An Inductive Framework for Semi-supervised Learning

(Discussion Paper)

Shuyi Yang[1,2], Dino Ienco[3], Roberto Esposito[1] and Ruggero G. Pensa[1]

[1] University of Turin, Italy
[2] Intesa Sanpaolo, Turin, Italy
[3] INRAE, UMR TETIS, Montpellier, France

Abstract

Semi-supervised learning is crucial in many applications where accessing class labels is unaffordable or costly. The most promising approaches are graph-based but they are transductive and they do not provide a generalized model working on inductive scenarios. To address this problem, we propose a generic framework for inductive semi-supervised learning based on three components: an ensemble of semi-supervised autoencoders providing a new data representation that leverages the knowledge supplied by the reduced amount of available labels; a graph-based step that helps augmenting the training set with pseudo-labeled instances and, finally, a classifier trained with labeled and pseudo-labeled instances. The experimental results show that our framework outperforms state-of-the-art inductive semi-supervised methods.

Keywords

semi-supervised learning, graph-based algorithms sep inductive methods

1. Introduction

Prediction is one of the most important outcomes of any machine learning algorithm, but its accuracy strongly depends on the amounts and quality of labeled instances and, unfortunately, labeling is a cost-intensive manual activity requiring time, money, and expertise. Hence, labeling often turns out to be unaffordable for many organizations and, consequently, only small amounts of labeled instances are available for training. Semi-supervised learning aims at mitigating the above-mentioned problem by leveraging the so-called *smoothness* and *cluster* assumptions: if two data instances are close to each other or belong to the same cluster in the input distribution, then they are likely to belong to the same class [1]. Graph-based models constitute one of the main families of semi-supervised techniques [2]. They leverage the manifold assumption: the graphs, typically nearest neighbor graphs built upon the local similarity between data points, provide a lower-dimensional representation of the high-dimensional input data.

SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy

shuyi.yang@unito.it (S. Yang); dino.ienco@inrae.fr (D. Ienco); roberto.esposito@unito.it (R. Esposito); ruggero.pensa@unito.it (R. G. Pensa)
0000-0002-8736-3132 (D. Ienco); 0000-0001-5366-292X (R. Esposito); 0000-0001-5145-3438 (R. G. Pensa)

CEUR Workshop Proceedings (CEUR-WS.org)

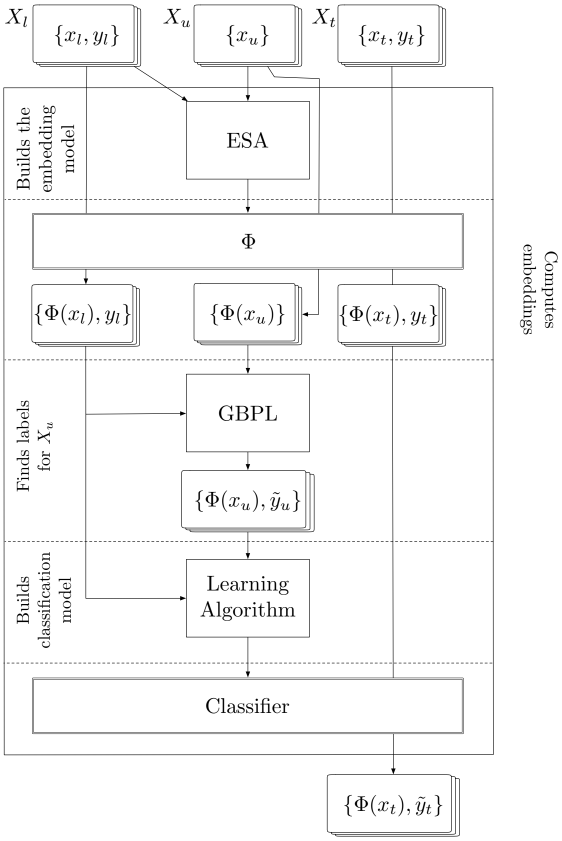


Figure 1: The proposed framework.

Unfortunately, graph-based methods are transductive [2], i.e., they do not construct any classification model and the prediction is limited to exactly those data instances that are already available during the training phase. Therefore, graph-based methods are unable to classify new data examples, unless they are trained again on the augmented dataset. A second limitation concerns the construction of the graph: in general, this phase is completely unsupervised even though, for some instances, labels are available. When the cluster assumption is not completely satisfied, this could lead to poor prediction results. In this paper, we present a novel graph-based semi-supervised framework, ESA*, that improves in the areas mentioned above: it takes into account the information carried out by labeled instances during the graph construction and is designed to work properly in inductive settings.

Our approach (sketched graphically in Fig. 1) first constructs a new representation using a semi-supervised autoencoder that takes all labeled and unlabeled training data as input. The representation learnt by the semi-supervised autoencoder, for both labeled and unlabeled training data, are then processed by a graph-based semi-supervised algorithm that propagates the label information from labeled to unlabeled data instances. This procedure provides pseudo-labels for the set of unlabeled instances. We propose two variants of our framework: the first, ESA$^{\text{LP}}$, is based on a graph-based label propagation algorithm [3]; the second one, ESA$^{\text{GAT}}$, exploits a graph convolutional neural network with masked self-attention layers [4]. All training instances (labeled and unlabeled with pseudo-labels) are then used to train a classification model, which can perform prediction for new unseen examples as well. We show that our approach outperforms state-of-the-art approaches and is competitive towards state-of-the-art transductive methods, even with extremely small amounts of labeled instances. A more in-depth presentation of our proposal is reported in the full version of this paper [5].

2. Inductive graph-based semi-supervised learning

In a semi-supervised learning setting, in addition to labeled instances, unlabeled ones are introduced as part of available data during the training phase: let $X_l \in \mathbb{R}^{n_l \times f}$ be the matrix of n_l labeled samples each with f predictors and y_l be the corresponding labels, then a supplementary matrix $X_u \in \mathbb{R}^{n_u \times f}$ representing n_u unlabeled instances is also provided without the corresponding y_u labels. Generally, the number n_l of labeled instances is limited and much smaller than the number n_u of unlabeled instances. Our framework aims to provide an inductive semi-supervised learning algorithm by leveraging graph-based semi-supervised learning in order to augment the amount of labeled instances to train a supervised classifier.

As shown in Figure 1, our framework consists of different parts: embedding computation, pseudo-labeling of unlabeled instances, and classification. In the embedding computation part, we train an ensemble of neural networks to extract a latent representation for each instance. These representations are used to build a graph over labeled and unlabeled instances so that a graph-based model can be employed to provide a pseudo-label for each unlabeled instance. Finally, labeled instances and pseudo-labeled ones are both used to train a supervised classification model.

In order to extract the data embeddings, an Ensemble of Semi-supervised Autoencoders (ESA) [6] is trained on both labeled and unlabeled data. The loss function we use to learn the internal parameters of the SSAE is a combination of reconstruction and classification loss. More formally

$$L_{\text{SSAE}} = L_{\text{AE}} + \lambda L_{\text{CL}} \tag{1}$$

where

$$L_{\text{AE}} = \frac{1}{n_l + n_u} \sum_{x_i \in X_l \cup X_u} ||x_i - D(E(x_i|\theta_E)|\theta_D)||^2, \tag{2}$$

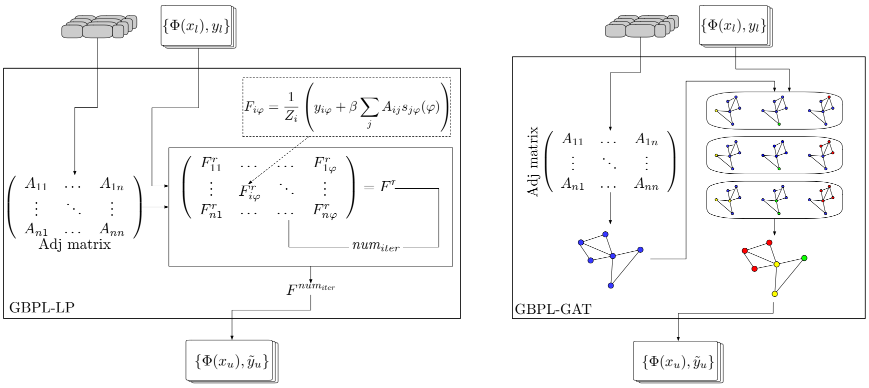
$$L_{\text{CL}} = -\frac{1}{n_l} \sum_{x_i \in X_l} \sum_{c=1}^{|C|} y_{lic} \cdot \log(CL(E(x_i|\theta_E)|\theta_{CL})_c), \tag{3}$$

and θ_E, θ_D and θ_{CL} are respectively the set of parameters of the encoder, decoder and classification layer, y_{lic} is the c-th element of the i-th row of y_l, $CL(\cdot)_c$ is the c-th element of the output vector of *CL* and λ is a parameter that controls the importance of the classification loss. In our architecture the encoder has an input layer followed by other two hidden layers; the decoder has one hidden layer of the same size of the first hidden layer of the encoder and an output layer. The size of the input layer, the output layer and the classification layer are respectively fixed to f, f and $|C|$, while $size_{hidden}$ and $size_{bottleneck}$ (respectively, the size of the hidden layer and that of the bottleneck one) can be varied. In order to get diverse and multi-resolution representations, similarly as in [6], we train K independent SSAEs, each with the sizes of the layers extracted randomly from the intervals $\frac{f}{2} \leq size_{hidden} < f$ and $\frac{f}{4} \leq size_{hidden} < \frac{f}{2}$. Once the ensemble is trained we obtain the new representations $\Phi(X_l)$, $\Phi(X_u)$ of X_l and X_u by concatenating the embeddings of these K SSAEs:

$$\Phi(\cdot) = ||_{k=1}^{k=K} E_k(\cdot|\theta_{E_k}) \tag{4}$$

where $||$ is the stack (concatenation) operator and E_k and θ_{E_k} are respectively the encoder of the k-th SSAE and its weights.

Given the latent representations, a kNN graph structure can be derived from the data points of $X_l \cup X_u$: embedding representations are nodes and two of them can be considered connected if both of them belong to the top k nearest neighbors of each other, respectively.


$\{\Phi(x_l), y_l\}$
$F_{i\varphi} = \frac{1}{Z_i}\left(y_{i\varphi} + \beta \sum_j A_{ij} s_{j\varphi}(\varphi)\right)$
F^r_{11} … … $F^r_{1\varphi}$
$F^r_{i\varphi}$
F^r_{n1} … … $F^r_{n\varphi}$
$= F^r$
num_{iter}
A_{11} … A_{1n}
A_{n1} … A_{nn}
Adj matrix
$F^{num_{iter}}$
GBPL-LP
$\{\Phi(x_u), \tilde{y}_u\}$
$\{\Phi(x_l), y_l\}$
Adj matrix
A_{11} … A_{1n}
A_{n1} … A_{nn}
GBPL-GAT
$\{\Phi(x_u), \tilde{y}_u\}$

(a) ESA$^{\text{LP}}$ (b) ESA$^{\text{GAT}}$

Figure 2: A graphical representation of the label propagation strategy of ESA$^{\text{LP}}$ (a) and ESA$^{\text{GAT}}$ (b)

At this point we perform graph-based pseudo-labeling (GBPL) to assign pseudo class labels to unlabeled instances. To this purpose, any graph-based semi-supervised learning algorithm (GBSSL) can be applied to infer the labels of the unlabeled portion of data $\tilde{y}_u$ by propagating the class information from the labeled data y_l over the graph constructed on the embeddings. Successively, a supervised classifier (SC) can be trained leveraging the union of the labeled data $(\Phi(X_l), y_l)$ with the pseudo-labeled one $(\Phi(X_u), \tilde{y}_u)$ as training set. In prediction, we first compute the latent representation of unseen data $\Phi(X_t)$ with the trained ESA, then we make predictions with the supervised classifier SC. It is worth pointing out that during the entire process, the transductive GBSSL process is used only during the training phase to provide pseudo-labels of the unlabeled data (as in wrapper methods) in order to help the supervised classifier to generalize better. Therefore, our approach, hereinafter referred as ESA*, is inductive. In the next two sections, we present two variants adopting different strategies to perform pseudo-labeling based on different graph-based semi-supervised learning approaches.

2.1. Pseudo-labeling based on confidence-aware label propagation

In this section we introduce the first variant of our framework for semi-supervised learning (see Figure 2(a)). The adopted strategy consists in instantiating the graph-based pseudo-labeling (GBPL) part with a confidence-aware label propagation algorithm working on both homophily and heterophily networks [3]. In the following we provide the details of this strategy, which we name ESA$^{\text{LP}}$. Given the adjacency matrix A, ESA$^{\text{LP}}$ computes the probability distribution over the classes as the solution of $F_{i\varphi} = \frac{1}{Z_i}\left(y_{i\varphi} + \beta \sum_j A_{ij} s_{ji}(\varphi)\right)$, where $F_{i\varphi}$ is the probability that i-th instance has label φ, Z_i is a normalization term, $y_{i\varphi}$ is the prior belief of i-th instance having label φ, $0 < \beta$ represents the importance of the neighborhood's influence, A_{ij} is the i, j entry of the adjacency matrix and $s_{ji}(\varphi)$ represents how intense the node j believes that the node i has class φ. More formally, $s_{ji}(\varphi) = \sum_l F_{jl} H_{l\varphi}$, where H is the modulation matrix. If $H_{l\varphi}$ is low then class l has a low correlation with the class φ, on the contrary, if it is high these two classes have a strong correlation. On homophily networks, H is the identity matrix, while on heterophily networks it can be designed empirically. In our experiments we assume that the graph obtained by the embeddings of ESA is a homophily network. We can rewrite the the last equation in matrix form and in an iterative way, i.e., $F^{r+1} = Z^{-1}(Y + \beta A F^r H)$, where $Z = I + \beta D$ and D is the node degrees diagonal matrix. Once obtained the adjacency matrix A of labeled and unlabeled instances, as described in the previous section, we initialize F^0 as a $(n_l + n_u) \times |C|$ matrix of zeros. Then we apply the iterative formula num_{iter} times to obtain $F^{num_{iter}}$, which represents the probability distributions of the instances over the classes. From $F^{num_{iter}}$ we extract only the predictions of X_u and keep the original labels for X_l. They are then used to feed a classifier SC.

2.2. Pseudo-labeling based on graph attention networks

For the second variant, we consider a completely different approach leveraging the convolution operation with graph attention networks (GAT) [4], which are able to capture different levels of importance of features of neighborhood nodes in the kNN graph built upon the embeddings computed by ESA. We call this strategy ESA^{GAT} and provide the details below (a graphical representation is given in Figure 2(b)).

Given the set of nodes, each represented by a b-dimensional real numbers array obtained from the ESA embedding process or as a result of a previous convolutional layer, we can compute the self-attention on nodes as $e_{ij} = a(Wh_i, Wh_j|\aleph)$, where $h_i, h_j \in \mathbb{R}^b$ are the embeddings of instance i and j, W is a $b' \times b$ shared linear transformation matrix, and $a : \mathbb{R}^{b'} \times \mathbb{R}^{b'} \to \mathbb{R}$ is the attentional mechanism consisting in a feedforward layer with weights $\aleph$ and LeakyReLU activation [7]. Each e_{ij} is computed only for connected nodes (masked attention) so that the graph structure is embedded into the coefficients. The attention coefficients are then normalized using the softmax function, i.e., $\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{\iota \in N(i)} \exp e_{i\iota}}$, with $N(i)$ representing the nodes connected to the node i. The new representation of the node i through the attention layer is then computed as $h_i^{'} = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} W h_j\right)$, with σ a non linear transformation. As in ESA, we can concatenate the outputs of different independent attention layers in order to employ a multi-head attention mechanism. To do that, once the ESA embeddings are obtained, we can apply the last formula multiple times: from the point of view of the neural network structure, the mechanism is realized by adding additional convolutional layers, each with its own weights to be trained and the number of nodes of the last layer should be equal to the number of classes.

3. Experiments

We assess the behavior of our framework under different settings, using Random Forests (RF) as final classifier. Thirteen publicly available real-world classification datasets, encompassing a wide variety of application scenarios, have been considered in our experiments. They exhibits different sizes (from 178 to 70 000 instances) and dimensionality (from 12 to 1087 features). Nine datasets (ANTIVIRUS, LANDSAT, MADELON, MALWARE, PARKINSON, SONAR, SPAMBASE, WAVEFORM, WINE) are from the UCI Machine Learning Repository[1], four are well-known

[1] http://archive.ics.uci.edu/ml

image datasets (USPS [8], MNIST[2], FMNIST[3], COIL20[4]). Each dataset is randomly split into three parts: labeled instances (p% of the dataset), unlabeled instances ($(70 - p)$% of the dataset), and test instances (30% of the dataset). The random split is stratified so that each dataset maintains the same proportion of labels as in the original datasets. Every supervised model is trained on labeled instances only and evaluated on test instances, while all semi-supervised models are trained both on labeled instances and unlabeled ones and evaluated on the test instances. During the experiments we vary the percentage p of labeled instances to study how the performances change when the portion of labeled instances increases in both supervised and semi-supervised models.

To obtain more robust performance indicators, for each combination of dataset, percentage p and model, we evaluate 25 different random splits as described above and then take the average performances. The model is re-trained for each of the 25 different splits and new predictions are made on every different test set. As performance index, we consider the micro-averaged F1-score computed on the test set. In all evaluations of the experiments, we compute detailed performance results for each dataset, but here we report a summary of the results (Figure 3). The latter is obtained as follows: for a given percentage p of labeled instances, we compute the average ranks across all the dataset for each algorithm, according to the micro-averaged F1 score, and then we plot them for increasing values of $p\%$. This allows us to obtain an overall picture of the relative performances of all competitors considered in our study. In the following, we present and discuss the results for each experiment.

3.1. Comparative analysis w.r.t. inductive methods

In the first experiment, we compare two configurations of our framework to four well-known supervised methods (Random Forests, Multilayer Perceptron, SVM and DRM [9]) and two recent state-of-the-art semi-supervised approaches: interpolation consistency training (ICT) [10] and ladder networks (LN) [11]. From the obtained results (Figure 3(a)), it emerges that, for every percentage of labeled examples, on average, the two variants of our framework outperform all other methods, including the four fully supervised classifiers considered in this study. It is worth noting that, in contrast, the two competing semi-supervised methods (*ICT* and *LN*) are not able to outperform the supervised competitors with the same consistency. The micro-averaged F1 score of *ICT* is below the one of RF, for any given value of p. Ladder networks (*LN*) are ranked third with less than 2% of labeled samples, but RF is still competitive w.r.t. *LN* despite the fact that it does not take advantage of unlabeled instances. Finally, it is worth pointing out that, not surprisingly, when the number of labeled instances increases, the differences between semi-supervised methods and fully supervised ones decrease.

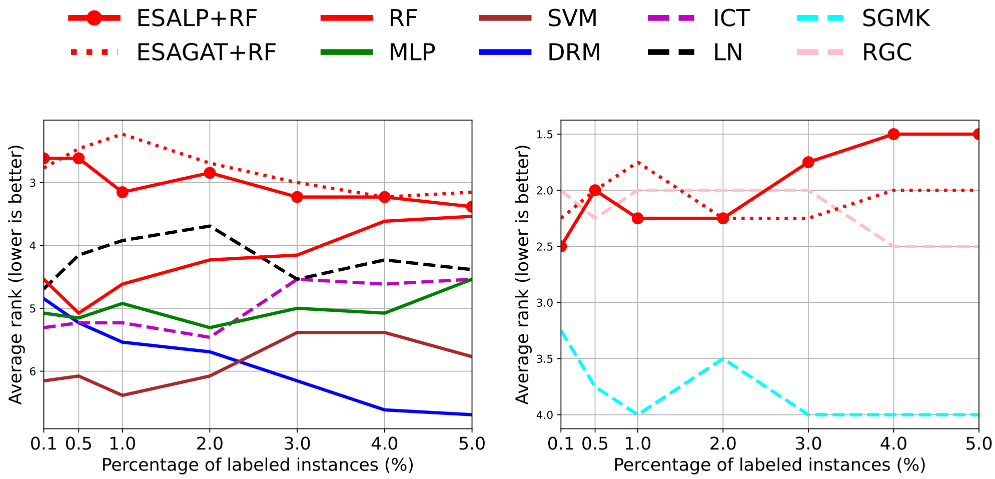
3.2. Comparative analysis w.r.t. transductive methods

We compare two configurations of our framework to two state-of-the-art transductive semi-supervised approaches: structured graph learning with multiple kernel (SGMK) [12] and robust

[2] http://yann.lecun.com/exdb/mnist/

[3] https://github.com/zalandoresearch/fashion-mnist

[4] https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php



(a) ESA* vs inductive methods

(b) ESA* vs transductive methods

Figure 3: Average rank according to micro-averaged F1-measure in the experiments.

graph construction (RGC) [13]. It is worth pointing out that, while the transductive setting has potential liabilities in terms of applicability, it has advantages in the possibility of leveraging more information than its inductive counterparts (since it can leverage the test set distribution when propagating the labels). As we shall see, while our method is at a disadvantage here, it works quite well nonetheless.

We compare the variants ESA$^{\mathrm{LP}}$+RF and ESA$^{\mathrm{GAT}}$+RF to the structured graph learning with multiple kernel (SGMK) [12] and to the robust graph construction (RGC) [13]. Due to computational limitations, the experiments are performed only on a subset of the available datasets, namely: ANTIVIRUS, SONAR, PARKINSON and WINE. From the results shown in Figure 3(b), we can conclude that, within our experimental settings, ESA$^{\mathrm{LP}}$+RF and ESA$^{\mathrm{GAT}}$+RF are able to reach competitive performances compared to SOTA transductive methods (i.e. RGC), and, in some cases, even outperform them (i.e. SGMK).

4. Conclusion

In this paper, we have presented a new inductive semi-supervised learning framework that takes the most of two successful approaches: semi-supervised autoencoders and graph-based semi-supervised learning. While the former supports the generation of new data representations improved by labeled instances, the latter spread the label information to unlabeled instances in the new representation space. Thanks to an extensive experimental study, we have shown that a classifier trained with both labeled instances and pseudo-labeled instances achieves better prediction accuracy than its supervised counterpart trained only on labeled ones, and also outperforms state-of-the-art semi-supervised competitors.

References

[1] O. Chapelle, B. Schölkopf, A. Zien (Eds.), Semi-Supervised Learning, The MIT Press, 2006.

[2] J. E. van Engelen, H. H. Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2020) 373–440.

[3] Y. Yamaguchi, C. Faloutsos, H. Kitagawa, CAMLP: confidence-aware modulated label propagation, in: S. C. Venkatasubramanian, W. M. Jr. (Eds.), Proceedings of the International Conference on Data Mining, SIAM 2016, Miami, Florida, USA, May 5-7, 2016, SIAM, 2016, pp. 513–521.

[4] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, OpenReview.net, 2018.

[5] S. Yang, D. Ienco, R. Esposito, R. G. Pensa, ESA*: A generic framework for semi-supervised inductive learning, Neurocomputing 447 (2021) 102–117.

[6] D. Ienco, R. G. Pensa, Enhancing graph-based semisupervised learning via knowledge-aware data embedding, IEEE Trans. Neural Networks Learn. Syst. 31 (2020) 5014–5020.

[7] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proceedings of the International Conference on Machine Learning, ICML 2013, Atlanta, USA, June 16-21, 2013, 2013.

[8] J. J. Hull, A database for handwritten text recognition research, IEEE Transactions on pattern analysis and machine intelligence 16 (1994) 550–554.

[9] C. Peng, Q. Cheng, Discriminative ridge machine: A classifier for high-dimensional data or imbalanced data, IEEE Transactions on Neural Networks and Learning Systems (2020). Available online.

[10] V. Verma, A. Lamb, J. Kannala, Y. Bengio, D. Lopez-Paz, Interpolation consistency training for semi-supervised learning, in: S. Kraus (Ed.), Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 3635–3641.

[11] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, T. Raiko, Semi-supervised learning with ladder networks, in: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (Eds.), Proceedings of the Annual Conference on Neural Information Processing Systems 2015, NIPS 2015, Montreal, Quebec, Canada, December 7-12, 2015, 2015, pp. 3546–3554.

[12] Z. Kang, C. Peng, Q. Cheng, X. Liu, X. Peng, Z. Xu, L. Tian, Structured graph learning for clustering and semi-supervised classification, Pattern Recognition 110 (2021) 107627.

[13] Z. Kang, H. Pan, S. C. Hoi, Z. Xu, Robust graph learning from noisy data, IEEE transactions on cybernetics 50 (2019) 1833–1843.