\begin{array}{cccc} 0\,p\ldots p & a\ldots a & a\ldots a & \ldots\ldots a\ldots a \\ p & & & \\ p\ldots 0 & a\ldots a & a\ldots a & \ldots\ldots a\ldots a \\ a\ldots a & 0\,p\ldots p & a\ldots a & \ldots\ldots a\ldots a \\ a\ldots a & p\,p\ldots 0 & a\ldots a & a\ldots a \\ a\ldots a & & & \\ a\ldots a & a\ldots a & a\ldots a & \ldots \\ a\ldots a & a\ldots a & a\ldots a & 0\,p\ldots p \\ & & & \\ a\ldots a & a\ldots a & a\ldots a & p\,p\ldots 0 \end{array}\right]$$

For each

WLOG consider the vertex v_1.

For v_1, there are $(n-1)$ edges with prob. p.

" " $(kn-n)$ " " " α.

$$\therefore E(d_{v_1}) = p(n-1) + n(k-1)\alpha$$

$$= np - \cancel{\alpha}p + kn\alpha - n\alpha$$

$$\approx n(p + k\alpha - \alpha)$$

$$= n\left[p + (k-1)\alpha\right]$$

$$\therefore E(D) = \text{diag}\left(n\left[p+(k-1)\alpha\right].\right)_{kn \times kn}.$$

1.3.

Suppose $k=2$, then,

$$\bar{L}_{sym} = \bar{I} - \bar{D}^{-1/2} \bar{A} \bar{D}^{-1/2}$$

$$\bar{D} = \text{diag}\left(n(p+q) \right)_{\frac{1}{2}n \times \frac{1}{2}n}$$

Notice that,

$$\bar{L}_{sym} \cdot \vec{1} = \left(1 - (n-1)p + nq\right) \vec{1}.$$

~~$\therefore \lambda_1 \approx 1 + (n+1)p$~~

$$\Rightarrow \quad \lambda_1 \approx 1 - n(p+q)$$

Note in the given question, if $\bar{L} = D^{-1/2} A D^{-1/2}$ then, $\lambda_1 \approx n(p+q)$. and $\vec{1}$ is an eigenvector.

Since the second eigenvector is constant on each cluster ~~fix~~, and orthonormal to $\bar{1}$, fix.

$$e_2 = \begin{cases} \frac{1}{\sqrt{n}} & \text{if } i \in V^1 \\ -\frac{1}{\sqrt{n}} & \text{if } i \in V^2 \end{cases}$$

$\therefore$ ~~$Me_2 = \lambda_2 e_2$~~ yields,

~~$\lambda_2 = n(p-q)$.~~

$$\lambda_2 = 1 - n(p-q).$$

Thus, $\lambda_2 = 1 - n(p-q)$ and $e_2 = \begin{cases} \frac{1}{\sqrt{n}} & \text{if } i \in V^1 \\ -\frac{1}{\sqrt{n}} & \text{if } i \in V^2 \end{cases}$.

1.4. ~~Generalize~~

Generalizing results of 2. we get, (in Question). and $k \geq 2$,

for $\bar{L} = \bar{D}^{-1/2} \bar{A} \bar{D}^{-1/2}$

$\lambda_1 = n\left[p + (k-1)\alpha v\right]$

$\lambda_2 = n\left[(p-\alpha v) + (k-2)\alpha v\right]$

$\lambda_3 = n\left[(p-2\alpha v) + (k-3)\alpha v\right]$

$\vdots$

$$\lambda_1 = n\left[p + (k-1)\alpha v\right] \rightarrow \text{Largest.}$$

Smallest will be

$$\lambda_k = n\left[p - (k-1)\alpha v\right]$$

The eigenvalues will be

$$n\left[p+(k-1)\alpha v\right] \leq n\left[p + (k-3)\alpha v\right] \leq n\left[p + (k-5)\alpha v\right] \leq$$

$$\cdots \leq n\left[p-(k+1)\alpha v\right] \leq n\left[p-(k-1)\right]$$

This is from the fact that there are $2(k-1)$ eigenvalues and difference between smallest and the largest is $2(k-1)\alpha v$.

~~No~~

1.5. Note that there is a distinction when the sign of (k - var) flips from +ve to -ve.

That is consistent with our eigengap heuristics $|\lambda_k - \lambda_{k+1}|$.

Problem-2

2.1. Random walk matrix is defined as,

$$W = D^{-1} A.$$

where, $D^{-1}_{ij} = 0$ if $D_{ij} = 0$

$$= \frac{1}{d_{ij}} \quad \text{if } D_{ij} = d_{ij} \neq 0. \quad = \frac{1}{d_i}$$

Let $\vec{e}$ be an eigen vector of Random walk matrix and λ is the corresponding eigenvalue.

$$\therefore \quad W\vec{e} = \lambda\vec{e}$$

$$\Rightarrow \quad D^{-1} A \vec{e} = \lambda \vec{e}$$

$$\Rightarrow \quad D^{-1} A -$$

Define Normalized Random Walk matrix as,

$$N = D^{-1/2} W D^{1/2}$$

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

Given, $W = D^{-1} A.$

$$\Rightarrow D^{1/2} W = D^{-1/2} A.$$

$$\Rightarrow D^{1/2} W D^{-1/2} = D^{-1/2} A D^{-1/2}$$

Let λ be an eigen vector of the matrix $N = D^{1/2} W D^{-1/2}$ with $\vec{u}$ being the eigenvector.

$$\therefore \quad N\vec{u} = \lambda\vec{u}$$

$$\Rightarrow \left(D^{1/2} W D^{-1/2}\right) u = \lambda \vec{u}$$

$$\Rightarrow \quad W D^{-1/2} u = \lambda D^{-1/2} u$$

** This gives $D^{-1/2}u$ is an eigen vector of W if $\vec{u}$ is an eigenvector of $N = D^{1/2} W D^{-1/2} = D^{-1/2} A D^{-}$

Let $\vec{v}$ be an eigenvector of Lsym with eigenvalue μ.

$$\therefore \quad L_{sym}\vec{v} = \mu\vec{v}$$

$$\Rightarrow \left(I - D^{-1/2} A D^{-1/2}\right)\vec{v} = \mu\vec{v}$$

$$\Rightarrow \left(I - D^{1/2} W D^{-1/2}\right)\vec{v} = \mu\vec{v}$$

$$\Rightarrow \left(D^{1/2} W D^{-1/2}\right)\vec{v} = (1-\mu)\vec{v}$$

** Clearly if μ is an eigenvalue of Lsym with eigenvector $\vec{v}$, we have, $(1-\mu)$ as eigenvalues of W with eigenvector $D^{-1/2}\vec{v}$.

2.2 Assuming π the ~~prob~~ stationary probability distribution π and W, the random walk matrix we get,

~~$P\pi = \pi$~~ $W\pi = \pi$.

with $\pi(i) > 0 \quad \forall\, i \in V$ and $\sum_i \pi(i) = 1$.

As W has a stationary distribution π then it satisfies detailed balance condition i.e.

$\pi(i) P_{ij} = \pi(j) P_{ji}$

It is easy to see, $\pi(i) = \pi(i) \sum_{j \in V} P_{ij} = \sum_{j \in V} \pi(i) P_{ij}$

$= \sum_j \pi(j) P_{ji}$

Let, $\frac{\pi(i)}{d(i)} = \frac{\pi(j)}{d(j)} = k$

$\therefore \quad 1 = \sum_{i \in V} \pi(i) = k \sum_{i \in V} d(i) = k\, 2m.$

$\therefore k = \frac{1}{2m}$

$\pi(i) = \frac{d(i)}{2m} \quad \forall\, i \in V.$

Recall from problem 2.1 that if v_i is an eigenvector of $N = D^{1/2} W D^{-1/2}$ then, $D^{-1/2} v_i$ is an eigenvector of W.

N being a symmetric matrix we have n eigenvalues and corresponding set of orthogonal vectors $v_1, v_2, \ldots, v_n$.

$\Rightarrow$ W has also n eigenvalues and eigenvectors corresponding to the eigenvalues. The vectors are $D^{-1/2} v_1, D^{-1/2} v_2, \ldots, D^{-1/2} v_n$.

$p^t = W^t$.

$$\begin{aligned} x^t &= W^t x^0 \\ &= D^{-1/2} N^t D^{1/2} x^0 \\ &= D^{-1/2} \left(\sum_{i=1}^{n} \lambda_i^t v^i (v^i)^T \right) D^{1/2} p_0 \\ &= D^{-1/2} \sum_{i=1}^{n} \lambda_i^t \left(D^{-1/2} v_i \right) \left(v_i^T D^{\frac{1}{2}} \right) p_0 \\ &= \pi + \sum_{i=2}^{n} \lambda_i^t \left(D^{-1/2} v_i \right) \left(v_i^T D^{\frac{1}{2}} \right) p_0 \end{aligned}$$

This will only converge if $|\lambda_i|^t < 1$. as $t \to \infty$.

And will not converge if $|\lambda_i| = 1$.

Thus there is no stationary distribution if graph G is bipartite. as for bipartite graphs $\lambda_2 = -1$.

2.3

Proximity similarity can be quantified by —

(i) Shortest path
(ii) Common neighbors [Not sure]
(iii) PageRank
(iv) Commute-time

Random-walk distances take account of
(i) All nodes, All paths, i.e. multiple paths.
(ii) Degree of the node
(iii) Indirect connections.

2.4

Another embedding approach will be as that of the DeepWalk, i.e. truncated Random Walk. As we know Random Walk respects proximity similarity it is safe to assume DeepWalk also respects that.

Algorithm: In short the algorithm is

- for $i=0$ to γ do
  - Generate permutation of V
  - for each vertex
    - W_{v_i} = RandomWalk
    - Update representations (embeddings) i.e: if v appears in W_u then v is more similar to u.

Objective: Estimate the likelihood

$$Pr\left(v_i \mid (\phi(v_1), \phi(v_2), \ldots, \phi(v_{i-1}))\right)$$

which relaxes to

$$\underset{\phi}{\text{minimize}} \; -\log Pr\left(\{v_{i-w}, \ldots, v_{i+w}\} \setminus v_i \mid \phi(v_i)\right) \quad \text{---(1)}$$

ϕ is the mapping function $\phi: v \in V \rightarrow \mathbb{R}^{|V| \times d}$.

$$\min_{h_v, v \in V} \sum_{u \in V} \sum_{v \text{ in walks of } u} \left(-\log \frac{\exp\langle h_v, h_u \rangle}{\sum_{v' \in V} \exp\langle h_{v'}, h_u \rangle}\right) \quad \text{---(2)}$$

(1) can be rewritten as output of hierarchical softmax output given in (2).

~~2.2~~ ~~2.5~~

~~We may~~

2.5

We can use local computation.

Suppose for a node $v_i \in V$ we take its k-nearest neighbors.

Node2vec uses BFS + DFS to compute node representations locally that corresponds to $O(|V| + |E|)$ and then uses SGD to ~~optimize the~~ maximize the likelihood of preserving network neighborhoods.

.. Both Deepwalk and Node2vec draw inspiration from NLP where local structures are important and aims to maximize the likelihood of neighborhood structure information by learning an embedding.

2.6

The similarity between nodes in social ~~works~~ networks ~~has~~ * can be characterized by:

(i) Triadic closure (User-user similarity)

(ii) Status, - Similarity: For example level of recognition, merit, reputation affects review.

Problem 3

3.1

Filtering: Graph filtering is the process of smoothing, band-limiting graph signals, i.e. multiplying spectra of a graph with filter function mimicking the classical signal processing.

The three steps are,

- Transform x to freaquency domain:

$$\hat{x} = F^T x.$$

- Filter $\hat{x}$ with transfer function:

$$\hat{y} = C\hat{x}$$

- Transform $\hat{y}$ to time domain

$$y = F\hat{y}.$$

i.e. $$y = FCF^T x.$$

F is generally obtained through eigen-decomposition.

i.e. $F = (f_1, f_2, \ldots, f_n)$

f_i is the eigenvector of Normalised Laplacian.

we may use,

$$y = \sum_{i=0}^{k} \theta_i \hat{A}^i x.$$

$$= F \left(\sum_{i=0}^{k} \theta_i \Psi^i \right) F^T x.$$

$\hat{A}$ = $I - L_{sym}$ and therefore Ψ is its eigenvalue matrix. with $\phi_i = 1 - \lambda_{n+1-i}$ as eigenvalues.

$$\therefore C_{uu} = \sum_{i=0}^{k} \theta_i \phi_u^k$$

3.3

Define $p^0 = (1-\alpha) 1_v$

$$\Rightarrow p^1 = (1-\alpha) 1_v + \alpha W (1-\alpha) 1_v$$

$$= (I + \alpha W)(1-\alpha) 1_v.$$

$$p^2 = (1-\alpha) 1_v + \alpha W p^1$$

$$= (1-\alpha) 1_v + \alpha W (I + \alpha W)(1-\alpha) 1_v.$$

$$= (I + \alpha W + \alpha^2 W^2)(1-\alpha) 1_v.$$

If we write it in the form of Graph convolution it turns out,

$$p_{pr} = \sum_{i=0}^{k} (\alpha W)^i (1-\alpha) 1_v.$$

$$= (1-\alpha) \sum_{i=0}^{k} (\alpha W)^i 1_v.$$

3.4

The filtering emphasizes low-freaument part as high-frea generally corresponds to noise.

For the heat-kernel Pagerank

$$\mathcal{S}_{t,f} = e^{-t} \sum_{k=0}^{\infty} \frac{t^k}{k!} f W^k$$

~~Says~~

For a graph G,

- G has a stationary distribution if random walk matrix is irreducible and aperiodic.

Notice that while deriving convergence of $W^k \pi \to \pi$ we argued that $|\omega_i| < 1$ reduces to 0 as $k \to \infty$.

we can think of $|\omega_i|$ as the freaurency of the spectra. [ω_i being i^{th} eigenvalue of W]

As $k \to \infty$, only $\omega_1 = 1$ component remains. From classical fourier-series expansion it is known that ω is the dc-component, i.e. low freaurency component.

Therefore, Pagerank preserves low-freaurency components.

From problem 2.2,

if $N = D^{-1/2} W D^{1/2}$, the normalized Random walk matrix

we know that,

$$W^t = D^{-1/2} N^t D^{1/2}$$

$$= \pi + \sum_{i=2}^{n} \lambda_i^t \left(D^{-1/2} v_i\right)\left(v_i D^{1/2}\right)^T$$

For personalized Page rank,

$$y(\lambda) = (1-\alpha) \sum_{t=0}^{k} (\alpha W)^t 1_v.$$

$$\sum_{t=0}^{k} (\alpha W)^t = \sum_{t=0}^{k} \alpha^t \left[\pi + \sum_{i=2}^{n} (\lambda_i)^t \left(D^{-1/2} v_i\right)\left(v_i D^{1/2}\right)^T\right]$$

$$= \sum_{t=0}^{k} \pi \alpha^t + \sum_{i=2}^{n} \left(D^{-1/2} v_i\right)\left(v_i D^{1/2}\right)^T \left(\sum_{t=0}^{k} (\lambda_i)^t\right)$$

Remember $|\alpha| < 1$, and let's say $t \to \infty$.

$$= \underbrace{\frac{\pi}{1-\alpha}}_{\text{DC-component}} + \underbrace{\sum_{i=2}^{n} \left(D^{-1/2} v_i\right)\left(v_i D^{1/2}\right)^T \frac{1}{1-\lambda_i}}_{\text{Low-Pass filter.}}$$

3.5

Two layers that use Graph convolution layer is

i) The classical Graph convolution layer.

ii) Sage Conv layer. → Sample and Aggregate conv layer.

Equation Given $X \in \mathbb{R}^{|V| \times d}$ and $Y \in \mathbb{R}^{|V| \times d'}$

(i) GCN layer:

$$y_i = \sum_{j \in N(i) \cup \{i\}} \frac{1}{\sqrt{\deg(i)}\sqrt{\deg(j)}} \cdot (\theta \cdot x_j)$$

where y_i is the i^{th} component of Y and θ is the weight-matrix (linear transformation) $\theta : d \to d'$.

(ii) SageConV layer:

$$y_i = W_1 x_i + W_2 \cdot \text{Aggr}_{j \in N(i)} x_j$$

Again, $W_1 : d \to d'$, $W_2 : N(i) \times d \to d'$

y_i denotes i^{th} component of $Y \in \mathbb{R}^{|V| \times d'}$.

Aggr is an aggregator function like mean, sum, max-pool.

3.6

As mentioned by Zhang et. al. deeper convolution layer results into Laplacian smoothing.

To overcome this issue, following remedies can be taken.

(i) Introduction of skip-connection like Jumping Knowledge Network.

(ii) Layerwise attention. (I don't know whether it is implemented).

(iii) Soft-clustering of similar nodes, adding the clustering information as features, forcing the network to distinguish between different clusters. [Like diff-pool].

(iv) Residual Network $h^{(t)} \leftarrow \sigma\left(\hat{A} h^{t-1} \theta^{t-1}\right) + \underbrace{h^{t-1}}_{\text{residual term.}}$

3.7 Spectral embedding computes the node embedding based on the whole graph, i.e. solving the entire Graph Laplacian.

GNN embeddings is generally learned through information sources from neighbors (by aggregating and pooling).

.. Spectral embedding is not scalable. For large social network graphs or sparse graphs solving the entire Laplacian is redundant. Whereas GNN has the capability to handle such large or sparse graphs. Also spectral embedding lacks inductive capability; GNN does not.

As of now GNN has problem when the layers are too deep. GCN does not perform well for semisupervised learning.

Problem-4 [In question paper it is Problem 5]

(5.1) 4.1 Minibatch training as opposed to full batch training provides two key benefits:

(1) The model may get stuck at local minima for full batch training. For mini-batch training gradients are calculated based on single point and therefore noisy. The ~~noisy~~ injection of noise helps the model to get out of local minima.

(2) Minibatch training is faster, parallelizable.

4.2 (5.2)

In GNN minibatch training is not trivial because $\{(\vec{x}_i, y_i)\}$ are correlated.

For example in node classification task, $\forall i \in V$, x_i feature of node i, $\hat{y}_i$ depends on not only x_i but also other nodes (probably neighbors of node i).

.. Problem with GraphSAGE minibatch:

For each node v, we need k-hop trees, even we subsample d neighbors each layer will have complexity of $O(d^k d_v)$. Total complexity $O(|E| d^k)$.

They cannot be deep.

5·3 (4·3)

Idea to avoid the problem in minibatch training of GraphSAGE:

① Tradeoff of space with time. Avoid training of redundant, repetitive computation of k-hop trees and store them somehow, i.e. perform convolution on the fly.

② Cluster the large graph into several block. For each minibatch take blocks instead of nodes. [Cluster-GCN]

③ Sample a subgraph. Do minibatch training. Then do full batch on whole graph. [Graph-SAINT].

④ Define importance based K-hop graph.

Problem-6

6.1

The two conditions for entire graph representations are the following:

Expressiveness: If for any permutations $\pi: V \to V$,

$$(\pi(A_1), \pi(X_1)) \neq (A_2, X_2)$$

then the most expressive graph representation should satisfy $f(A_1, X_1) \neq f(A_2, X_2)$.

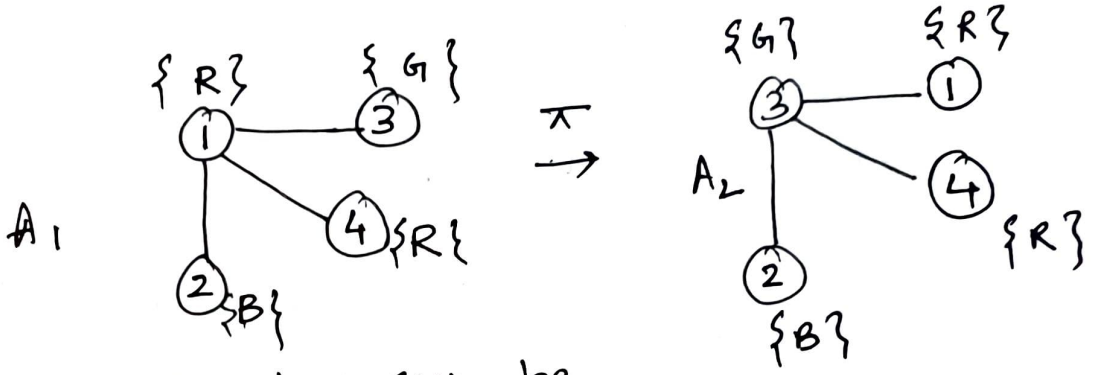
Permutation Invariance: If there is a π making

$$(\pi(A_1), \pi(X_1)) = (A_2, X_2) \text{ then } f(A_1, X_1) = f(A_2, X_2).$$

6.2:

Intuitively the first condition tells that if a graph f is a graph representation then, it should not represent different graphs (i.e. permutation of a graph that is different) in a similar way.

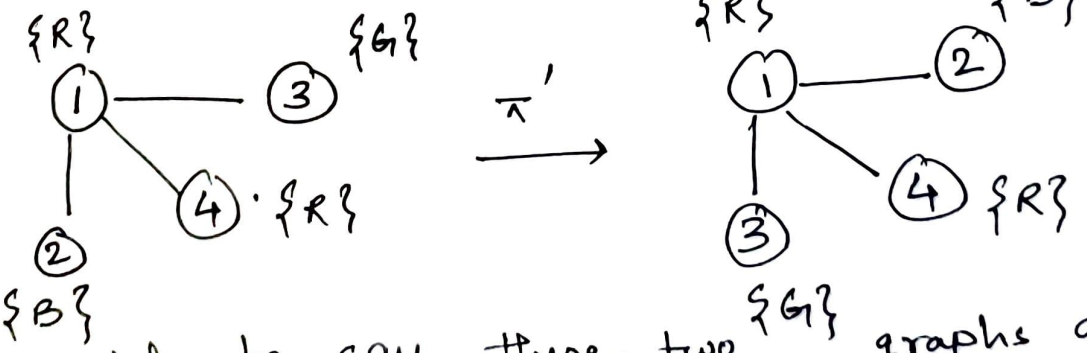
Example:



Then f should be able to distinguish between the representations.

The next condition implies that if they are equal then f should be able to tell that they are equal.

Example:



f should be able to say these two graphs are equal.

** In the above graphs {R} → denotes node attribute color red, {G} → green and {B} → blue.

6.3

WL-Test: WL-tests are family of algorithms that checks graph isomorphism.

1-WL test: The steps are,

<i> Initialize the nodes of G with some colors, guaranteeing the injective mapping.

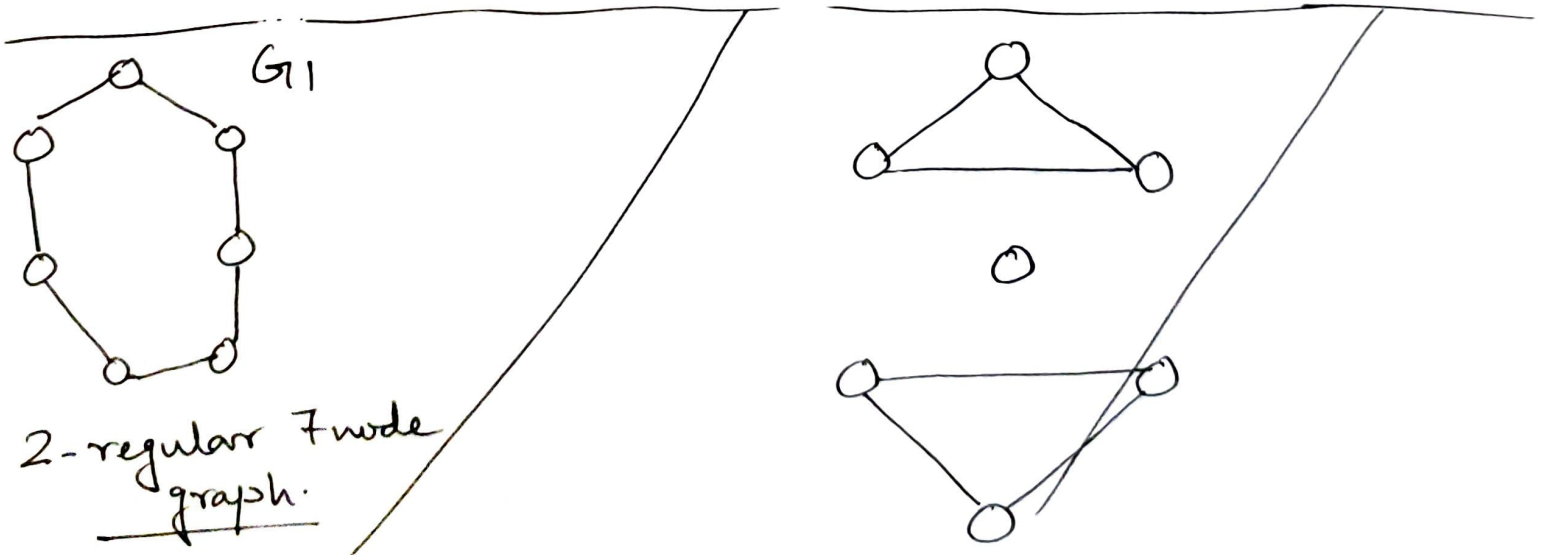
<ii> Collect colors from their neighbors. Construct an injective mapping from

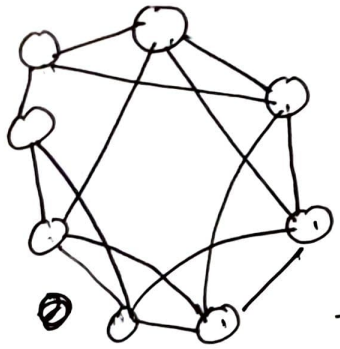
~~<iii> Check the c~~

(self color, set of colors from nbrs.) → New color.

<iii> Check the current colors. If same perform step <ii> again. If not they are not isomorphic

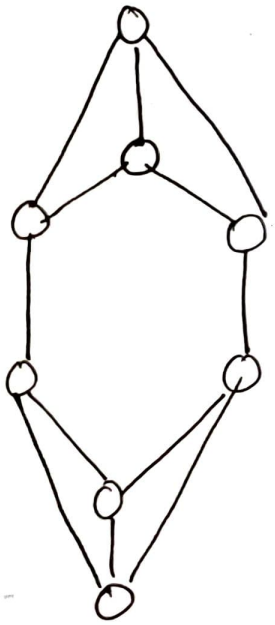
Two non-isomorphic k-regular graphs will fail 1-WL test, they will be isomorphic but 1-WL test will be inconclusive.



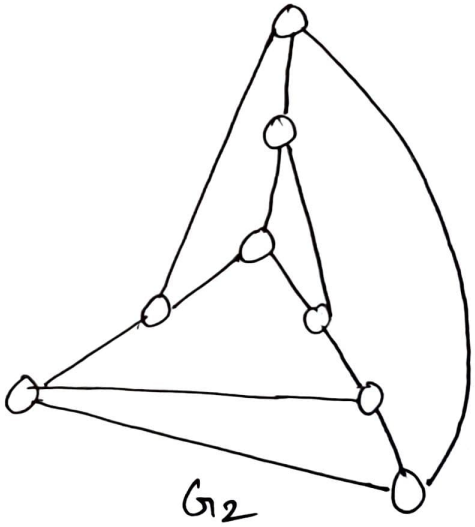


4-regular graph with 8 nodes

3-regular graph with 8-nodes



G_1



G_2

$G_1 \not\cong G_2$ because G_2 has cycles with 6 vertices that has chords but G_1 although has 6-vertex cycle but cannot have chords.

1-WL test will fail because it is 3-regular and the labelling will be similar for both of them.

6.4

Message passing based GNN's are basically as expressive as 1-WL test. As we have seen many simple graphs fail the 1-WL test.

.. To remove the limitation key tricks are to use ~~Kernel that~~ :

i> Graph kernels that was generalized to higher order k-WL tests such as Shortest-path Kernel, Motif kernel etc.

ii> ~~Intuit~~ We convert most expressiveness to more-expressiveness by defining a metric ~~dist~~ $d(\cdot,\cdot)$ such that

$$d\left((A_1,x_1),(A_2,x_2)\right) \leq \left| f(A_1,x_1) - f(A_2,x_2) \right|$$

$$\leq c\, d\left((A_1,x_1),(A_2,x_2)\right).$$

although in most of the cases suitable d and c is not known.

6.5 This problem follows the problem of ~~Set-representation~~ Set-representation.

· Note that k-layer message-passing framework is actually gains ~~from~~ information from k-hop neighbors.

Evidently we need 2-layer message-passing scheme.

Our target is to compute

$$f(S) = \sum_{\{u,v\} \in S_2} x_v x_u$$

From Zaheer. et al therefore $f(S)$ should be represented as (expressed) $f(S) = \rho\left(\sum_{x \in S} \phi(x)\right)$.

$$\phi(x) = (x, x^2)$$

$$\rho((a,b)) = \frac{1}{2}(a^2 - b)$$

s.t. $$\rho\left(\sum_{x \in S} \phi(x)\right) = \rho\left((x_1, x_1^2) + (x_2, x_2^2)\right)$$

$$= \frac{(x_1+x_2)^2 - (x_1^2 + x_2^2)}{2} = x_1 x_2$$

The layer becomes,

$$h_v^{(k+1)} \leftarrow f\left(h_v^k, \sum_{u \in N_v} \phi(h_v, h_u)\right)$$

↑ Update fn. ↑ sum-pool

Information sheet
CS59000-MLG: Computation and Machine Learning on Graphs

Assignment Submission Fill in and include this information sheet with each of your assignments. This page should be the last page of your submission. This exam is due at 1 am Nov. 12. All students please submit their homework via Brightspace. Students can typeset or scan their homework. Make sure that you answer each (sub-)question on a separate page. That is, one answer per page regardless of the answer length. Regarding the programing question, please answer them with your idea/pseudo codes. Students also need to upload their code together. Put all the code for a single question (not for each sub-question) into a single file, compress all the files and this PDF into one file (.zip) and upload the compressed file.

Late Homework Policy Not applicable.

Honor Code Not applicable. The students can check materials to answer the questions but are not allowed to discuss with other students.

Your name: Soham Mukherjee

Email: mukher26@purdue.edu **SUID:** 29920150

Discussion Group: _____

I acknowledge and accept the Honor Code.

(Signed) SM