LEARNING DIVERSE GRAPH REPRESENTATION VIA MAXIMAL RATE MINIMAL ENERGY

Anonymous authors
Paper under double-blind review

ABSTRACT

We present **Maximal Rate Minimal Energy** (MRME), a principled objective enlightened by the maximum entropy principle, to learn diverse graph representations in an unsupervised manner. MRME minimizes Dirichlet energy to smooth the representation while maximizes coding rate to expand dimensionality, therefore it can intuitively encode graph topology and prevent oversmoothing. We prove that the spectra of graph Laplacian is just a special solution derived from MRME, which provides a strong theoretical justification. Such a natural connection then unifies the celebrated spectral embedding and clustering in our MRME perspective, interpreting why these smooth spectra can unroll a manifold and cut a graph in a physical way. Moreover, optimizing MRME via a basic gradient ascent scheme naturally leads to forward deep graph network, named RENet, where the graph convolution aggregates local neighbors together while the linear operator expands the global representation volume. To optimize MRME in a popular backpropagation and end-to-end manner, we further design a simple linear graph network encoder called RELGN motivated from RENet. Our preliminary simulations and experiments on both RENet and RELGN clearly show the effectiveness of MRME.

1 INTRODUCTION

Graph is a ubiquitous structure to model complex real-world data scenes. To perform machine learning on graph, represent each node as a low-dimensional vector is the first step. To facilitate the subsequent graph data mining tasks, such as node classification and link prediction, seeking a "good" graph representation is always the primary challenge.

Typically, a graph can be described as $\mathcal{G} = (\mathcal{V}, \boldsymbol{A}, \boldsymbol{X})$, where $\mathcal{V}$ denotes the node set, $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ is an adjacency matrix (binary or weighted) with respect to the graph structure and $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\} \in \mathbb{R}^{m\times D}$ is the feature matrix with $\boldsymbol{x}_i \in \mathbb{R}^D$ being the feature vector of node i. Learning representation for graph $\mathcal{G}$ then involves designing a continuous mapping or encoder $f(\cdot)$ such that each node $i \in \mathcal{V}$ can be represented as a low-dimensional vector $\boldsymbol{z}_i \in \mathbb{R}^d$:

$$\boldsymbol{Z} = f(\boldsymbol{X}, \boldsymbol{A}, \theta). \tag{1}$$

Here, $\theta \in \Theta$ is the parameters and $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m\} \in \mathbb{R}^{m\times d}$ is the graph representation matrix. According to the encoder, graph representation methods can be roughly partitioned into two categories.

Shallow embedding methods focus on designing an objective so as to optimize an embedding matrix $\boldsymbol{Z}$ directly. Traditional spectral methods, such as Laplacian eigenmaps and spectral clustering, use the spectra of graph Laplacian for embedding and clustering. Due to the explicit singular value decomposition (SVD), these methods face with the problems of scalability and performance. Random-walk based shallow embedding methods are mainly motivated from the skip-gram model in word embedding. [work embedding, 2018] shows that these methods are closely related to matrix-factorization methods. However, these methods take a stochastic gradient scheme to optimize $\boldsymbol{Z}$, therefore they have a much lower complexity when processing large-scale graph.

As these methods focus on optimizing an embedding matrix $\boldsymbol{Z}$ directly, therefore the encoder behind these methods can be seen as a linear encoder Wang & Leskovec (2020) that takes the one-hot vector

of node i as input:

$$\boldsymbol{z}_i = \boldsymbol{Z}^T \mathbf{1}_i = f(\mathbf{1}_i, \boldsymbol{Z}). \tag{2}$$

Here the embedding matrix $\boldsymbol{Z}$ can be seen as the parameters of encoder $f(\cdot, \theta)$. Such an encoder then suffers some drawbacks. First, they can not leverage the node features matrix $\boldsymbol{X}$. More importantly, the encoder is lack of parameter sharing, which is statistically inefficient.

Graph Neural Network based methods transform the input original feature to the final output representation by a message passing neural network where the graph structure is encoded in the forward propagation.

$$\boldsymbol{z}_i = GNN(\mathbf{1}_i, \boldsymbol{Z}). \tag{3}$$

As such, the encoder of GNN exploits the feature information and graph structure together to generate node representation. Although GNN have achieved great success in many graph analysis task, it also faces with some theoretical problems. Some theoretic analysis have proven the power of GNN comes from the low-frequency passing filter. As a result, the GNN suffers from over-smoothing problem when put in more layers. The second problem is the interpretation. Since the back propagation training paradigm is from deep learning, the interpretation of the linear operators, the graph attentions, even the designed architecture is lost.

Recently, some authors show that a white-box deep network ReduNet can be derived naturally from optimizing the objective of the maximal coding rate reduction (MCR^2), which is designed to learn linear discriminant representation for data sampling from a mixed distribution. Therefore, it means that a good objective and an effective encoder are coming together. Motivated from these encouraging works, this paper studies the graph representation learning from a principled way. Our research methodology and some main contributions is summarized as follows:

1. We first show that Dirichlet energy play a key role in graph representation
2. We then argue that, given a conserved quantity Dirichlet energy, the distribution $\mathcal{D}_{\boldsymbol{z}}$ of an informative representation $\boldsymbol{Z}$ should have a maximum entropy. This strong hypothesis is enlightened from the maximum entropy principle, it means that we

   We argue that the distribution $\mathcal{D}_{\boldsymbol{z}}$ of a "good" graph representation should follow the maximum entropy principle. It means that, given a conserved quantity of Dirichlet energy, the entropy $H(\boldsymbol{z})$ should be maximized so as to learn a diverse graph representation to prevent oversmoothing.
3. As entropy is not well defined in continue variable, we take the coding rate as a computable measure to estimate the minimal number of bits to encode a subspace-like distribution. We then prove the spectra of graph Laplacian is the optimal solution under a special conserved quantity. Such a theoretical connection give a strong justification of our argument.
4. The Lagrange function of the constraint model then derive a objective: Maximal Rate Minimal Energy (MRME). Directly optimizing MRME via a basic gradient ascent scheme then derive a natural deep graph network called RENet, which then inspires us to design a linear graph network called RELGN so as to optimize MRME in a backpropagation and end-to-end manner.
5. Our preliminary simulations have clearly verified the effectiveness of MRME to learn diverse graph representation.

2 OVERSMOOTHING PROBLEM

Our discussion begin at the Dirichlet energy. This section we will show that Dirichlet energy is a double-edged sword in graph representation learning: minimizing it smooths the representation to encode graph topology, on the other hand, minimizing it too much also causes the curse of oversmoothing. We clarify how previous methods solve this problem in different ways.

Generally, for a differentiable mapping $\boldsymbol{z} = f(\boldsymbol{x}, \theta)$, Dirichlet energy is a non-negative quantity to measure how variable mapping f is over a open set Ω:

$$E(f) = \frac{1}{2} \int_{\Omega} \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \theta)\|^2 d\boldsymbol{x}. \tag{4}$$

For graph $\mathcal{G}$, given a weighted adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ and a graph representation matrix $\boldsymbol{Z}$, the Dirichlet energy corresponding to $(\boldsymbol{Z}, \boldsymbol{A})$ is defined as:

$$E(\boldsymbol{Z}, \boldsymbol{A}) = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} \boldsymbol{A}_{i,j} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 = tr(\boldsymbol{Z}^T \boldsymbol{L} \boldsymbol{Z}), \tag{5}$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ and $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{A}\boldsymbol{1})$ is the diagonal degree matrix. To simplify our discussion, in this paper we consider the symmetric normalized graph Laplacian $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{A}$ where $\boldsymbol{A} := \boldsymbol{D}^{-1/2} \boldsymbol{A} \boldsymbol{D}^{-1/2}$. It is intuitive that minimizing Dirichlet energy $E(\boldsymbol{Z}, \boldsymbol{A})$ smooths representation $\boldsymbol{Z}$ to encode the graph structure, as the distance between $(\boldsymbol{z}_i, \boldsymbol{z}_j)$ tends to be small if $(i, j) \in \mathcal{E}$. However, minimizing $E(\boldsymbol{Z}, \boldsymbol{A})$ too much also erases the graph topology from $\boldsymbol{Z}$. As we can see, every $\boldsymbol{Z}^*$ satisfying $\boldsymbol{z}_i^* = \boldsymbol{z}_j^*$ if $(i, j) \in \mathcal{E}$ is a global minimum due to $E(\boldsymbol{Z}^*, \boldsymbol{A}) = 0$. Therefore, for a connected graph $\mathcal{G}$, the resulting $\boldsymbol{Z}^*$ will collapse to one point, which is known as oversmoothing. Previous methods alleviate the curse of oversmoothing from three ways: the constraint, the objective and the encoder structure.

2.1 Constraint on the Stiefel Manifold

To avoid dimensional collapse, classic dimensionality reduction methods require the representation $\boldsymbol{Z}$ to satisfy the orthogonal constraint $\boldsymbol{Z}^T \boldsymbol{Z} = \boldsymbol{I}$, which is known as Stiefel manifold $\mathcal{M}$. Therefore, spectral embedding methods like Laplacian eigenmaps and spectral clustering can be seen to minimize Dirichlet energy over the Stiefel Manifold:

$$\min_{\boldsymbol{Z}} \; E(\boldsymbol{Z}, \boldsymbol{A}), \quad \textit{subject to} \quad \boldsymbol{Z} \in \mathcal{M}. \tag{6}$$

Suppose that $\boldsymbol{Z}$ is centralized, then every $\boldsymbol{Z} \in \mathcal{M}$ have two properties: 1) Each dimension of $\boldsymbol{Z}$ have a same fixed variance. 2) Different dimensions is uncorrelated. As such, Stiefel Manifold $\mathcal{M}$ can removes those over-smoothing points.

Surprisingly, despite (6) is a non-convex problem, its optimal solution is those smooth spectra of graph Laplacian $\boldsymbol{L}$. In this paper we show that these spectra is a special solution of the our framework, which then provides an unifying perspective to understand spectral embedding and clustering intuitively: why these smooth spectra can unroll a manifold and cut a graph into several subgraphs.

2.2 Contrastive Objective

Contrasive learning try to solve the oversmoothing problem by introducing the negative sampling loss. Random-walk based methods introduce the skip-gram model from natural language processing to graph representation learning.

The key idea of these methods is to optimize the embedding to be similar if they tend to co-occur on short random walk:

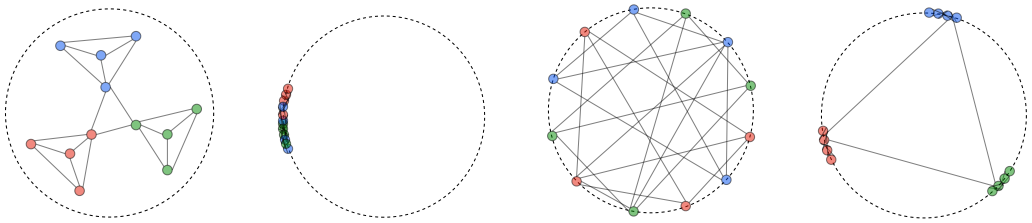
$$\min_{\boldsymbol{Z}} \sum_{(i,j) \in \mathcal{E}(T)} -\log(\sigma(\boldsymbol{z}_i^T \boldsymbol{z}_j)) - k \cdot \mathbb{E}_{t \sim \mathcal{P}_n(\mathcal{V})} \left[\log(-\sigma(\boldsymbol{z}_i^T \boldsymbol{z}_t))\right], \tag{7}$$

where $\sigma(\cdot)$ is the logistic function, $\mathcal{P}(\mathcal{V})$ is a negative sampling distribution and $\mathcal{E}(T)$ is the set of random walks sampling from edge set $\mathcal{E}$ on a T-length. Suppose we have $\|\boldsymbol{z}\| = 1$, then the first term will minimize the graph Dirichlet energy while the negative sampling tends to pull away different points. As such, the learned representation can avoid oversmoothing problem.

Recently, there are also some work try to solve this problem by negative sampling, which involves the following objective:

$$\min \; tr(\boldsymbol{Z}^T \boldsymbol{L} \boldsymbol{Z}) - \lambda \, tr(\boldsymbol{Z}^T \boldsymbol{L}^- \boldsymbol{Z}), \tag{8}$$

where $\boldsymbol{L}^-$ is the Laplacian of the negative graph $\boldsymbol{A}^-$. In graph embedding framework, the MDA utilize the labeled information to constructed such a negative graph, expecting the corresponding graph embedding having a large margin between different class. The recent work contrasive Laplacian eigenmaps adapts a random selection scheme to negative sampling.



(a) input graph (b) low entropy (c) high entropy (d) keep a balance

Figure 1: Different entropy.

2.3 ENCODER

GNN try to design a parametric mapping $f(\boldsymbol{X}, \mathcal{E}, \theta)$. Linear GNN design the encoder as $\boldsymbol{Z} = f(\boldsymbol{X}, \mathcal{E}, \theta) = g(\boldsymbol{L})\boldsymbol{X}\boldsymbol{W}$.

Denoting symmetric adjacency matrix as $\bar{\boldsymbol{A}} = (\boldsymbol{A} + \boldsymbol{A}^T)/2$, then the gradient of Dirichlet energy is given by

$$\frac{1}{2}\nabla_{\boldsymbol{Z}} E = \bar{\boldsymbol{L}}\boldsymbol{Z}, \quad \textit{where} \quad \bar{\boldsymbol{L}} = \boldsymbol{I} - \bar{\boldsymbol{A}}. \tag{9}$$

Minimizing Dirichlet energy via gradient descent scheme then leads to a simple update formulation:

$$\boldsymbol{Z}^{k+1} := \boldsymbol{Z}^k - \eta\bar{\boldsymbol{L}}\boldsymbol{Z}^k = (1-\eta)\boldsymbol{Z}^k + \eta\bar{\boldsymbol{A}}\boldsymbol{Z}^k = \hat{\boldsymbol{A}}\boldsymbol{Z}^k, \tag{10}$$

where renormalized adjacency matrix $\hat{\boldsymbol{A}} = (1-\eta)\boldsymbol{I} + \eta\bar{\boldsymbol{A}}$, which is the symmetric adjacency matrix $\bar{\boldsymbol{A}}$ with added self-connections. Now consider the linear GNN:

$$\boldsymbol{Z}^K = f(\boldsymbol{X}, \mathcal{E}, \theta) = \hat{\boldsymbol{A}}^K\boldsymbol{X}\boldsymbol{W}. \tag{11}$$

we then can say it is strictly equivalent to perform K-iteration gradient optimization to descend Dirichlet energy, with the initial point $\boldsymbol{Z}^0 = \boldsymbol{X}\boldsymbol{W}$. Therefore, as the K increasing, the learned representation inevitably suffer from over-smoothing problem. There are plenty of works try to design a deep GNN encoder to avoid over-smoothing.

2.4 TOWARDS DIVERGE GRAPH REPRESENTATIONS

As we have discuss above, previous methods all try to optimize the representation such that the local neighbors have similar embeddings, and then prevent oversmoothing either from importing constraint, or from contrastive objective, or from designing encoder. Therefore, it is now clearly intuitive that a "good" graph representation should have two basic properties:

1. *Local Smoothness*: Neighbors should have similar representations so that the graph topology can be encoded into the learned vector space.
2. *Global Diversity*: The total dimension of the learned representation should be expanded as large as possible, which prevents the oversmoothing problem and therefore leads to a maximally informative graph representation.

To some extent, smoothness and diversity are two conflicting objectives: a locally smooth representation tends to be oversmooth in global, which violates global diversity. Conversely, when the representation is diverse globally, it is more likely to break the smoothness of local neighbors. The discussion above have convinced us that Dirichlet energy $E(\boldsymbol{Z}, \boldsymbol{A})$ gives a simple and good-enough measure to quantify and optimize local smoothness. The rest of this paper we will focus on how to naturally derive a diverse graph representation from the maximum entropy principle.

3 METHODOLOGY

In this section we introduce the maximum entropy principle to learn diverse graph representation. Then we utilize the coding as a computable measure to estimate entropy from a finite number of

samples, and we show that the smooth spectra of graph Laplacian can be derived from our framework naturally. Finally, we show that Lagrangian function of the constrained model derive our principle objective Maximal Rate Minimal Energy.

3.1 MAXIMUM ENTROPY PRIOR DISTRIBUTION

Generally, we consider a prior distribution $\mathcal{P}(\mathbf{Z})$ where the graph representation $\boldsymbol{Z}$ is sampled from, and a conditional probability distribution $\mathcal{P}(\boldsymbol{Z} \mid \boldsymbol{A})$ when given the graph structure $\boldsymbol{A}$. The discussion above have convinced us that Dirichlet energy $E(\boldsymbol{Z}, \boldsymbol{A})$ gives a simple and good-enough measure to quantify and optimize local smoothness of the representation $\boldsymbol{Z}$. Here we generalize Dirichlet energy to a random matrix $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\} \in \mathbb{R}^{m \times d}$ corresponding to the graph structure $\boldsymbol{A}$:

Definition: Given a adjacency matrix $\boldsymbol{A}$, the Dirichlet energy of random matrix variable $\mathbf{Z}$ is defined as follows:

$$E_{\boldsymbol{A}}(\mathbf{Z}) \doteq \int E(\boldsymbol{Z}, \boldsymbol{A})\, \mathcal{P}(\boldsymbol{Z} \mid \boldsymbol{A})\, d\mathbf{Z}. \tag{12}$$

Subject to a conserved Dirichlet energy quantity $E_{\boldsymbol{A}}(\mathbf{Z}) \leq c$, a natural question to learn diverse graph representation is that which prior distribution $\mathcal{D}_{\mathbf{Z}}$ should be preferred. The answer here is enlightened by the maximum entropy principle. According to this principle, the maximum entropy distribution that satisfies a set of given conserved quantities is the best choice. From the Occam's razor perspective, maximum entropy makes fewest assumption about the prior distribution $\mathcal{D}_{\mathbf{Z}}$. In our case here, it means to admit the most ignorance beyond the conserved Dirichlet energy quantity.

For a joint distribution, we always have $H(\mathbf{Z}) \leq H(\mathbf{z}_1) + \cdots + H(\mathbf{z}_m)$ where the equality holds if and only if each random vector $\mathbf{z}$ is independent from the others. Therefore, maximizing $H(\mathbf{Z})$ will result in a set of independent random vector $\{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ with the same maximum entropy distribution $\mathcal{D}_{\mathbf{z}}$. At this time we have $H(\mathbf{Z}) = mH(\mathbf{z})$. The optimization problem we consider here then can be formulated as follows:

$$\max\ H(\mathbf{z}) \quad \textit{subject to} \quad E_{\boldsymbol{A}}(\mathbf{Z}) \leq c. \tag{13}$$

We argue that maximizing the entropy of the distribution $\mathcal{D}_{\mathbf{z}}$ can naturally prevent oversmoothing. To give a intuitive sense, suppose that the distribution $\mathcal{D}_{\mathbf{z}}$ is discrete and supported on an unit circle that is equally divided into m parts, and consider two extreme cases of the distribution $\mathcal{D}_{\mathbf{z}}$:

- *Minimum Entropy*: In this case $H(\mathbf{z}) \to 0$ the probability mass is concentrated in a limit range, therefore the i.i.d. samples $\boldsymbol{Z}$ will collapse together, which worsen the oversmoothing problem, as shown in figure 1b.
- *Maximum entropy*: With increasing entropy $H(\mathbf{z}) \to \log m$, the probability mass distributes more uniformly on the unit circle, therefore the generated samples $\boldsymbol{Z}$ is more likely to be diverse, as shown in figure 1c.

Figure 1d shows that what a good graph representation $\boldsymbol{Z}$ that satisfies local smoothness and global diversity looks like. In this ideal case, As we can see, each sample $\boldsymbol{z}$ is sampling from a distribution $\mathcal{D}_{\mathbf{z}}$ with relatively large entropy, while taken the $\boldsymbol{Z}$ as a whole sampled from $\mathcal{P}(\mathbf{Z}, \boldsymbol{A})$, it have a relatively small Dirichlet energy.

3.2 OBJECTIVE OF MAXIMAL RATE MINIMAL ENERGY

Instead of caring the probability distribution $p(\boldsymbol{z})$ and $\mathcal{P}(\mathbf{Z}, \boldsymbol{A})$, here we will derive an optimization problem with respect to the representation matrix $\boldsymbol{Z}$ from the theoretical model (13). Similarly, given a representation matrix $\boldsymbol{Z}$, the Dirichlet energy $E(\boldsymbol{Z}, \boldsymbol{A})$ is constrained to a fixed quantity c. The problem then is about how to estimate the entropy $H(\mathbf{z})$ from the observed samples $\boldsymbol{Z}$. For a discrete random variable, entropy is the minimum number of bits needed to encode the variable under the optimal coding scheme.

Suppose the $\mathcal{D}_{\mathbf{z}}$ is a subspace-like distribution, the author in xx provides the coding rate to given a precise estimate on the average number of bits needed to encode each $\boldsymbol{z} \in \boldsymbol{Z}$ up to precision ϵ:

$$R(\boldsymbol{Z}, \epsilon) \doteq \frac{1}{2} \log\det\left(\boldsymbol{I} + \frac{d}{m\epsilon^2} \boldsymbol{Z}^T \boldsymbol{Z}\right). \tag{14}$$

Here we choose coding rate as a computable measure to estimate the entropy $H(\boldsymbol{z})$. As such, we can derive a constrained matrix optimization problem:

$$\max_{\boldsymbol{Z}} \; R(\boldsymbol{Z}, \epsilon) \quad \textit{subject to} \quad E(\boldsymbol{Z}, \boldsymbol{A}) \leq c. \tag{15}$$

In [], coding rate is used as a measure of the dimension of representation $\boldsymbol{Z}$. From this perspective, the model (15) have a clear geometric meaning: Subject to a conserved quantity of Dirichlet energy, the volume of the representation $\boldsymbol{Z}$ should be as large as possible.

Integrated view for spectral embedding and clustering. Surprisingly, we show that the smooth spectra can be derived from model (15) when the conserved Dirichlet energy quantity c is up to the sum of the d smallest eigenvalues of graph Laplacian. Formally, we have the following theorem:

Theorem 3.1. *Suppose that $\|\boldsymbol{Z}\|_F^2 = d$, $\{\lambda_i\}_{i=1}^d$ are the d smallest eigenvalues of graph Laplacian and $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d]$ are the corresponding eigenvectors. When the conserved quantity $c^* = \sum_{i=1}^d \lambda_i$, then the optimal solution of model (15) is $\boldsymbol{Z}^* = \boldsymbol{U}\boldsymbol{R}$ where $\boldsymbol{R} \in \mathbb{R}^{d\times d}$ is an orthogonal matrix.*

Theorem (3.1) gives a strong justification for model (15), as it establish a natural connection with the spectra of graph Laplacian. In other words, these smooth spectra are the representations that satisfy a special constrained Dirichlet energy c^* while have a maximum coding rate. Model (15 then can be viewed as a generalization beyond the spectra of graph Laplacian. As a result, we can now understand the spectral embedding and clustering in an integrated view:

- *Laplacian Eigenmaps*: In manifold learning, the constrained Dirichlet energy guarantees the representation $\boldsymbol{Z}$ to preserve the neighbor structure of a manifold, while a maximum coding rate make sure the $\boldsymbol{Z}$ have a large dimension so that it can unroll the manifold.
- *Spectral Clustering*: In clustering, the constrained Dirichlet energy guarantees the node representation in a dense subgraph to bind more tightly, while maximizing coding rate will exert a force to separate different dense subgraphs away from each other.

Figure 1 give a visualization example for these two celebrated methods. Now we introduce the Lagrange multiplier $\lambda \geq 0$ to derive an unconstrained optimization problem:

$$\max_{\boldsymbol{Z}} \; \mathcal{L}(\boldsymbol{Z}, \boldsymbol{A}, \epsilon, \lambda) = R(\boldsymbol{Z}, \epsilon) - \lambda\, E(\boldsymbol{Z}, \boldsymbol{A}), \tag{16}$$

where a large Lagrange multiplier λ corresponds to a small conserved quantity c and vice versa. Therefore, we can tune λ to control the amount of Dirichlet energy. The Lagrange function $\mathcal{L}$ then provides a principled objective to learn diverse graph representation: Maximal Rate Minimal Energy (MRME).

3.3 Properties of MRME on $\mathbb{S}^{d-1}$

Some recent works show that optimizing the representation in a hypersphere is more effective. Rather than normalizing the scale by $\|\boldsymbol{Z}\|_F^2 = d$, here we will renormalize each node representation $\boldsymbol{z}$ to have a unit length, as in spectral clustering []. To facilitate the subsequent tasks, e.g. clustering, classification and link prediction, the learned graph representation should well capture the cluster structure of the graph, meaning that $\boldsymbol{Z}$ have a small within-cluster distance and a large between-cluster distance. On the surface of an unit sphere, the cosine similarity gives a natural "distance" measure that is equivalent to the Euclidean distance. We now show that optimizing MRME on $\mathbb{S}^{d-1}$ can encode the cluster structure into the cosine similarity of $\boldsymbol{Z}$. Formally, we have the following theorem:

Theorem 3.2. *Suppose the graph $\boldsymbol{A}$ has k connected components, i.e. $\operatorname{rank}(\boldsymbol{L}) = k$, the optimal representation $\boldsymbol{Z}^*$ that maximizes MRME on $\mathbb{S}^{d-1}$ ($d \geq k$) will reach an orthogonal structure as $\lambda \to +\infty$, stated as follows:*

- Maximum Within-cluster Similarity*: When node i, j are from a same connected component, the cosine similarity* $\cos(\boldsymbol{z}_i^*, \boldsymbol{z}_j^*) \to 1$.
- Minimum Between-cluster Similarity*: When node i, j are from different connected components, the cosine similarity* $\cos(\boldsymbol{z}_i^*, \boldsymbol{z}_j^*) \to 0$.

In other words, there exist k mutually orthogonal vectors in the vector space $\mathbb{R}^d$, say $\boldsymbol{Y} \in \mathbb{R}^{k\times d}$ subject to $\boldsymbol{Y}\boldsymbol{Y}^T = \boldsymbol{I}$, so that as the conserved Dirichlet energy $c \to 0$, the learned representation $\boldsymbol{Z} = \boldsymbol{Z}_1^* \cup \cdots \cup \boldsymbol{Z}_k^*$ will cluster around $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$ more closely. If the nodes is drawn from a mixture distributions $\mathcal{D} = \{\mathcal{D}_j\}_{j=1}^k$, Theorem 3.2 shows that optimizing the unsupervised MRME objective on $\mathbb{S}^{d-1}$ have the potential to learn a discriminative representation $\boldsymbol{Z}$. In other words, if the graph $\boldsymbol{A}$ have k densely connected clusters $\{\mathcal{S}_{j=1}\}_{j=1}^k$ where cluster $\mathcal{S}_j$ is drawn from distribution $\mathcal{D}_j$, then the learned representation $\boldsymbol{Z}$ is tightly-bound within cluster while well-separated between clusters.

Comparison to SNE on $\mathbb{S}^{d-1}$. Given a precomputed graph structure $\boldsymbol{A}$, stochastic neighbor embedding (SNE) minimizes the Kullback-Leibler divergence (also called relative entropy) between two distribution $(\boldsymbol{A}, \boldsymbol{Q})$, where $\boldsymbol{Q}$ is the affinity matrix computed from the representation $\boldsymbol{Z}$.

Stochastic neighbor embedding (SNE) is a successful nonlinear dimensionality reduction method widely used for visualizing data. SNE first computes a probability matrix $\boldsymbol{A}$ from original features, and then minimizes the Kullback-Leibler divergence between $\boldsymbol{A}$ and $\boldsymbol{Q}$ where $\boldsymbol{Q}$ is computing from the low-dimensional representation $\boldsymbol{Z}$. As such, $\boldsymbol{Z}$ is expected to encode the neighbor structure of original features into the representation space. We now show that, although with a totally different motivation, SNE shares a very similar idea and objective to MRME. To derive a more clear formulation for comparison, we consider $\boldsymbol{Z} \subset \mathbb{S}^{d-1}$, and then SNE optimizes the following objective:

$$\min_{\boldsymbol{Z}} \sum_{i=1}^{m} \mathrm{KL}(\boldsymbol{a}_i \,\|\, \boldsymbol{q}_i) \quad \textit{where} \quad \boldsymbol{Q}_{i,j} = \frac{e^{\boldsymbol{z}_i^T \boldsymbol{z}_j}}{\sum_{t\neq i} e^{\boldsymbol{z}_i^T \boldsymbol{z}_t}}. \tag{17}$$

With some simple transformation, the objective can be rewritten as:

$$-\langle \boldsymbol{A}, \log \boldsymbol{Q} \rangle = -\sum_{i=1}^{m}\sum_{j=1}^{m} \boldsymbol{A}_{i,j}\, \boldsymbol{z}_i^T \boldsymbol{z}_j + \sum_{i=1}^{m} \log \sum_{t\neq i} e^{\boldsymbol{z}_i^T \boldsymbol{z}_t}. \tag{18}$$

As the cosine similarity is equivalent to Euclidean distance on $\mathbb{S}^{d-1}$, the first term of (18) is the Dirichlet energy. We derive the matrix formulation of (18) as follows:

$$\min_{\boldsymbol{Z}} \; E(\boldsymbol{Z}, \boldsymbol{A}) + \left(\mathbf{1}^T \log\left[\left(\exp(\boldsymbol{Z}\boldsymbol{Z}^T) - e\boldsymbol{I}\right)\mathbf{1}\right]\right), \quad s.t. \quad \boldsymbol{Z} \subset \mathbb{S}^{d-1}, \tag{19}$$

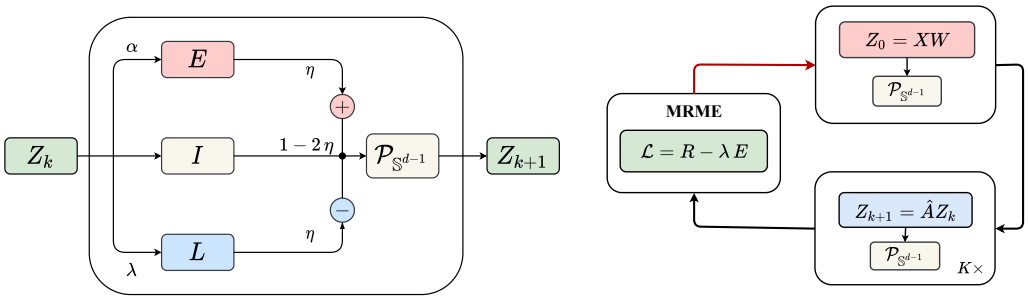
Compare (19) with the MRME objective, the only difference is in the diversity measure they use to maximize dimension. In SNE, it minimize the pairwise similarities to learn a diverse representation. However, such a measure is implicitly defined and it is unclear weather it can perform well for a higher representation space (for $d > 3$ dimensions). By contrary, the coding rate here is a information-theoretic measure designed to measure the dimension of the representation directly. It is useful to learned linear discriminant representation as in [].

Connection to Rate Reduction. The recent proposed rate reduction principle use the coding rate as the compactness measure. The basic idea is to maximize the gain between the total and the class structure. In our case, we takes coding rate as a approximate

Connection to DGI. DGI tries to maximize the mutual information between local pairs and global representation.

4 Deep Graph Networks from Optimizing MRME

There are two popular ways to optimize MRME on $\mathbb{S}^{d-1}$: In a shallow embedding scheme, we can directly optimize the representation matrix $\boldsymbol{Z}$. In a encoder-based scheme, one can design a graph neural network as a "black-box" encoder, and then optimize MRME to learn the parameter of the encoder in a backpropagation manner. In this section, we will show that these two optimization schemes can be unified together. The gradient ascent scheme will lead to forward deep graph network RENet, and we will design a linear graph network RELGN from RENet to optimize MRME in a back propagation manner.



(a) RENet (b) RELGN

Figure 2: Influence of model depth (number of layers) on classification performance.

4.1 RENET: A "WHITE-BOX" FORWARD GRAPH NEURAL NETWORK

To simplify the formulation, we here assume the adjacency matrix $\boldsymbol{A}$ is symmetric, therefore the gradient of MRME is:

$$\frac{1}{2}\frac{\partial \log\det(\boldsymbol{I}+\alpha\boldsymbol{Z}^T\boldsymbol{Z})}{\partial \boldsymbol{Z}} = \alpha\,\boldsymbol{Z}\boldsymbol{E} \quad \textit{where} \quad \boldsymbol{E}=(\boldsymbol{I}+\alpha\boldsymbol{Z}^T\boldsymbol{Z})^{-1}, \tag{20}$$

$$\frac{1}{2}\frac{\partial \lambda\, tr(\boldsymbol{Z}^T\boldsymbol{L}\boldsymbol{Z})}{\partial \boldsymbol{Z}} = \lambda\,\boldsymbol{L}\boldsymbol{Z} \quad \textit{where} \quad \boldsymbol{L}=\boldsymbol{D}-\boldsymbol{A}. \tag{21}$$

Taking the gradient to optimize MRME leads to the following update formulation:

$$\hat{\boldsymbol{Z}} \propto \boldsymbol{Z}+\eta\cdot\nabla_{\boldsymbol{Z}}\mathcal{L} = \boldsymbol{Z}+\eta\cdot(\alpha\,\boldsymbol{Z}\boldsymbol{E}-\lambda\,\boldsymbol{L}\boldsymbol{Z}), \quad \textit{subject to} \quad \boldsymbol{Z}\subset\mathbb{S}^{d-1}. \tag{22}$$

Instead of setting the step size η, we set $\eta := \eta\,/\,(1+2\eta) \in (0,1)$, therefore update formation can be rewritten as:

$$\hat{\boldsymbol{Z}} \propto (1-2\eta)\cdot\boldsymbol{Z}+\eta\cdot(\alpha\,\boldsymbol{Z}\boldsymbol{E})-\eta\cdot(\lambda\,\boldsymbol{L}\boldsymbol{Z}). \tag{23}$$

We see it naturally leads to a deep graph network. In each layer, the linear operator $\boldsymbol{E}$ expands the dimension of $\boldsymbol{Z}$ to maximize coding rate, while the Laplacian operator $\boldsymbol{L}$ smooths the representation so as to minimize Dirichlet energy. As such, we call such a forward deep graph network as RENet, meaning that it is derived from optimizing *Rate* and *Energy*. Figure 2 shows the structure of RENet. The following we will show that RENet from the spectral and residual.

RENet on the Macroscopic Scale. Assuming that $\boldsymbol{Z}^T\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{V}^T$ and $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ where $\boldsymbol{S}=\mathrm{diag}(s_1,\ldots,s_d)$ and $\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1,\ldots,\lambda_m)$ and $\boldsymbol{S}=\mathrm{diag}(s_1,\ldots,s_d)$. Therefore, for these two operator $\boldsymbol{E}$ and $\boldsymbol{L}$, we have:

Assuming that $\boldsymbol{Z}^T\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{V}^T$ where s_i is the variance of $\boldsymbol{Z}$ in direction $\boldsymbol{v}_i$, we have:

$$(\boldsymbol{Z}\boldsymbol{E}\boldsymbol{V}) = (\boldsymbol{Z}\boldsymbol{V})\mathrm{diag}\left\{\frac{1}{1+\alpha\boldsymbol{s}_1},\ldots,\frac{1}{1+\alpha\boldsymbol{s}_d}\right\}. \tag{24}$$

Therefore, $\boldsymbol{Z}\boldsymbol{E}$ shrink the direction of $\boldsymbol{Z}$ corresponding to the large variance while vanishing variance direction are kept. From a total perspective, it means that every step, the one operator shrink the while the other shrink the large variance.

Assuming that $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ where λ_i is the frequency of spectrum $\boldsymbol{u}_i$, we have:

$$(\boldsymbol{U}^T\boldsymbol{L}\boldsymbol{Z}) = \mathrm{diag}\left\{\lambda_1,\ldots,\lambda_m\right\}(\boldsymbol{U}^T\boldsymbol{Z}). \tag{25}$$

Therefore, $\boldsymbol{L}\boldsymbol{Z}$ shrinks the direction of $\boldsymbol{Z}$ corresponding to the small eigenvalues while large direction are kept, which then leads to over-smoothing problem.

RENet on the Microscopic Scale. In [xx], the matrix $\boldsymbol{E}$ on $\boldsymbol{z}_i$ means the residual $\boldsymbol{r}_i$

$$\boldsymbol{E}\boldsymbol{z}_i == \boldsymbol{z}_i - \boldsymbol{Z}\boldsymbol{q}_i \quad \textit{where} \quad \boldsymbol{q}_i = \arg\min \alpha\|\boldsymbol{z}_i-\boldsymbol{Z}\boldsymbol{q}_i\|_2^2+\|\boldsymbol{q}_i\|_2^2. \tag{26}$$

Therefore, we can approximately see $\tilde{\boldsymbol{z}}_i = \boldsymbol{Z}\boldsymbol{q}_i$ as the orthogonal projection onto the subspace spanned by $\boldsymbol{Z}$. Denote the global residual of $\boldsymbol{z}_i$ as $\tilde{\boldsymbol{r}}_i = \boldsymbol{z}_i - \tilde{\boldsymbol{z}}_i$ and define the pairwise residual as

$\boldsymbol{r}_j = \boldsymbol{z}_i - \boldsymbol{z}_j$, then for each layer, $\boldsymbol{z}_i$ is updated by:

$$\hat{\boldsymbol{z}}_i \propto (1-2\eta)\cdot \boldsymbol{z}_i + \eta \cdot \alpha\, \tilde{\boldsymbol{r}}_i - \eta \cdot \sum_{(i,j)\in\mathcal{E}} \boldsymbol{A}_{i,j}\, \lambda\, \boldsymbol{r}_j, \tag{27}$$

This clearly show why RENet can prevent oversmoothing. In each iteration, adding the $\tilde{\boldsymbol{r}}_i$ pushes $\boldsymbol{z}_i$ away from the others points, while subtracting $\boldsymbol{r}_j$ pull $\boldsymbol{z}_i$ to its neighbor $\boldsymbol{z}_j$.

4.2 RELGN: A Linear Graph Network Encoder

One can optimize MRME directly by RENet, however, for large graph structure data with high dimensional feature vector, a forward optimization may not be good choice, since the complexity of RENet is $O(k(md^2 + d^3 + msd))$. Besides, one can perform dimensionality reduction at the first, but such a initialization may be important to construct RENet. Therefore, we design a linear graph network $f(\boldsymbol{X}, \boldsymbol{A})$ based on the gradient scheme of MRME.

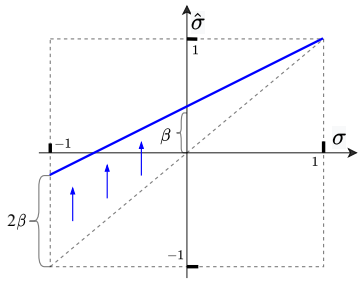


Figure 3: RELGN.

Given a feature matrix $\boldsymbol{X}$, we first reduce its dimensionality linearly, leading to $\hat{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{P}$, then we consider maximizing coding rate $R(\boldsymbol{Z}, \epsilon)$ by gradient:

$$\boldsymbol{Z}_0 \propto \hat{\boldsymbol{X}} + \eta \hat{\boldsymbol{X}}\boldsymbol{E} = \hat{\boldsymbol{X}}(\boldsymbol{I} + \eta \boldsymbol{E}). \tag{28}$$

Therefore, By denote $\boldsymbol{W} = \boldsymbol{P}(\boldsymbol{I} + \eta\boldsymbol{E}) \in \mathbb{R}^{D\times d}$, we then have $\boldsymbol{Z}^0 = \boldsymbol{X}\boldsymbol{W}$. After that, we optimize the Dirichlet energy $E(\boldsymbol{Z}, \boldsymbol{A})$ by gradient:

$$\boldsymbol{Z}^{k+1} \propto \boldsymbol{Z}^k + \eta \boldsymbol{A}\boldsymbol{Z}^k \propto \hat{\boldsymbol{A}}\boldsymbol{Z}^k \quad \textit{where} \quad \hat{\boldsymbol{A}} = \beta \boldsymbol{I} + (1-\beta)\boldsymbol{A}. \tag{29}$$

Here $\beta = \frac{1}{1+\eta} \in (0, 1)$, $\hat{A}$ is the adjacency matrix $\boldsymbol{A}$ with added self-connections. Such a renormalization has clear effect. Suppose the $\boldsymbol{A}$ is symmetric normalized, meaning that $\boldsymbol{A} := \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$, so that the eigenvalue $\sigma(\boldsymbol{A}) \in [-1, 1]$. Therefore the eigenvalue of the $\hat{\boldsymbol{A}}$ satisfies: $\hat{\sigma} = \sigma + \beta(1-\sigma) \in [2\beta - 1, 1]$. As such, $\hat{\boldsymbol{A}}\boldsymbol{Z}$ can. It should be noted that at every step, we normalize the each node representation. It we ignore this normalization, it is a Linear Graph Network (LGN). We can this network as RELGN.

$$f(\boldsymbol{X}, \mathcal{E}, \theta) = \hat{\boldsymbol{A}}^K \boldsymbol{X}\boldsymbol{W}. \tag{30}$$

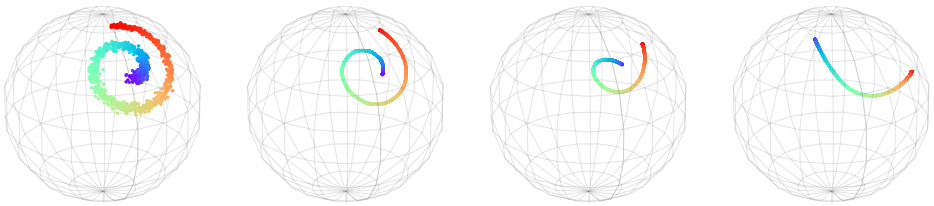
There is only a $\boldsymbol{W} \in \mathbb{R}^{D\times d}$ as parameters. It is strictly equivalent to perform K-iteration gradient to minimize Dirichlet energy, with the initial point $\boldsymbol{Z}^0 = \boldsymbol{X}\boldsymbol{W}$. Therefore, as the K increasing, the learned representation suffers from oversmoothing problem. Note that we are not to show that the RELGN encoder can prevent the oversmoothing problem, but to show that RELGN is the most simple structure motived from RENet. RELGN actually share a very similar network structure with SGC and S2GC. In experiments, we then can use backpropagation to revise the network parameter so as to optimize the MRME objective.

5 Experiments

Our theoretical analysis have shown the effectiveness of MRME to learn diverse graph representation.

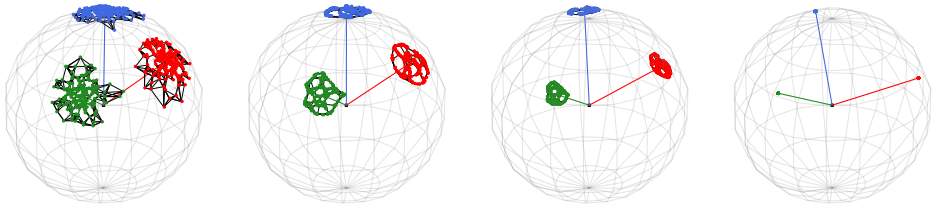
5.1 Manifold learning and Clustering Subgraphs

Unrolling Manifold. Now we show optimize MRME via RENet can unroll manifold and seperate different subgraph. To show the MRME to unroll manifold, we first generate the data points from Archimedean spiral, which have the simple equation in polar coordinates: $r = a + b\theta$. we then utilize k-neighbor to construct a graph and use the $\boldsymbol{X}$ as initial point to optimize MRME. We show how the representation change during the RENet.



(a) (b) (c) (d)

Figure 4: RENet on Archimedean Spiral.



(a) (b) (c) (d)

Figure 5: RENet on Gaussian Spiral.

Clustering Subgraphs.

6 Conclusion and Future Work

References

Hongwei Wang and Jure Leskovec. Unifying Graph Convolutional Neural Networks and Label Propagation, February 2020.

A Proof of the Main Results