From Canonical Correlation Analysis to Self-supervised Graph Neural Networks

Hengrui Zhang[1], Qitian Wu[2], Junchi Yan[2], David Wipf[3], Philip S. Yu[1]

1 University of Illinois, Chicago
2 Shanghai Jiao Tong University
3 AWS Shanghai AI Lab

Contrastive Learning

Contrastive Learning is a popular method for self-supervised representation learning.
x^A and x^B are two views of the same instance
z^A and z^B are the corresponding representations (usually normalized): $z^A = f_A(x^A), z^B = f_B(x^B)$.
One typical contrastive loss–the InfoNCE loss:

$$\mathcal{L}^A = -\sum_{i=1}^{N} \log \frac{e^{f(z_i^A, z_i^B)/\tau}}{\sum_{j=1}^{N} e^{f(z_i^A, z_j^B)/\tau}} \tag{1}$$

$$\mathcal{L}^a = -\sum_{i=1}^{N} \log \frac{e^{sim(h_i^a, h_i^b)/\tau}}{\sum_{j=1}^{N} (e^{sim(h_i^a, h_j^a)/\tau} + e^{sim(h_i^a, h_j^b)/\tau})} \tag{2}$$

$$\mathcal{L} = \mathcal{L}^a + \mathcal{L}^b \tag{3}$$

$f(\cdot, \cdot)$ is a similarity measure and could be the simple dot product.

Contrastive Learning

Theoretical foundation: maximizing a lower bound of mutual information. E.g. the InfoNCE loss is a tight lower bound of the mutual information of two views' representations:

$$I(X, Y) \geq \mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\log\frac{e^{f(x_i, y_i)}}{\sum_{j=1}^{K} e^{f(x_i, y_j)}}\right] \triangleq I_{InfoNCE}(X, Y) \tag{4}$$

Other contrastive learning loss such as InfoMax, MINE are also lower bounds of mutual information.

Self-supervised Learning for Nodes Representation Learning

- DGI and MVGRL use InfoMax as the objective function
- GRACE/GCA uses InfoNCE loss
- BGRL adopts the structure of BYOL to avoid contrasting

Despite their empirical success, they suffer from the following limitations:

- DGI/MVGRL requires parameterized MI estimator.
- GRACE/GCA has an $\mathcal{O}(N^2)$ complexity and is not scalable to large graphs.
- BGRL requires complex asymmetric structures and is not theoretically explainable.

To tackle these limitations, we propose a novel framework for self-supervised learning on graphs, which is based on canonical correlation analysis.

The simplest SSL method

Table 1: Technical comparison of self-supervised node representation learning methods. We provide a conceptual comparison with more self-supervised methods in Appendix G. *Target* denotes the comparison pair, N/G/F denotes node/graph/feature respectively. *MI-Estimator*: parameterized mutual information estimator. *Proj/Pred*: additional (MLP) projector or predictor. *Asymmetric*: asymmetric architectures such as EMA and Stop-Gradient, or two separate encoders for two branches. *Neg examples*: requiring negative examples to prevent trivial solutions. *Space* denotes space requirement for storing all the pairs. Our method is simple without any listed component and memory-efficient.

	Methods	Target	MI-Estimator	Proj/Pred	Asymmetric	Neg examples	Space
Instance-level	DGI [48]	N-G	✓	-	-	✓	$O(N)$
	MVGRL [15]	N-G	✓	-	✓	✓	$O(N)$
	GRACE [57]	N-N	-	✓	-	✓	$O(N^2)$
	GCA [58]	N-N	-	✓	-	✓	$O(N^2)$
	BGRL [39]	N-N	-	✓	✓	-	$O(N)$
	CCA-SSG (Ours)	F-F	-	-	-	-	$O(D^2)$

Canonical Correlation Analysis

Given two random variables $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^n$, whose covariance matrix is $\Sigma_{XY} = Cov(X, Y)$. CCA aims at seeking two vectors $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$ such that the correlation $\rho = \text{corr}(a^\top X, b^\top Y) = \frac{a^\top \Sigma_{XY} b}{\sqrt{a^\top \Sigma_{XX} a}\sqrt{b^\top \Sigma_{YY} b}}$ is maximized:

$$\max_{a,b} a^\top \Sigma_{XY} b, \text{ s.t. } a^\top \Sigma_{XX} a = b^\top \Sigma_{YY} b = 1. \tag{5}$$

Multi-dimensional and non-linear cases:

$$\max_{\theta_1,\theta_2} \text{Tr}\left(P_{\theta_1}^\top(X_1)P_{\theta_2}(X_2)\right) \text{ s.t. } P_{\theta_1}^\top(X_1)P_{\theta_1}(X_1) = P_{\theta_2}^\top(X_2)P_{\theta_2}(X_2) = I. \tag{6}$$

Soft decorrelation:

$$\min_{\theta_1,\theta_2} \mathcal{L}_{dist}\left(P_{\theta_1}(X_1), P_{\theta_2}(X_2)\right) + \lambda\left(\mathcal{L}_{SDL}(P_{\theta_1}(X_1)) + \mathcal{L}_{SDL}(P_{\theta_2}(X_2))\right), \tag{7}$$

Our Method

- Input: Graph $G = (X, A)$.
- Graph augmentations: **edge dropping** and **node feature masking**. Then $\tilde{G}_A = (\tilde{X}_A, \tilde{A}_A)$ and $\tilde{G}_B = (\tilde{X}_B, \tilde{A}_B)$
- Encoder: Graph Neural Network. $Z_A = f_\theta(\tilde{X}_A, \tilde{A}_A)$, $Z_B = f_\theta(\tilde{X}_B, \tilde{A}_B)$.
- Normalization along feature dimension: $\tilde{Z} = \frac{Z - \mu(Z)}{\sigma(Z) * \sqrt{N}}$

Objective Function:

$$\mathcal{L} = \underbrace{\left\| \tilde{Z}_A - \tilde{Z}_B \right\|_F^2}_{\text{invariance term}} + \lambda \underbrace{\left(\left\| \tilde{Z}_A^\top \tilde{Z}_A - \mathbf{I} \right\|_F^2 + \left\| \tilde{Z}_B^\top \tilde{Z}_B - \mathbf{I} \right\|_F^2 \right)}_{\text{decorrelation term}} \tag{8}$$

Advantages over previous methods

- No reliance on negative samples.
- No MI estimator, projector network nor asymmetric architectures.
- Better efficiency and scalability to large graphs.

Theoretical Analysis

Some notations:

1. X: the input data.
2. S: the augmented data.
3. T: downstream task.
4. $Z_X = f_\theta(X)$.
5. $Z_S = f_\theta(S)$.
6. $I(A, B)$: mutual information.
7. $I(A, B|C)$: conditional mutual information.
8. $H(A)$: entropy.
9. $H(A|B)$: conditional entropy.

Interpretation with Entropy and Mutual Information

Assumption 1: Gaussian assumption of $P(Z_S|X)$ and $P(Z_S)$:

$$P(Z_S|X) = P(Z_S|X) = \mathcal{N}(\mu_X, \Sigma_X), P(Z_S) = \mathcal{N}(\mu, \Sigma). \tag{9}$$

We have the following propositions:
Proposition 1: In expectation, minimizing $\mathcal{L}_{inv}$ is equivalent to minimizing the entropy of Z_S conditioned on input X, i.e.,

$$\min_\theta \mathcal{L}_{inv} \cong \min_\theta H(Z_S|X).$$

Proposition 2: Minimizing $\mathcal{L}_{dec}$ is equivalent to maximizing the entropy of Z_S, i.e.,

$$\min_\theta \mathcal{L}_{dec} \cong \max_\theta H(Z_S).$$

Theorem 1

Combining Proposition 1 and Proposition 2, we have the following theorem.

Theorem

By optimizing Eq (8), we maximize the mutual information between the augmented view's embedding Z_S and the input data X, and minimize the mutual information between Z_S and the view itself S, conditioned on the input data X. Formally we have

$$\min_{\theta} \mathcal{L} \Rightarrow \max_{\theta} I(Z_S, X) \text{ and } \min_{\theta} I(Z_S, S|X). \tag{10}$$

The proof is simple and based on the following two equations:
1) $I(Z_S, S|X) = H(Z_S|X)$ and 2) $I(Z_S, X) = H(Z_S) - H(Z_S|X)$.

Connection with the Information Bottleneck Principle

First, let's recall the Supervised Information Bottleneck Principle.
Definition 1. The supervised IB aims at maximizing an Information Bottleneck Lagrangian:

$$\mathcal{IB}_{sup} = I(Y, Z_X) - \beta I(X, Z_X), \text{ where } \beta > 0. \tag{11}$$

$\mathcal{IB}_{sup}$ attempts to maximize the information between the data representation Z_X and its corresponding label Y, and concurrently minimize the information between Z_X and the input data X (i.e., exploiting compression of Z_X from X). The intuition of IB principle is that Z_X is expected to contain only the information that is useful for predicting Y.

Connection with the Information Bottleneck Principle

Apply Information Bottleneck Principle to Self-supervised Learning:

Definition 2. (Self-supervised Information Bottleneck[1][2][3]). The Self-supervised IB aims at maximizing the following Lagrangian:

$$\mathcal{IB}_{ssl} = I(X, Z_S) - \beta I(S, Z_S), \text{ where } \beta > 0. \tag{12}$$

Intuitively, $\mathcal{IB}_{ssl}$ posits that a desirable representation is expected to be informative to augmentation invariant features, and to be a maximal compressed representation of the input.

[1] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. "Barlow twins: Self-supervised learning via redundancy reduction". ICML 2021.

[2] Tsai, Yao-Hung Hubert, et al. "Self-supervised learning from a multi-view perspective". ICLR 2021.

[3] Federici, Marco, et al. "Learning robust representations via multi-view information bottleneck". ICLR 2020.

Connection with the Information Bottleneck Principle

Theorem (2)

Assume $0 < \beta \leq 1$, then by minimizing the loss function $\mathcal{L}$, the self-supervised Information Bottleneck objective is maximized, formally:

$$\min_{\theta} \mathcal{L} \Rightarrow \max_{\theta} \mathcal{IB}_{ssl}$$

Connection with the Information Bottleneck Principle

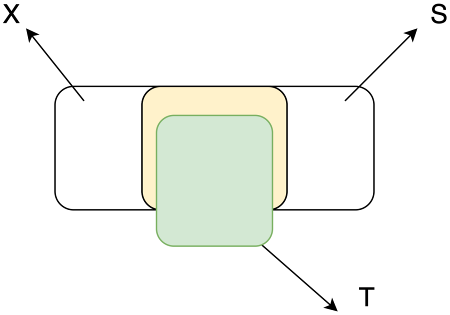
Theorem

Assume $0 < \beta \leq 1$*, then by minimizing Eq.* (8)*, the self-supervised Information Bottleneck objective is maximized, formally:*

$$\min_{\theta} \mathcal{L} \Rightarrow \max_{\theta} \mathcal{IB}_{ssl} \tag{13}$$

Influence on Downstream Tasks

Assumption 2 (Task-relevant information and data augmentation):
All the task-relevant information is shared across the input data X and its augmentations S, i.e., $I(X, T) = I(S, T) = I(X, S, T)$, or equivalently, $I(X, T|S) = I(S, T|X) = 0$.



Influence on Downstream Tasks

Theorem (Task-relevant/irrelevant information)

By optimizing Eq. (8)*, the task-relevant information* $I(Z_S, T)$ *is maximized, and the task-irrelevant information* $H(Z_S|T)$ *is minimized. Formally,*

$$\min_{\theta} \mathcal{L} \Rightarrow \max_{\theta} I(Z_S, T) \textit{ and } \min_{\theta} H(Z_S|T). \tag{14}$$

Major experimental results

Table 2: Test accuracy on citation networks. The *input* column highlights the data used for training. (**X** for node features, **A** for adjacency matrix, **S** for diffusion matrix, and **Y** for node labels).

	Methods	Input	Cora	Citeseer	Pubmed
Supervised	MLP [47]	**X**, **Y**	55.1	46.5	71.4
	LP [56]	**A**, **Y**	68.0	45.3	63.0
	GCN [22]	**X**, **A**, **Y**	81.5	70.3	79.0
	GAT [47]	**X**, **A**, **Y**	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3
Unsupvised	Raw Features [48]	**X**	47.9 ± 0.4	49.3 ± 0.2	69.1 ± 0.3
	DeepWalk [32]	**A**	70.7 ± 0.6	51.4 ± 0.5	74.3 ± 0.9
	GAE [21]	**X**, **A**	71.5 ± 0.4	65.8 ± 0.4	72.1 ± 0.5
	DGI [48]	**X**, **A**	82.3 ± 0.6	71.8 ± 0.7	76.8 ± 0.6
	MVGRL[1] [15]	**X**, **S**, **A**	83.5 ± 0.4	$\mathbf{73.3 \pm 0.5}$	80.1 ± 0.7
	GRACE[2] [57]	**X**, **A**	81.9 ± 0.4	71.2 ± 0.5	80.6 ± 0.4
	CCA-SSG (Ours)	**X**, **A**	$\mathbf{84.2 \pm 0.4}$	73.1 ± 0.3	$\mathbf{81.6 \pm 0.4}$

Major experimental results

Table 3: Test accuracy on co-author and co-purchase networks. We report both mean accuracy and standard deviation. Results of baseline models are from [58].

	Methods	Input	Computer	Photo	CS	Physics
	Supervised GCN [22]	**X**, **A**, **Y**	86.51 ± 0.54	92.42 ± 0.22	93.03 ± 0.31	95.65 ± 0.16
	Supervised GAT [47]	**X**, **A**, **Y**	86.93 ± 0.29	92.56 ± 0.35	92.31 ± 0.24	95.47 ± 0.15
Unsupervised	Raw Features [48]	**X**	73.81 ± 0.00	78.53 ± 0.00	90.37 ± 0.00	93.58 ± 0.00
	DeepWalk [32]	**A**	85.68 ± 0.06	89.44 ± 0.11	84.61 ± 0.22	91.77 ± 0.15
	DeepWalk + features	**X**, **A**	86.28 ± 0.07	90.05 ± 0.08	87.70 ± 0.04	94.90 ± 0.09
	GAE [21]	**X**, **A**	85.27 ± 0.19	91.62 ± 0.13	90.01 ± 0.71	94.92 ± 0.07
	DGI [48]	**X**, **A**	83.95 ± 0.47	91.61 ± 0.22	92.15 ± 0.63	94.51 ± 0.52
	MVGRL [15]	**X**, **S**, **A**	87.52 ± 0.11	91.74 ± 0.07	92.11 ± 0.12	95.33 ± 0.03
	GRACE[1] [57]	**X**, **A**	86.25 ± 0.25	92.15 ± 0.24	92.93 ± 0.01	95.26 ± 0.02
	GCA[1] [58]	**X**, **A**	87.85 ± 0.31	92.49 ± 0.09	93.10 ± 0.01	$\mathbf{95.68 \pm 0.05}$
	CCA-SSG (Ours)	**X**, **A**	$\mathbf{88.74 \pm 0.28}$	$\mathbf{93.14 \pm 0.14}$	$\mathbf{93.31 \pm 0.22}$	95.38 ± 0.06

The End