# Probabilistic End-to-End Graph-based Semi-Supervised Learning

Mariana Vargas Vieyra, Aurélien Bellet, Pascal Denis

# Probabilistic End-to-End Graph-based Semi-Supervised Learning

**Mariana Vargas Vieyra, Aurélien Bellet, Pascal Denis**
Inria Lille Nord Europe, France
first.last@inria.fr

## Abstract

In this paper we address the problem of graph-based semi-supervised learning in tasks where a graph describing the relationships between data points is not available. We propose a method to jointly learn the graph and the parameters of a semi-supervised model using a probabilistic framework. We empirically show our proposal achieves competitive results in a variety of datasets.

## 1 Introduction

Graph-based semi-supervised models are good at leveraging unannotated data when small amounts of labels are available: they take as input a graph whose nodes are data points (both labeled and unlabeled) and edges describe how points are related to each other. There exists a variety of such models which propagate labels based on a smoothness criterion [1, 2, 3, 4, 5]. A more recent approach uses Graph Convolutional Networks (GCN) to learn node representations based on all the input data while backpropagating the error on the labeled data [6]. Unfortunately, in many applications the graph structure is not readily available.

A standard solution is to compute graphs using classical heuristics such as $k$-nn or $\epsilon$-graphs [7], but those choices poorly adapt to the underlying data manifold and disregard label information, thus yielding suboptimal results. Dhillon et al. [8] propose a metric learning based framework in which a graph is constructed via a metric that maximizes the confidence of label assignments. Following a different route, Alexandrescu et al. [9] use a supervised model on the labeled subset to transform the data into a new space consisting in soft label predictions where the graph is constructed. These two approaches still rely on the classic heuristics for graph construction, and are not able to learn complex data representations. The recent work of Franceschi et al. [10] represents edges as Bernoulli random variables and uses a bilevel programming framework to fit the parameters of the graph and a GCN for semi-supervised classification.

In this paper, we present a probabilistic model to learn the parameters of a semi-supervised classification model and the graph jointly. We model edges as latent variables, and we learn by minimizing a reconstruction error over the predicted labels. Our choice of a probabilistic framework allow us to explicitly define a prior over the graph. This enables the model to account for prior knowledge and provides a principled mechanism to impose specific structures (such as sparsity) upon the graph.

## 2 Model

Let us assume a training set of the form $\mathcal{D} = \mathcal{D}_L \cup \mathcal{D}_U$ where $\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^{l}$ is the set of labeled data points with labels in some discrete set $\mathcal{Y}$, and $\mathcal{D}_U = \{x_i\}_{i=l+1}^{l+u}$ the set of unlabeled points. Let $X = [x_1; \ldots; x_{l+u}]^T$ be the design matrix and $y_L = [y_1, \ldots, y_l]$ the label vector, $W_{ij}$ a random variable representing an edge between $x_i$ and $x_j$. Our goal is to find labels $\{y_{l+1}, \ldots, y_{l+u}\}$.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

We start by assuming labels $y$ are generated by a random process that depends on the data $X$ and unknown parameters $W$ that encode how data points are connected. That is, $W_{ij}$ represents an edge between $x_i$ and $x_j$. We can describe this process through the conditional probability $p_\theta(y|X, W)$ parameterized by $\theta$ that gives the likelihood of labels $y$ provided the dataset $X$ and the graph $W$. In a variational Bayesian context parameters $W$ are latent variables with prior distribution $p_\theta(W)$ and approximate posterior $q_\phi(W_{ij}|X, y)$ parameterized by $\phi$, on which we can make inference.

We therefore aim to find the parameters $[\theta, \phi]$ so as to maximize the likelihood of the labels while keeping the distribution $q_\phi$ close to the prior. To do this, we maximize the evidence lower bound (ELBO) given by:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(W|X,y)}[\log p_\theta(y|W, X)] - \mathrm{KL}[q_\phi(W|X, y)||p_\theta(W)] \tag{1}$$

This model is similar to a variational autoencoder [11] where $q_\phi$ is the encoder and $p_\theta$ the decoder. Maximizing this bound is equivalent to minimizing the reconstruction error of the estimations the decoder produces while the encoder remains as close to the prior as possible.

To instantiate our model, we choose $p_\theta(y|W, X)$ to be a categorical distribution over the labels $\mathcal{Y}$ parametrized by a GCN [6], and $q_\phi(W|X, y)$ to model edges as a collection of Bernoulli distributions parameterized by a Graph Neural Network (GNN) [12]. Each Bernoulli distribution represents the probability of an edge connecting two nodes $i$ and $j$, hence $W_{ij}$ is a binary variable.

We took inspiration from [13] to use a message passing Graph Neural Network (GNN) [14] that computes node and edge embeddings as follows:

$$h_i^{(1)} = \mathrm{MLP}_{\text{node}}^{(1)}(x_i), \tag{2}$$

$$h_{ij}^{(1)} = \mathrm{MLP}_{\text{edge}}^{(1)}([h_i^{(1)}, h_j^{(1)}]), \tag{3}$$

$$h_i^{(2)} = \mathrm{MLP}_{\text{node}}^{(2)}\left(\left[h_i^{(1)}, \sum_{j \in \mathcal{N}(i)} h_{ij}^{(1)}\right]\right), \tag{4}$$

$$h_{ij}^{(2)} = \mathrm{MLP}_{\text{edge}}^{(2)}([h_i^{(2)}, h_j^{(2)}]), \tag{5}$$

and finally $g_\phi(X) = \mathrm{Softmax}(h_{ij}^{(2)})$. $\mathcal{N}(i)$ denotes the neighbors of point $i$ in the GNN (in practice we simply take them to be the closest points to $x_i$, or even all other points). Finally, we define $g_\phi(X) = \mathrm{Softmax}(h_{ij}^{(2)})$. Here $[x_i, x_j]$ denotes concatenation and MLP is short for multilayer perceptron.

We also pick the prior distribution $p_\theta(W)$ to be a collection of Bernoulli distributions.

**Training details.** We train our model by backpropagation. We first run the encoder $q_\phi$, which is a distribution taking discrete values. This prevent us from being able to directly backpropagate the error through its reparametrized samples. We then use the concrete distribution [15] to get a continuous approximation of $q_\phi$ and apply the reparametrization trick to compute the gradients. More specifically, we draw samples $W$ as follows: we draw a vector $\xi$ from a $\mathrm{Gumbel}(0, 1)$ distribution and then we compute $W_{ij} = \mathrm{Softmax}((h_{ij}^{(2)} + \xi)/\tau)$, where $\tau$ is a parameter controlling how smooth the resulting distribution is (the bigger $\tau$ is, the more it will resemble a uniform distribution). To control the variability we take several samples this way, $W^{(1)}, \ldots, W^{(r)}$ and feed the decoder to get $\hat{y}^{(1)}, \ldots, \hat{y}^{(r)}$ where $\hat{y}^i = GCN(X, W^{(i)})$. We then backpropagate the error with respect to $y_L$ through the mean $\hat{y} = \frac{1}{r}(\hat{y}^{(1)} + \cdots + \hat{y}^{(r)})$.

The reconstruction error that corresponds to the first term of Equation 1 is the average cross-entropy over the labeled examples:

$$\mathcal{L}_{\text{reconstruction}} = \frac{1}{l}\mathrm{CrossEnt}\big(y_L, \hat{y}\big|_{\mathcal{D}_L}\big). \tag{6}$$

The $KL$-divergence of $q_\phi$ that corresponds to the second term of Equation 1, given a Bernoulli prior $\rho$, is given by:

$$\mathcal{L}_{KL\text{-divergence}} = \sum_{i=1}^{l+u} \sum_{j:\rho_{ij} \neq 0} W_{ij} \log \frac{W_{ij}}{\rho_{ij}} + (1 - W_{ij}) \log \frac{1 - W_{ij}}{1 - \rho_{ij}}. \tag{7}$$

Table 1: Statistics of datasets.

| Dataset | Size | Dimension | Nb. of classes | Train/Val/Test |
|---|---|---|---|---|
| Wine | 178 | 13 | 3 | $10/20/158$ |
| 20news3 | 2756 | 229 | 3 | $20/40/2696$ |
| Cora | 2708 | 1433 | 7 | $140/300/1000$ |

Table 2: Mean and standard deviation of test accuracy over five random splits.

| | Wine | 20news3 | Cora |
|---|---|---|---|
| LogReg | $\mathbf{.95 \pm .02}$ | $.77 \pm .01$ | $.53 \pm .00$ |
| SVM | $.94 \pm .03$ | $.76 \pm .00$ | $.50 \pm .00$ |
| FFNN | $.93 \pm .01$ | $.77 \pm .01$ | $.55 \pm .01$ |
| GCN+$k$nn | $\mathbf{.95 \pm .03}$ | $.77 \pm .02$ | $.61 \pm .01$ |
| GCN+S$k$nn | $.93 \pm .03$ | $.66 \pm .05$ | $.31 \pm .00$ |
| GCN+RBF | $.94 \pm .03$ | $.76 \pm .01$ | $.51 \pm .01$ |
| PSSL | $\mathbf{.95 \pm .01}$ | $\mathbf{.83 \pm .01}$ | $\mathbf{.65 \pm .04}$ |

## 3 Experiments and Results

We carried out experiments to compare our probabilistic approach (denoted by PSSL) with supervised algorithms such as logistic regression (LogReg), support vector machines with a radial kernel (SVM) and feed-forward neural networks (FFNN),[1] and with the state-of-the-art semi-supervised method based on GCNs, which we fed with different types of heuristically computed graphs. We use three strategies for building a graph: $k$ nearest neighbors ($k$nn), radial kernel (RBF), and a random variant of the $k$-NN graph constructed as follows: denoting by $K$ the regular $k$-nn graph, an edge $e_{ij}$ between $x_i$ and $x_j$ is sampled according to a Bernoulli distribution with some high probability $\alpha$ if $K_{ij} = 1$, or with probability $1 - \alpha$ otherwise (S$k$nn). The prior distribution for our model PSSL is constructed in the same way as S$k$nn. We also specified different sparsity patterns over the prior.

We evaluate the baselines and our method on three datasets: Cora [16], a subset of 20 Newsgroups with three classes and a TFIDF feature space, and Wine, a benchmark dataset available in scikit-learn [17].

We used Adam to optimize our objective function. For all methods we tune the main hyperparameters over five random splits with train, test, and validation sizes as described in Table 1.

Results are shown in Table 2. We can observe that we achieve competitive results in Wine, and outperform the baselines by a considerable margin in 20news3 and Cora.

## 4 Conclusion and Discussion

We presented preliminary work on a framework based on autoencoding variational bayes that learns the parameters of a semi-supervised model and the underlying graph structure of the data simultaneously. We empirically showed that our model can achieve considerable gains over different baselines in different semi-supervised datasets.

We plan to run experiments on other semi-supervised datasets, and to compare this method empirically with that of Franceschi et al. [10]. We believe our proposal exhibits two advantages. First, we can explicitly specify a prior over the graph, which allow us to bias towards specific structures and sparsity patterns. Second, [10] requires two separate validation sets while we require only one: we therefore have access to more training data.

An interesting future research line is to extend this work to an inductive setting in order to be able to elegantly handle unseen test examples.

[1] This model is equivalent to a GCN with no graph.

## References


[1] Xiaojin Zhu and Zoubin Ghahramani. Learning from Labeled and Unlabeled Data with Label Propagation. Technical Report CMU-CALD-02-107, 2002.

[2] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised Learning Using Gaussian Fields and Harmonic Functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 912–919. AAAI Press, 2003.

[3] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.

[4] Thorsten Joachims. Transductive Learning via Spectral Graph Partitioning. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 290–297. AAAI Press, 2003.

[5] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *J. Mach. Learn. Res.*, 7:2399–2434, December 2006.

[6] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[7] Amarnag Subramanya and Partha Pratim Talukdar. *Graph-Based Semi-Supervised Learning*. Morgan & Claypool Publishers, 2014.

[8] Paramveer Dhillon, Partha Talukdar, and Koby Crammer. Inference Driven Metric Learning (IDML) for Graph Construction. *Technical Reports (CIS)*, 07 2010.

[9] Andrei Alexandrescu and Katrin Kirchhoff. Data-Driven Graph Construction for Semi-Supervised Graph-Based Learning in NLP. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 204–211, Rochester, New York, April 2007. Association for Computational Linguistics.

[10] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning Discrete Structures for Graph Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 1972–1982, 2019.

[11] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[12] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *Trans. Neur. Netw.*, 20(1):61–80, January 2009.

[13] Thomas N. Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard S. Zemel. Neural Relational Inference for Interacting Systems. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 2693–2702, 2018.

[14] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. *CoRR*, abs/1704.01212, 2017.

[15] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[16] Prithviraj Sen, Galileo Mark Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective Classification in Network Data. *AI Magazine*, 29(3):93–106, 2008.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.