Dong Liu

Introduction

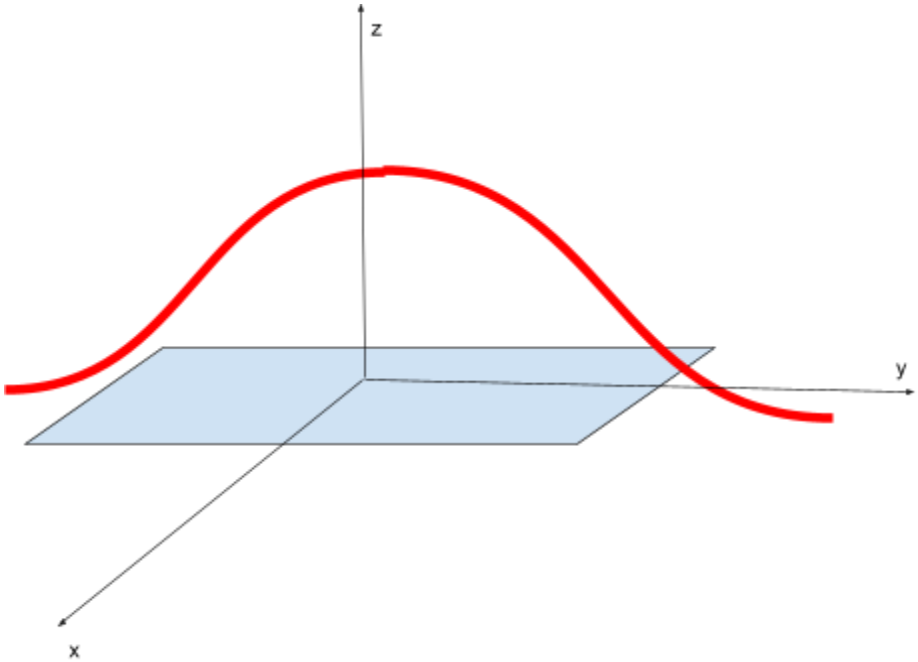
1. Information Spread and Topic Diffusion in Heterogeneous Information Networks: This work defines propagation probability in heterogeneous graphs, where only those paths with the probability higher than a predefined threshold will be used as information transmission.
2. Multi-hop Attention Graph Neural Networks: In this work, vertices and edges embedding are trained, for distances between one hop neighbors, vertex-edge-vertex pairs are used to recompute the distances; for distances between n-hop (n<=k) neighbors, they will be recomputed by considering all possible i-hop (i<=k) reachable paths between them.
3. Diffusion Improves Graph Learning: In this work, the diffusion equation is approximated by the series on infinity to obtain the approximate expression of the diffusion equation on the large graph, which speeds up the operation.
4. Diffusion in Thermodynamics: A heat transferring process where heat will exponentially decrease from the heat source to its surroundings.

Methodology

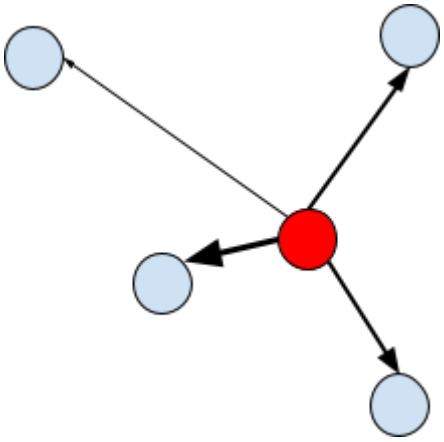
In this work, we designed a graph diffusion model that considers the `distance` and time property.

The equation of the diffusion model is like $\triangle f_t(x,y) = \frac{\partial}{\partial t} e^{-\frac{x^2+y^2}{4t}}$

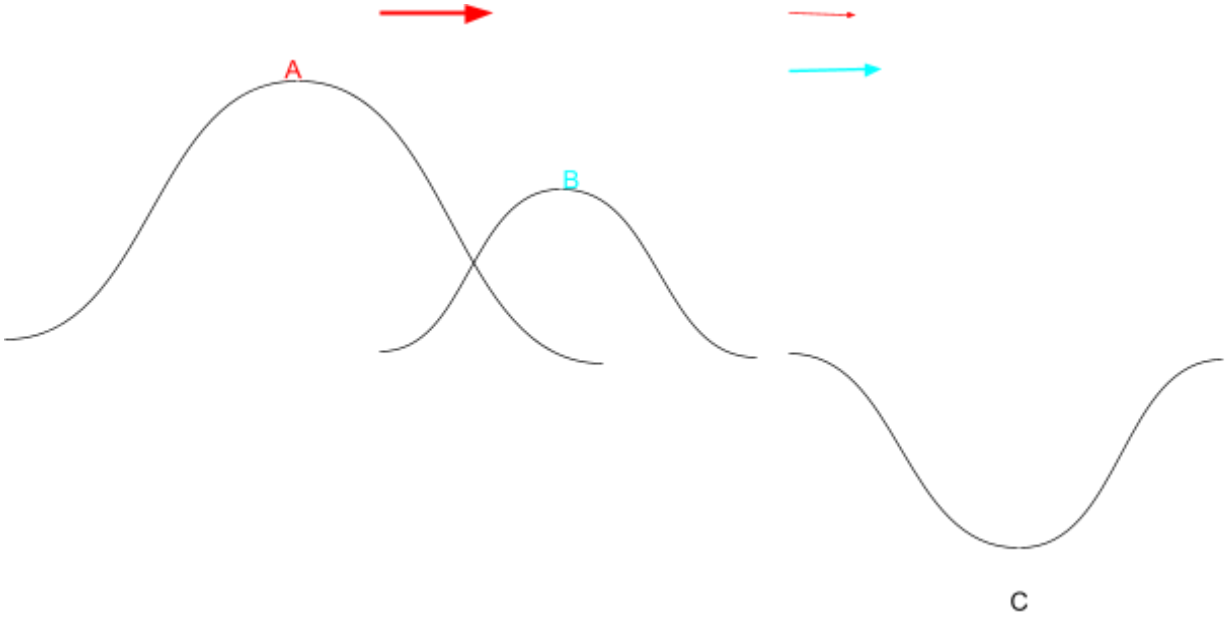
A diffusion model is like that $f_t(x,y) = \frac{1}{4\pi t} e^{-\frac{x^2+y^2}{4t}}$



According to the equation, we have developed that this equation is both considering the time but also the location info.



It is such an intuitive thought that the propagation from nodes to nodes is like the scene of heat transfer where there are multiple heat sources.



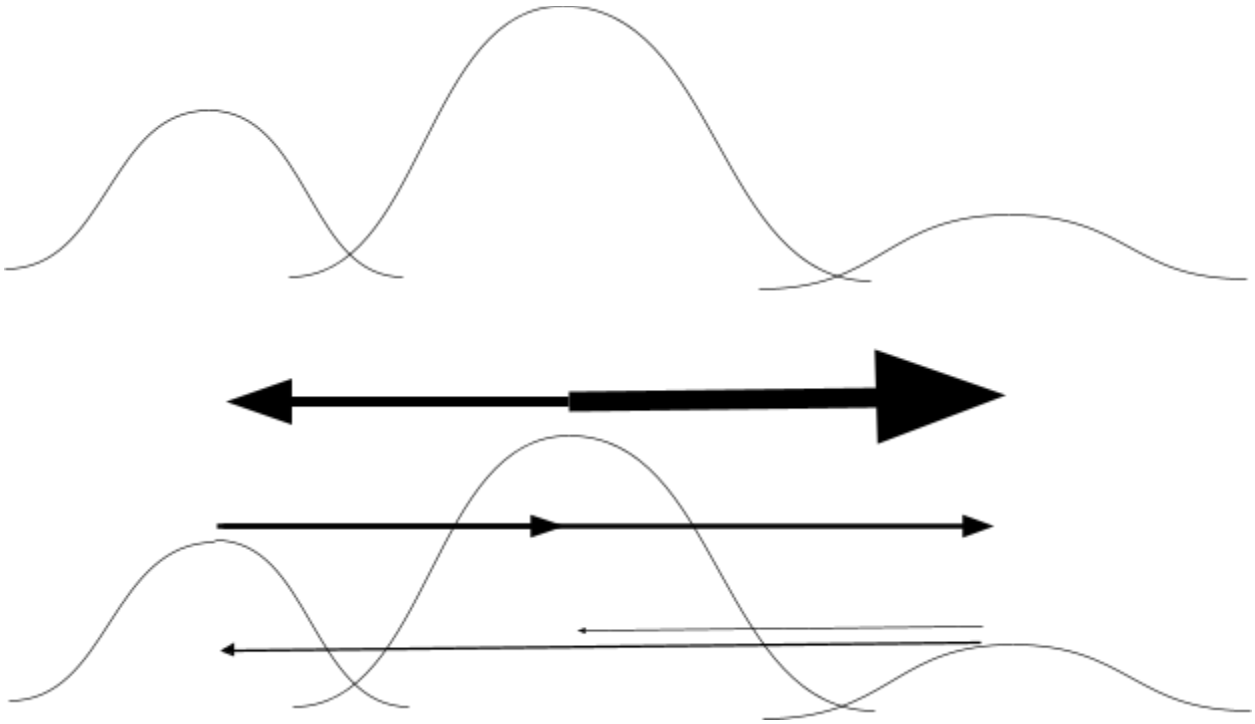
The effect of information propagation should consider the distance, traversing time, value of info, topology structure. It is an intuitive thinking that the value of one traversing info should take the neighborhood structure, the self-feature distinction, “self2neighborhood” similarity and the transferring nodes pair feature similarity into account, and those characteristics are be taken into the definition in the time, distance in the graph node-level info transferring.

Distance in the diffusion has an intuition like it spatially describes the heat descent process, the heat scalar decreases exponentially as the distance increases.

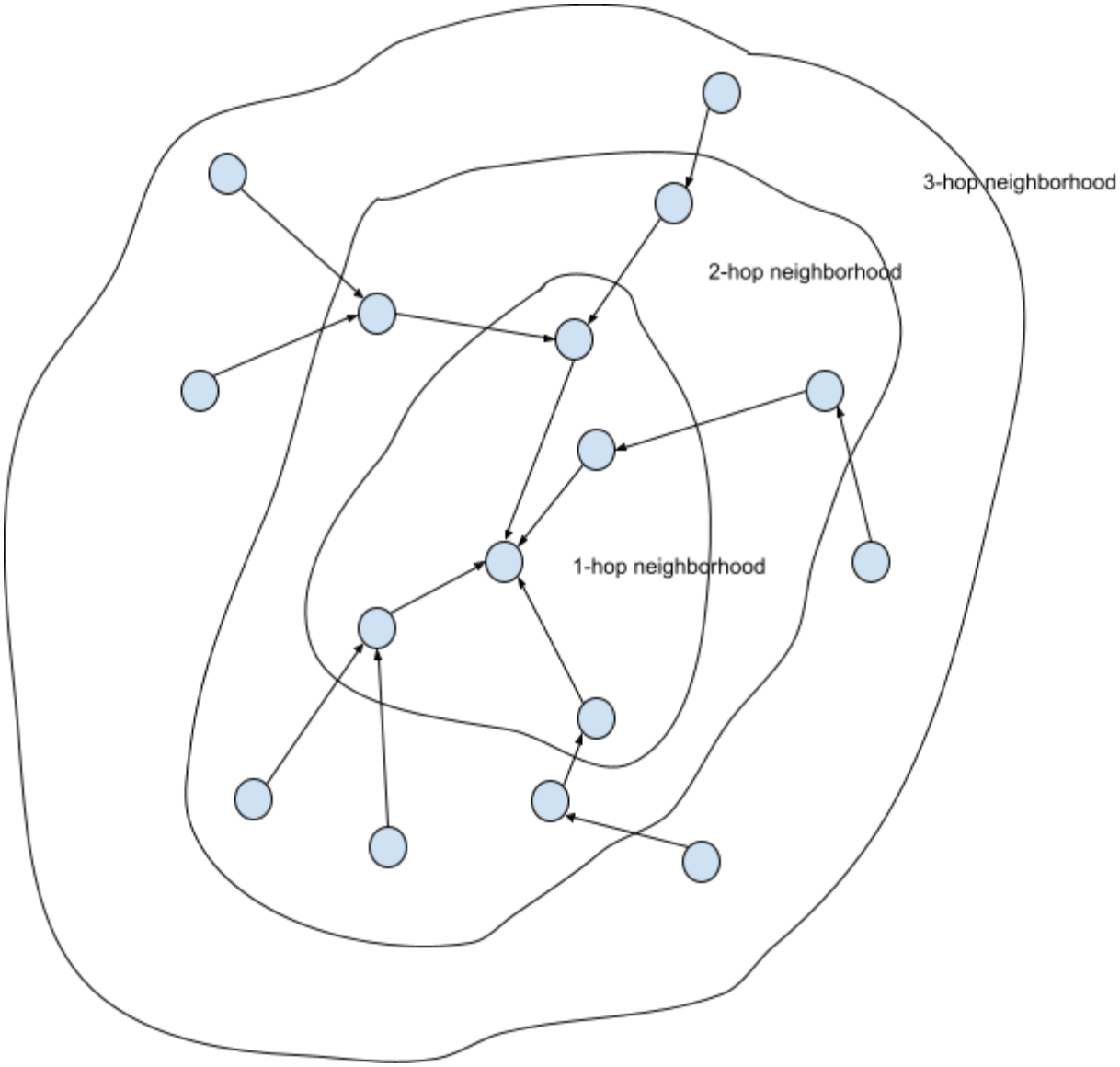
Two important factors in the diffusion, which is distance and time, are defined in considering the network topology and the importance of information. The distance should take into account the importance of transferring link importance when considering the sender and receiver node topology. Besides, the distance should consider the feature correspondence between the

feature vectors of sender and receiver nodes, it is intuitive that the transferring distance should be proportional to the feature dissimilarity. Another important factor is the time, in physics, as time elapsed, heat diffuses dynamically in space, which is similar to the information propagation dynamically in a network. The information load should be a notable metric when considering the transferring speed, a thermal scenario can be described as, the heat will transfer more quickly where the temperature is distributed sharply uneven rather than those places with smooth distribution of temperature. The differential temperature in the information propagation should consider the transferring information load and feature correspondence. we assume that one node will shoot out all the info to its neighbors at the same speed in considering the information load, but shows anisotropy when considering the feature correspondence. A well performed propagation model will not only show the neighborhood sensitivity but also the global insight, which means the transferring speed should not only view the one-hop neighbors info and one-hop neighborhood structure, which will make the “distance” and “time” have a better interpretability at macro level. The different speeds of propagation results in the asynchronous update of the original model. It is like the diffusion is a function of time and position, the heat transfer can be modeled as one static scene which is the scalar addition of the heat function of multiple heat sources and one dynamic scene where the heat source will change the temperature of itself as closer to that of its surrounding environment. In the world of thermodynamics these procedures work so fluently that it seems so natural and as if they are co-occurred. The info transferring is a time-discrete process where info propagation and update are strictly set to accurate moments to complete them. For the diffusion in the graph network, the propagation and update will show an asynchronous property rather than using a constant-time hop that organizes a bunch of points to propagate or update.

In the dynamic process of heat exchanges between heat sources, there is an idea of energy conservation, so as to the information update process should also be an interaction process, which means that after each update action the updated node should send out its new info to its neighborhood. In figure the thickness of lines shows the speed of propagation.



The details of the implementation on graphs should consider a general function that can make the info propagation and update in a local (for a node) diffusion environment then use “information conservation” for interactive update between nodes to be numerically reasonable. In considering making the update function of each node at once, a “pull” calculation structure(from neighborhood to itself) can be applied.

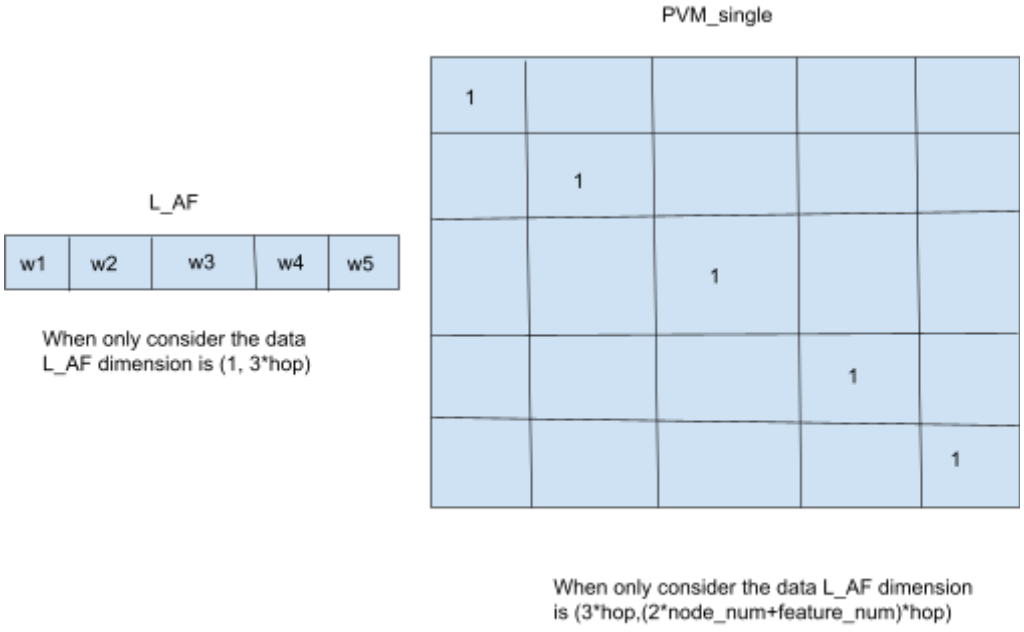


A “pull” propagation and update structure can reduce the cost of update, from remote to center, from large-hop neighborhood to close neighborhood.

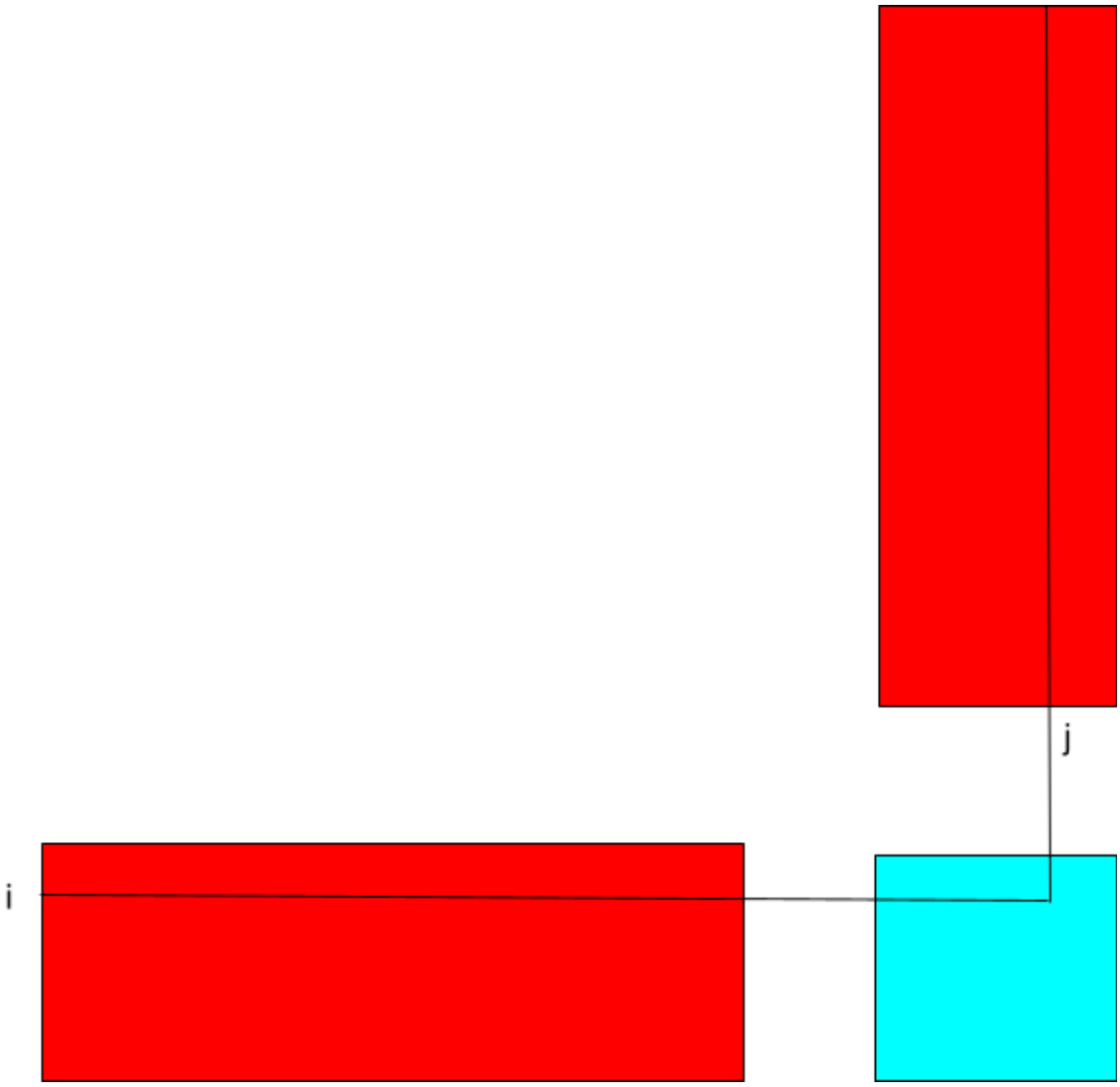
The feature space is large while the result dimension is relatively small, so the convolution network is always applied in tasks on graphs. Three matrices of convolution on graphs are always considered, adjacency matrix A, feature matrix F, and a learnable linear layer to realize convolution. A general convolution form(without considering the activation) is that $x^i = A * F * L^i$, which performs a one-hop feature neighborhood aggregation and convolution to the result. As considering the importance of links to their ending nodes, a degree based normalization is applied, $(D^{-\frac{1}{2}})^T A (D^{-\frac{1}{2}})$. This matrix normalization method has obvious shortcoming, only the one-hop degree information is used as considering the link importance without any other topology information or any feature information. The A-F matrix is introduced to fix this problem.

A	A^hop	F	F^hop	F(mean,var)	F^hop(mean,var)	D	D^hop	D(mean,var)	D^hop(mean,var)

Concatenation is used to generate a vector that contains multi-hop neighborhood local topo structure of a node and feature info of that neighbor with consideration of the hop difference. This A-F matrix or A-F vector can be used to compute the interactive importance of two nodes, where one is in the multi-hop neighborhood of another, while taking the topo and feature information into computation. Note that it is mentioned as “interactive importance”, it implements one diffusion nature that the info from a non-adjacent place can make an influence to the central node, realizing information interaction with each other in the end. In considering the normalization to make each concatenation part to have reasonable weight, a linear layer L_{AF} is suggested to be applied (In the detailed application, the linear layer L_{AF} should be multiplied with a constant matrix: position-vec matrix PVM). Besides, traditional normalization methods of each part of the A-F vectors can be used to make each part remain in a reasonable state. As mentioned above, the simple graph convolution has a general form(without considering the activation functions), $x^i = A * F * L^i$. As for A-F convolutional form $x^i = (AF * L_{AF}{}^i * PVM)^T * (AF(*)L_{AF}{}^i * PVM)$. Or, using a learnable derived matrix L_{derM}, the formula can be rewrite as $x^i = (AF(*)L_{derM})^T * (AF(*)L_{derM})$Note that (*) means the in-place multiplication and the * means matrix multiplication.
As for the dimension form for the matrices in the formula, the AF shape = (hop, (2*node_num+feature_num)*hop), while L_{derM} has the same shape as the AF matrix.

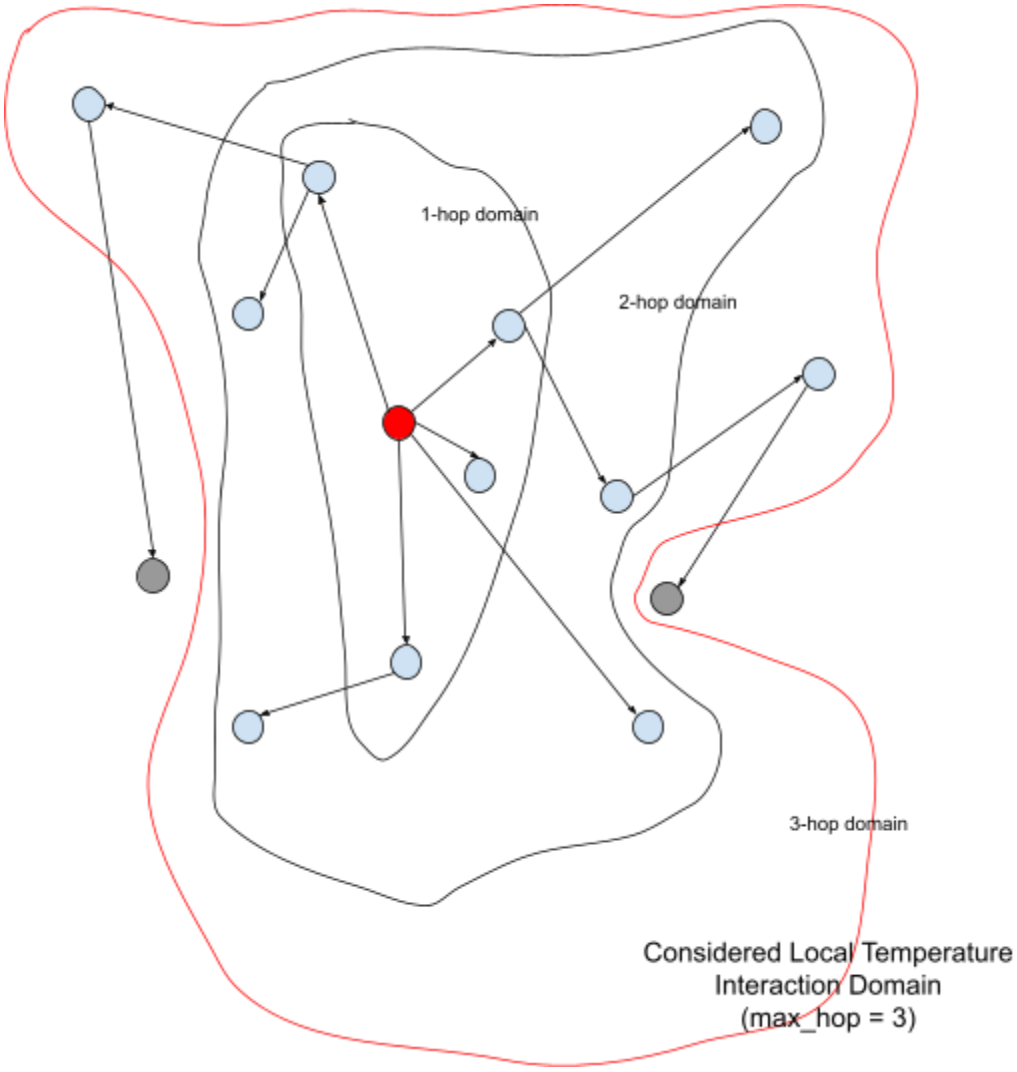


As considering the weight form of all nodes at i'th step propagation, $W^i = AF * L_{AF}{}^i * PVM$. It can be viewed as the A-F similarity computation between nodes in multi-hop neighborhood range.



From the above calculation, we got the matrix that shows a multi-hop domain interactive score. As we considered the transferring speed, in a simple case, here considered a constant-speed in a small domain. So as considering the general function of diffusion, $f = Ae^{-\frac{d^2}{t}}$. The distance is the hop number and the time can be calculated as $d = v * t$, take the distance proportional to the hop number while the transferring speed proportional to the dissimilarity between nodes when considering the topo & feature, so the final form can be represented as $f = Be^{-hop*dissim}$, as asynchronous propagation and update, for the n-hop domain, the time of update for 1,...,n hop neighbors are 1,...,1/n as considering a domain constant hop time, but if taking the transferring speed into account, we should notice that the update time of the 1,...,n neighbors

should be $\frac{1}{dissim_1}, \frac{2}{dissim_2}, \ldots, \frac{n}{dissim_n}$. Besides, information should be set to be conserved when considering the interaction.



Considering such a subgraph looks like in the figure, with regard to the red node, the blue nodes are computation active while the gray ones show an inactive computation property. In each epoch nodes will interact with its neighbors in the “Local Temperature Interaction Domain” to realize local propagation and updation. In each epoch the nodes at different distance will be calculated at different loop, and the nodes at the same loop will show an asynchronous and heat diffusion property which will be implemented as the heterogeneous diffusion time $\frac{hop-num_{j->i}}{dissim_{ij}}$ and the different head diffusion factor $f = Be^{-hop*dissim}$ when considering the topo and feature dissimilarity.

A table shows the comparison of the performance of different methods, which include: Preprocess_PPR, Preprocess_HK, HK_Kernel, PPR_Kernel, 1-hop Normalization, K-hop Normalization, Information Conservation, DiffusionDescent+AsynchronousPropagation

(k-domain), 1-hop matrix, 1-hop AF matrix(AF_Matrix), k-hop matrix(AK_Matrix), k-hop AF matrix(AFK_Matrix), ……

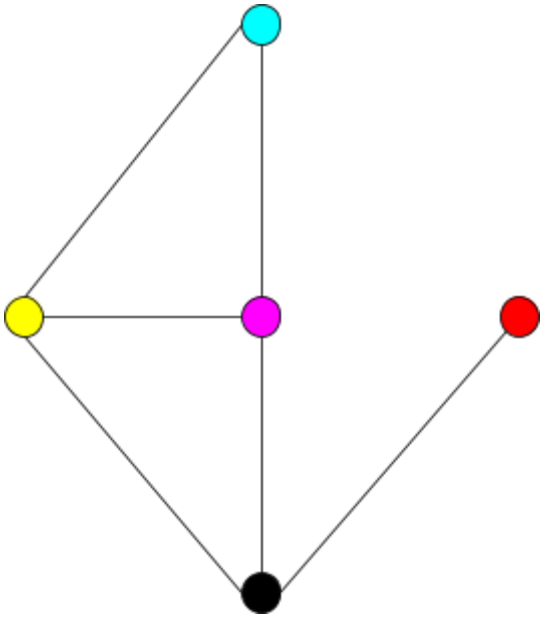
Methods	Accuracy	Parameter Setting
Preprocess_PPR+PPR_Kernel	73.04%	
Preprocess_HK+HK_Kernel	78.82%	alpha = 0.5; t = 5.0; k=128; eps = 0.0001
Preprocess_PPR+1-hop matrix+1-hop Normalization	79.45%	
Preprocess_PPR+1-hop AF matrix+1-hop Normalization	23.48%	
Preprocess_PPR+k-hop matrix+k-hop Normalization	81.73%	k = 3
Preprocess_PPR+k-hop AF matrix+k-hop Normalization	45.23%	k = 3
Preprocess_PPR+DiffusionDescent+AsynchronousPropagation	75.68%	k = 3
Preprocess_PPR+PPR_Kernel+Information Conservation	79.40%	
Preprocess_HK+HK_Kernel+Information Conservation	83.27%	alpha = 0.5; t = 5.0; k=128; eps = 0.0001
Preprocess_PPR+1-hop matrix+1-hop Normalization	80.61%	
Preprocess_PPR+1-hop AF matrix+1-hop Normalization +Information Conservation	32.64%	
Preprocess_PPR+k-hop matrix+k-hop Normalization +Information Conservation	82.57%	k = 3; alpha = 0.1
Preprocess_PPR+k-hop AF matrix+k-hop Normalization +Information Conservation	51.18%	k = 3; alpha = 0.1
Preprocess_PPR+DiffusionDescent+AsynchronousPropagation+Information Conservation	81.23%	k = 3; alpha = 0.1

Apart from the propagation anisotropy (asynchronous propagation) which is mentioned above, a topology anisotropy is introduced below, where we try to generate a “balanced” topology. In asynchronous propagation, information propagation speed is recomputed the feature dissimilarity, where previously the speed is the same in each direction (the matrix multiplication operation), the anistopic is being considered in the feature, but the topo is still to be sampled in each hop cycle, it means that the distance between is still 1,2,3,..., which is a positive integer that equals the hop number from the center node. It has been taken as granted that the distance

between nodes i and j where A_{ij} in the adjacency matrix A is 1, since the information transmission between nodes occurs in one-hop neighborhood in each propagation time, and the transmission information is recorded as the data stored in the adjacency matrix. An anisotropy of distance computation in the neighborhood is needed (the information matrix in one-hop neighborhood), more intuitively, the “imbalanced” topo computation. As for the implementation for imbalanced topo computation, the extraction and similarity computation of local topology is necessary to redefine the distance so as to reshape to topo, proximity and kernel methods are applied. Further, we think sampling and clustering are also good to compute the local structure similarity, and link prediction can be applied to link those high related vertices.

As for proximity implementation, proximity topo for each node can be defined as a k-hop neighborhood for each node, some attributes of proximity can be applied to enhance the performance of imbalance topo computation, such as denstiy. we did not perform subgraph matching tasks on local-sampled subgraphs to compute topo similarity since the computation will cause $O(V^2)$ parameters, which is unachievable for large graph applications. The recomputation of distance in considering the one-hop density is represented as $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$. As considering the multi-hop density, we introduce a decay factor ɑ to measure the importance of k-hop density. The k-hop density can be implemented as $(\mathrm{a}D + (\mathrm{a}D)^2 + \ldots + (\mathrm{a}D)^k)^{-\frac{1}{2}} A (\mathrm{a}D + (\mathrm{a}D)^2 + \ldots + (\mathrm{a}D)^k)^{-\frac{1}{2}}$.

Also, similar to information conservation that normalizes vertex information, a distance normalization can also be performed to reduce the imbalancing problem during training, but note that it is not a wise way to normalize all the vertices with their sum of edge length being the same in a specific way of computation. So it is important to make the normalization at a global level to achieve a more practical distance normalization, so an average local domain density of the graph is applied; besides, the local perturbation of density is important to make the final . The average k-hop density of the global graph can be calculated as $\rho = \frac{1}{|V|} \sum_{v \in V} (\mathrm{a}D + (\mathrm{a}D)^2 + \ldots + (\mathrm{a}D)^k)$, for a vertex that has a k-hop density of $\lambda\rho$, if $\lambda > 1$, it means that the vertex neighborhood has a density larger than the average density of the whole graph, more information,more complex info structure; while when $\lambda < 1$ means a more sparse structure.

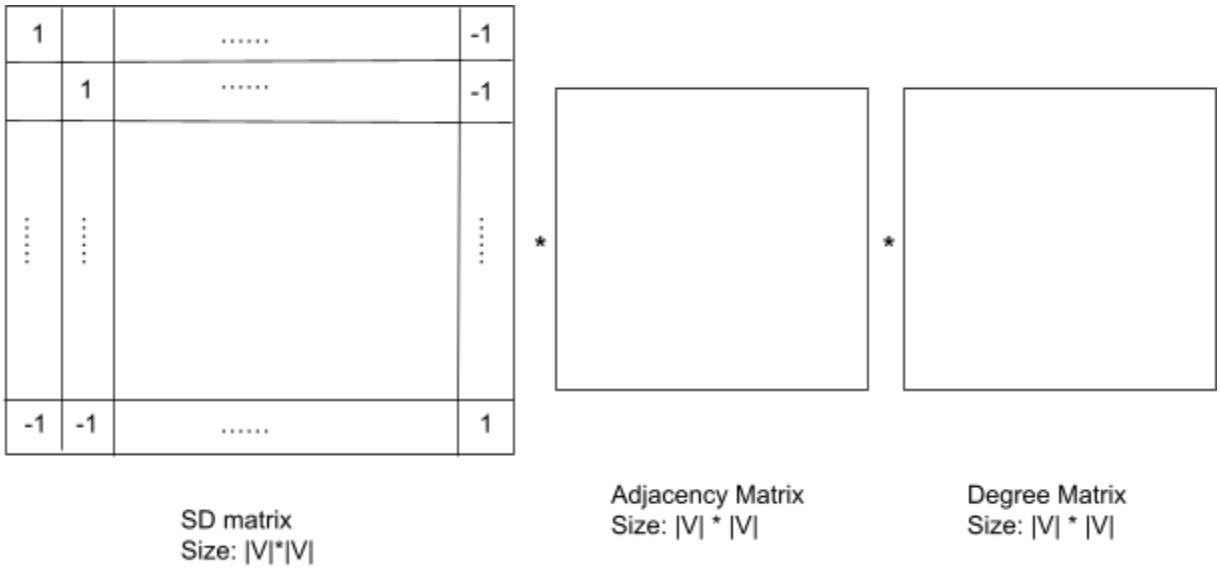


Take a specific example as in the figure, for simplicity, take $k = 1$ and $\alpha = 1$, the subgraph which is like the blue node has a total path length of $\frac{1}{\sqrt{2}} * \frac{1}{\sqrt{3}} + \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{3}} = \frac{\sqrt{6}}{3}$, the purple node has a total path length of $\frac{1}{\sqrt{3}} * \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} * \frac{1}{\sqrt{3}} + \frac{1}{\sqrt{3}} * \frac{1}{\sqrt{3}} = \frac{2}{3} + \frac{\sqrt{6}}{6} = \frac{4+\sqrt{6}}{6}$, the yellow node has a total length of $\frac{1}{\sqrt{3}} * \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} * \frac{1}{\sqrt{3}} + \frac{1}{\sqrt{3}} * \frac{1}{\sqrt{3}} = \frac{4+\sqrt{6}}{6}$. the black node has a total length of $\frac{1}{\sqrt{3}} * \frac{1}{\sqrt{1}} + \frac{1}{\sqrt{3}} * \frac{1}{\sqrt{3}} + \frac{1}{\sqrt{3}} * \frac{1}{\sqrt{3}} = \frac{2+\sqrt{3}}{3}$, the red node has a total length of the graph $\frac{1}{\sqrt{3}} * \frac{1}{\sqrt{1}} = \frac{\sqrt{3}}{3}$. The total length of all edges in this subgraph is $2 + \frac{2\sqrt{6}+2\sqrt{3}}{3}$, the ratio is like $blue: purple: yellow: black: red = \frac{\sqrt{6}}{3}: \frac{4+\sqrt{6}}{6}: \frac{4+\sqrt{6}}{6}: \frac{2+\sqrt{3}}{3}: \frac{\sqrt{3}}{3} = 1.41: 1.86: 1.86: 2.15: 1.$ While the original ratio for this is $2: 3: 3: 3: 1$. As we notice in the computation those dense space vertices local domain features are assigned with smaller length of transmission than before.

Two features will be considered to recompute the distance:

1. Local domain density (specifically, for a single vertex, it means degree)
2. Sharpness: the density/degree fluctuation in local

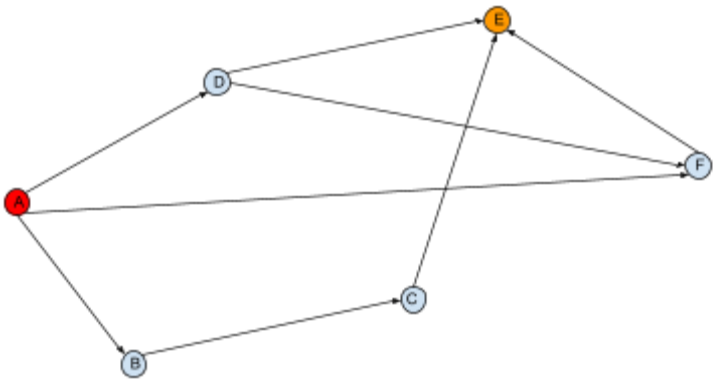
Next we will describe the calculation method of sharpness. For adjacency matrix A, if A (i, j) is not zero, it means that there is a connected edge between vertices i and j, and here we need to compute the sharpness between i and j. To calculate the degree difference, we firstly construct a matrix whose diagonal is 1 and the others are -1. Use this matrix to multiply A and then D , calculate such a degree difference matrix. Then sum each row of this matrix to find the degree fluctuation. That is, it compares to the surrounding degrees. This matrix named DegreeDelta also only has values on the diagonal, and we defined $\beta * DeltaDegree$ as H, where β is a learnable weight parameter. Besides, sharpness computation can be extended to a multi-hop domain by replacing the $A * D$ as $(\alpha A + (\alpha A)^2 + \ldots + (\alpha A)^k) * D$.



The matrix mentioned above with $\frac{1}{\lambda}$ as the diagonal matrix is P. The recalculated balance topology matrix is

$$(\alpha D + (\alpha D)^2 + \ldots + (\alpha D)^k)^{-\frac{1}{2}} * H * P * A * P * H * (\alpha D + (\alpha D)^2 + \ldots + (\alpha D)^k)^{-\frac{1}{2}}.$$

A typical work is “Multi-hop Attention Graph Neural Networks”, in this work it computes the attention scores of topology on a multi-hop domain, in another word it recomputes the distance in the multi-hop domain, The distance between two vertices will be calculated by all the k-hop reachable paths between them except those directly connected vertices. Imbalance anisotropy also has such an intuition mentioned above besides it has an improvement on it by computing all possible paths between every vertex pair in k-hop, which means that the distance of two directly connected vertices will also be recomputed. In the implementation the MAGN trained vertices and edges embedding to compute the attention score of each vertex-edge-vertex pair. So a pre-training phase can be carried out by conducting the imbalance topo anisotropy at a large-k hop before training to get a new D matrix D', which will be used at the training phase where a small-k hop is used to compute the neighbor.



Multi-Hop Distance Recomputation Based on All Possible K-Hop Reachable Paths

distance(A,E)=f(diffusion_path(A->B->C->E), diffusion_path(A->D->E), diffusion(A->D->F->E), diffusion(A->F->E))

Methods	Accuracy	Parameter Setting
k-hop reachable paths	**82.46%**	$k_r = 3, \alpha = 0.1$
k-hop reachable paths+local density	**84.27%**	$k_r = 3, \alpha = 0.1$
k-hop reachable paths+1-hop Sharpness	**75.28%**	$k_r = 3, \alpha = 0.1, \beta = 0.9$
k-hop reachable paths+local density+1-hop Sharpness	**78.03%**	$k_r = 3, \alpha = 0.1, \beta = 0.9$
k-hop reachable paths+k-hop Sharpness	**81.42%**	$k_r = 3, \alpha = 0.1, \beta = 0.9$
k-hop reachable paths+local density+k-hop Sharpness	**85.61%**	$k_r = 3, \alpha = 0.1, \beta = 0.9$
k-hop reachable paths+large k pre training	**72.84%**	$k_r = 3, \alpha = 0.1, k_t = 5$
k-hop reachable paths+large k pre training+Density+k-hop sharpness	**76.10%**	$k_r = 3, \alpha = 0.1, \beta = 0.9, k_t = 5$

Heterogeneous Graph Computation for Thermo-Diffusion

The dynamic diffusion model that mentioned above can also be applied on heterogeneous graphs, in which there are multiple types of vertices and edges in the graph. Taking the OAG dataset as an example. The venue schema describes the conference information, the paper

schema describes the basic information of one paper (authors, keywords, abstract, etc.), the author schema describes the basic information of one author.

We focused on the following methods to solve the problem.

1. JK-Attention+Kernel Methods
2. JK-Attention+propagation anisotropy
3. JK-Attention+Imbalanced Topo anisotropy
4. AF fusion & info conservation

Conclusions

The main contribution of this work can be summarized as follows:

1. Different from the current graph neural network models where every node has the same computation frequency, we developed a model to realize the asynchronous propagation and update both considering the topo & feature.
2. We created a regularization method called “information conservation”, which shows really good performance when applied to node classification, it solves the unbalancing propagation problem from a “conservation” of each feature in the whole network perspective,
3. Diffusion Descent Propagation → propagation anisotropy. We constructed a model that simulated the diffusion process in thermodynamics to realize the information propagation factor which decays with regard to distance and time. (distance means the dissimilarity between features of nodes, time is transferring time of diffusion processing)
4. Imbalanced topo computation → topology anisotropy. Different from the prevalent models that treat all neighbors as the same distance, we constructed a model that can recompute the distance which will later be used in redefining the feature matrix.
5. We applied the heat kernel to achieve complete diffused propagation.

Future Perspective:

1. We wish we can develop such a model in the near future: sampling the local structure, then combining the sampling result and a global vector to get the attention score of the local structure (also it is partly achieved by the current version of imbalance topo, but some more sensitive but computable local structure sample methods are expected), which will finally realize a information propagation model not only considered the local domain but also have global insight.