Supplementary material for "Symmetric Graph Convolutional Autoencoder for Unsupervised Graph Representation Learning"

Jiwoong Park[1] Minsik Lee[2] Hyung Jin Chang[3] Kyuewang Lee[1] Jin Young Choi[1]

[1]ASRI, Dept. of ECE., Seoul National University [2]Div. of EE., Hanyang University

[3]School of Computer Science, University of Birmingham

{ptywoong,kyuewang,jychoi}@snu.ac.kr, mleepaper@hanyang.ac.kr, h.j.chang@bham.ac.uk

In this supplemental material, we present the derivation of the computationally-efficient subspace clustering cost function, the details of every dataset used in the experiments and the implementation details of the experiments.

1. Derivation of computationally-efficient subspace clustering cost function

By substituting A_H^* derived from Eq. (20) to Eq. (19) in the paper, the cost function becomes

$$\min_{\bar{X},H} \quad \frac{1}{2}\|X-\bar{X}\|_F^2+\frac{\lambda}{2}\left\|H-H(H^TH+\frac{\mu}{\lambda}I_n)^{-1}H^TH\right\|_F^2+\frac{\mu}{2}\left\|(H^TH+\frac{\mu}{\lambda}I_n)^{-1}H^TH\right\|_F^2. \tag{A.1}$$

However, minimizing the above problem is computationally expensive since it requires the inverse of n by n matrix and lots of matrix multiplications, and we need a computationally-efficient form.

The singular value decomposition (SVD) of H can simplify the tangled cost function. Let $svd(H)=U\Sigma V^T$, where $U\in\mathbb{R}^{k\times k}$ ($UU^T=U^TU=I_k$) is a unitary matrix composed of the left singular vectors of H, $V\in\mathbb{R}^{n\times n}$ ($VV^T=V^TV=I_n$) is those of the right singular vectors of H, and $\Sigma\in\mathbb{R}^{k\times n}$ is a diagonal matrix whose diagonal elements are non-zero singular values $\{\sigma_i\}_{i=1}^k$ of H. Then the second term of Eq. (A.1), $\|H-H(H^TH+\frac{\mu}{\lambda}I_n)^{-1}H^TH\|_F^2$ can be simplified as $\|\mu(\mu I_k+\lambda HH^T)^{-1}H\|_F^2$ from that

$$\begin{aligned}H-H(H^TH+\frac{\mu}{\lambda}I_n)^{-1}H^TH &= U(\Sigma-\Sigma(\Sigma^T\Sigma+\frac{\mu}{\lambda}I_n)^{-1}\Sigma^T\Sigma)V^T\\ &= U\mu(\mu I_k+\lambda\Sigma\Sigma^T)^{-1}\Sigma V^T\\ &= \mu(\mu I_k+\lambda HH^T)^{-1}H,\end{aligned} \tag{A.2}$$

where the diagonal elements of the diagonal matrices at right hand sides of the first and the second equal signs are identical to $\{\mu\sigma_i/(\mu+\lambda\sigma_i^2)\}_{i=1}^k$.

Also, the third term of Eq. (A.1), $\|(H^TH+\frac{\mu}{\lambda}I_n)^{-1}H^TH\|_F^2$ can be simplified as $\|\lambda H^T(\mu I_k+\lambda HH^T)^{-1}H\|_F^2$ from that

$$\begin{aligned}(H^TH+\frac{\mu}{\lambda}I_n)^{-1}H^TH &= V(\Sigma^T\Sigma+\frac{\mu}{\lambda}I_n)^{-1}\Sigma^T\Sigma V^T\\ &= V\Sigma^T\lambda(\mu I_k+\lambda\Sigma\Sigma^T)^{-1}\Sigma V^T\\ &= \lambda H^T(\mu I_k+\lambda HH^T)^{-1}H,\end{aligned} \tag{A.3}$$

where the diagonal elements of the diagonal matrices at right hand side of the first and the second equal sign are identical to $\{\lambda\sigma_i^2/(\mu+\lambda\sigma_i^2)\}_{i=1}^k$ and 0 for from the $(k+1)$-th to n-th diagonal elements. From the above calculation, Eq. (A.1) is transformed into

$$\min_{\bar{X},H} \quad \frac{1}{2}\|X-\bar{X}\|_F^2+\frac{\lambda}{2}\|\mu(\mu I_k+\lambda HH^T)^{-1}H\|_F^2+\frac{\mu}{2}\|\lambda H^T(\mu I_k+\lambda HH^T)^{-1}H\|_F^2. \tag{A.4}$$

We can attain a more simplified version of Eq. (A.4) by merging the second term and the third term by using SVD and the final computationally-efficient subspace clustering cost function is as follows:

$$\min_{\bar{X},H} \quad \frac{1}{2}\|X-\bar{X}\|_F^2+\frac{\mu\lambda}{2}tr((\mu I_k+\lambda HH^T)^{-1}HH^T). \tag{A.5}$$

Furthermore, the affinity matrix is given by $A_H^* = (H^{*T}H^* + \frac{\mu}{\lambda}I_n)^{-1}H^{*T}H^*$ as derived in the Section 3.3 in the paper, which requires an $n \times n$ matrix inversion. Its computationally-efficient version using a $k \times k$ matrix inversion is given by $A_H^* = \lambda H^{*T}(\mu I_k + \lambda H^* H^{*T})^{-1}H^*$.

2. Details of datasets

We have used four network datasets (Cora, Citeseer, Wiki, and Pubmed) and three image datasets (COIL20, YALE, and MNIST) for clustering tasks. Every network dataset has a feature matrix X and an affinity matrix A, and every image dataset has a feature matrix X only. The details of each dataset are as follows:

Cora [6] is a citation network between scientific publications which consists of 2,708 nodes where their feature dimension is 1,433 and there are 5,429 edges between nodes. The number of the clusters is 7.

Citeseer [6] is also a citation network between scientific publications, which has 3,312 nodes with 3,703 feature dimensions, and 4,732 edges between nodes. The number of the clusters of Citeseer is 6.

Wiki [8] is a network dataset whose number of nodes is 2,405, the dimension of the features is 4,973, and there are 17,981 edges between nodes. The number of the clusters of Wiki is 17. We observed that several papers [7], [8] have written the number of the clusters of Wiki as 19. Although nodes are labeled up to 19, the actual cardinality of the labels is 17.

Pubmed [6] is a citation network which consists of 19,717 nodes where their feature dimension is 500 and there are 44,338 edges between nodes. The number of the clusters is 3.

COIL20 [4] is the Columbia Object Image Library dataset. There are 1,440 images for 20 objects and each object contains 72 images since the pose changes every 5 degrees $(360° = 72 \times 5°)$. The size of each image is 32 $\times$ 32, so the feature dimension is 1024. The number of the clusters of COIL20 is 20.

YALE [2] is a face dataset which contains 5,850 face images with 9 poses and 65 illumination variations. Original images were cropped to 30 $\times$ 40 pixels, so the feature dimension of each face image is 1200. The number of the clusters is 10.

MNIST [3] is a handwritten digit dataset. There are 60,000 training samples and 10,000 test samples and among them, we use 10,000 test samples for image clustering. The size of each digit image is 28 $\times$ 28, so the feature dimension is 784. The number of the clusters of MNIST is 10.

3. Implementation details

We conduct our algorithm on a GPU environment (a Nvidia GTX 1080Ti GPU) in TensorFlow [1]. For the Cora dataset, we set the encoder's number of layers to two (1600, 400 neurons). For the Citeseer dataset, we set the number of layers of the encoder to two (2000, 500 neurons). For the Wiki dataset, we set the encoder's number of layers to one (500 neurons). For the Pubmed dataset, we set the encoder's number of layers to two (600, 100 neurons). For the COIL20 dataset, we set $\lambda = 9.0$, $\mu = 1.0$, and the number of layers of the encoder to three (1100, 800, 500 neurons). For the YALE dataset, we set $\lambda = 3.0 \times 10$, $\mu = 1.0$, and the encoder's number of layers to three (1300, 800, 500 neurons). For the MNIST dataset, we set $\lambda = 5.0 \times 10$, $\mu = 1.0$, and the number of layers of the encoder to three (800, 700, 500 neurons). The encoder and decoder have symmetrical structures for all datasets. We set the parameters of the compared methods following the instructions of their papers. Among the parameter sets noted in each paper, we reported the best results [7, 5]. For the image datasets, we construct k-nearest neighborhood graphs using each dataset's feature matrix. We set k as 9, 9, and 20 for COIL20, YALE, and MNIST respectively. Also, we normalize the feature of each image on unit interval for all image datasets.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 2

[2] Athinodoros S Georghiades, Peter N Belhumeur, and David J Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):643–660, 2001. 2

[3] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 2

[4] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). 1996. 2

[5] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2609–2615. International Joint Conferences on Artificial Intelligence Organization, 2018. 2

[6] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, 2008. 2

[7] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. Mgae: Marginalized graph autoencoder for graph clustering. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 889–898. ACM, 2017. 2

[8] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *IJCAI*, pages 2111–2117, 2015. 2