A Consistent Diffusion-Based Algorithm for Semi-Supervised Graph Learning

Thomas Bonald, Nathan de Lara

A Consistent Diffusion-Based Algorithm for Semi-Supervised Graph Learning

Thomas Bonald* and Nathan De Lara

Télécom Paris, Institut Polytechnique de Paris, France

Abstract. The task of semi-supervised classification aims at assigning labels to all nodes of a graph based on the labels known for a few nodes, called the seeds. One of the most popular algorithms relies on the principle of heat diffusion, where the labels of the seeds are spread by thermo-conductance and the temperature of each node at equilibrium is used as a score function for each label. In this paper, we prove that this algorithm is not consistent unless the temperatures of the nodes at equilibrium are centered before scoring. This crucial step does not only make the algorithm provably consistent on a block model but brings significant performance gains on real graphs.

1 Introduction

The principle of heat diffusion has proved instrumental in graph mining [5]. It has been applied for many different tasks, including pattern matching [9], ranking [7], embedding [4], clustering [10], classification [13, 12, 2, 6] and feature propagation [8]. In this paper, we focus on the task of semi-supervised node classification: given labels known for a few nodes of the graph, referred to as the *seeds*, how to infer the labels of the other nodes? This can be viewed as a problem of heat diffusion with boundary constraints, known as the Dirichlet problem [13]. Specifically, one Dirichlet problem is solved per label, setting at 1 the temperature of the seeds with this label and at 0 the temperature of the other seeds. Each node is then assigned the label with the highest temperature over the different Dirichlet problems. In this paper, we prove using a simple block model that this algorithm is actually not consistent, unless the temperatures are *centered* before label assignement. This step of temperature centering does not only make the algorithm consistent but also brings substantial performance gains on real datasets. This is a crucial observation given the popularity of the algorithm[1].

The rest of this paper is organized as follows. In section 2, we introduce the Dirichlet problem on graphs. Section 3 describes our algorithm for node classification. The analysis showing the consistency of our algorithm on a simple block model is presented in section 4. Section 5 presents the experiments and section 6 concludes the paper.

* Contact author: thomas.bonald@telecom-paris.fr

[1] The number of citations of the paper [13] exceeds 4 000 according to Google Scholar.

2 Dirichlet problem on graphs

In this section, we introduce the Dirichlet problem on graphs and characterize the solution, used later in the analysis.

2.1 Heat equation

Consider an undirected graph G with n nodes indexed from 1 to n. Denote by A its adjacency matrix. This is a symmetric matrix with non-negative entries. Let $d = A1$ be the degree vector, which is assumed positive, and $D = \text{diag}(d)$. The Laplacian matrix is defined by:

$$L = D - A.$$

Now let S be some strict subset of $\{1, \dots, n\}$ and assume that the temperature of each node $i \in S$ is set at some fixed value T_i. We are interested in the evolution of the temperatures of the other nodes, we refer to as the *free* nodes. Heat exchanges occur through each edge of the graph proportionally to the temperature difference between the corresponding nodes so that:

$$\forall i \notin S, \quad \frac{dT_i}{dt} = \sum_{j=1}^{n} A_{ij}(T_j - T_i),$$

that is,

$$\forall i \notin S, \quad \frac{dT_i}{dt} = -(LT)_i,$$

where T is the vector of temperatures. This is the heat equation in discrete space. At equilibrium, the vector T satisfies Laplace's equation:

$$\forall i \notin S, \quad (LT)_i = 0. \tag{1}$$

With the boundary constraint giving the temperature T_i for each node $i \in S$, this defines a Dirichlet problem. Observe that Laplace's equation (1) can be written equivalently:

$$\forall i \notin S, \quad T_i = (PT)_i, \tag{2}$$

where $P = D^{-1}A$ is the transition matrix of the random walk in the graph.

2.2 Solution to the Dirichlet problem

We now characterize the solution to the Dirichlet problem (1). Without any loss of generality, we assume that free nodes (i.e., not in S) are indexed from 1 to $n - s$ so that the vector of temperatures can be written

$$T = \begin{bmatrix} X \\ Y \end{bmatrix},$$

where X is the vector of temperatures of free nodes at equilibrium, of dimension $n-s$, and Y is the vector of temperatures of the seeds, of dimension s. Writing the transition matrix in block form as

$$P = \begin{bmatrix} Q & R \\ \cdot & \cdot \end{bmatrix},$$

it follows from (2) that:

$$X = QX + RY, \tag{3}$$

so that:

$$X = (I - Q)^{-1} RY. \tag{4}$$

Note that the inverse of $I - Q$ exists whenever the graph is connected [3]. The solution to the Dirichlet problem exists and is unique.

3 Node classification algorithm

In this section, we introduce a node classification algorithm based on the Dirichlet problem. The objective is to infer the labels of all nodes given the labels of a few nodes called the *seeds*. Our algorithm is a simple modification of the popular method proposed by [13]. Specifically, we propose to *center* temperatures before label assignment.

3.1 Binary classification

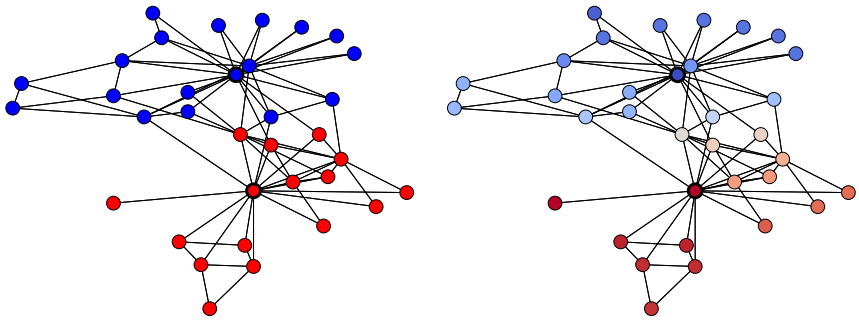
When there are only two different labels, say 0 and 1, the classification follows from the solution of a single Dirichlet problem. The idea is to set at 0 the temperature of seeds with label 0 and at 1 the temperature of seeds with label 1. The solution to this Dirichlet problem gives temperatures between 0 and 1 to the free nodes, as illustrated by Figure 1 for the Karate Club graph [11].

A natural decision rule is to use a threshold of $1/2$ for classification: any free node with temperature above $1/2$ is assigned label 1, any other free node is assigned label 0. The analysis of Section 4 suggests that it is preferable to set the threshold to the mean temperature at equilibrium,

$$\bar{T} = \frac{1}{n} \sum_{i=1}^{n} T_i. \tag{5}$$

Specifically, any free node with temperature above $\bar{T}$ is assigned label 1, any other free node is assigned label 0. Equivalently, temperatures are *centered* before classification: after centering, free nodes with positive temperature are assigned label 1, the others are assigned label 0.

It is worth noting that the threshold (5) is the mean temperature of *all* nodes at equilibrium, including seed nodes. Another option, suggested by the *class mass normalization* step of [13] for instance, is to set the threshold at the mean temperature of *free* nodes at equilibrium. This variant of the algorithm is not provably consistent, however.



(a) Ground truth (b) Solution to the Dirichlet problem

Fig. 1: Binary classification of the Karate Club graph with 2 seeds (indicated with a black circle). Blue nodes have label 0, red nodes have label 1.

3.2 Multi-class classification

In the general case with K labels, we use a *one-against-all* strategy: the seeds of each label alternately serve as hot sources (temperature 1) while all the other seeds serve as cold sources (temperature 0). After centering the temperatures (so that the mean temperature of each diffusion is equal to 0), each node is assigned the label that maximizes its temperature. This algorithm, we refer to as the Dirichlet classifier, is parameter-free.

Algorithm 1 Dirichlet classifier

Require: Seed set S and associated labels $y \in \{1, \dots, K\}$
1: **for** k in $\{1, \dots, K\}$ **do**
2: $T = 0$
3: **for** $i \in S$ **do**
4: **if** $y_i = k$ **then**
5: $T_i = 1$
6: **end if**
7: **end for**
8: $T \leftarrow \text{Dirichlet}(S, T)$
9: $\Delta(k) \leftarrow T - \frac{1}{n}\sum_{i=1}^{n} T_i$
10: **end for**
11: **for** $i \notin S$ **do**
12: $\hat{y}_i = \arg\max_{k=1,\dots,K}(\Delta_i(k))$
13: **end for**
14: **return** $\hat{y}$, predicted labels of free nodes (outside S)

The solution to the Dirichlet problem (line 8 of the algorithm) can be obtained either from (4) or from iterations of the fixed-point equation (3).

4 Analysis

In this section, we prove the consistency of Algorithm 1 on a simple block model. In particular, we highlight the importance of temperature centering (line 9 of the algorithm) for the consistency of the algorithm.

4.1 Block model

Consider a graph of n nodes consisting of K blocks of respective sizes $n_1, \dots, n_K$, forming a partition of the set of nodes. There are $s_1, \dots, s_K$ seeds in these blocks, which are respectively assigned labels $1, \dots, K$. Intra-block edges have weight p and inter-block edges have weight q. We expect the algorithm to assign label k to all nodes of block k whenever $p > q$, for all $k = 1, \dots, K$.

4.2 Dirichlet problem

Consider the Dirichlet problem when the temperature of the s_1 seeds of block 1 is set to 1 and the temperature of the other seeds is set to 0. We have an explicit solution to this Dirichlet problem, given by Lemma 1. All proofs are deferred in the appendix.

Lemma 1. *Let T_k be the temperature of free nodes of block k at equilibrium. We have:*

$$\begin{aligned}(s_1(p-q)+nq)T_1 &= s_1(p-q)+n\bar{T}q,\\ (s_k(p-q)+nq)T_k &= n\bar{T}q \qquad k=2,\dots,K,\end{aligned}$$

where $\bar{T}$ is the average temperature, given by:

$$\bar{T} = \frac{1}{n}\sum_{i=1}^{n} T_i = \left(\frac{s_1}{n}\frac{n_1(p-q)+nq}{s_1(p-q)+nq}\right) \Big/ \left(1 - \sum_{k=1}^{K}\frac{(n_k-s_k)q}{s_k(p-q)+nq}\right).$$

4.3 Classification

We now state the main result of the paper: the Dirichlet classifier is a consistent algorithm for the block model, in the sense that all nodes are correctly classified whenever $p > q$.

Theorem 1. *If $p > q$, then $x_i = k$ for each free node i of each block k, for any parameters $n_1, \dots, n_K$ (label distribution) and $s_1, \dots, s_K$ (seed distribution).*

Observe that the temperature centering is critical for consistency. In the absence of centering, free nodes of block 1 are correctly classified if and only if

their temperature is the highest in the Dirichlet problem associated with label 1. In view of Lemma 1, this means that for all $k = 2, \dots, K$,

$$s_1 q \frac{n_1(p-q)+nq}{s_1(p-q)+nq} + s_1(p-q)\left(1 - \sum_{j=1}^{K} \frac{(n_j - s_j)q}{s_j(p-q)+nq}\right) > s_k q \frac{n_k(p-q)+nq}{s_k(p-q)+nq}.$$

This condition might be violated even if $p > q$, depending on the parameters $n_1, \dots, n_K$ and $s_1, \dots, s_K$. In the simple case of $K = 2$ blocks with $p = 0.1$ and $q = 0.01$ for instance, the classification is incorrect in the cases $n_1 = n_2 = 100, s_1 = 10, s_2 = 5$ (seed asymmetry) and $n_1 = 100, n_2 = 10, s_1 = s_2 = 5$ (label asymmetry). This sensitivity of the algorithm to both forms of asymmetry will be confirmed by the experiments. The step of temperature centering is crucial for consistency.

5 Experiments

In this section, we show the impact of temperature centering on the quality of classification using both synthetic and real data. The Python code is available as a Jupyter notebook in Python[2], making the experiments fully reproducible.

5.1 Synthetic data

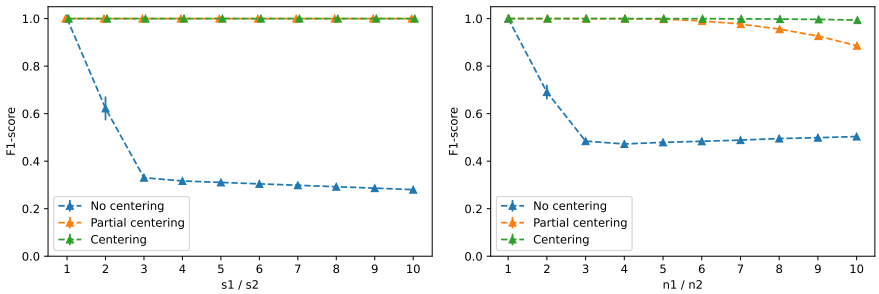
We first use the stochastic block model [1] to generate graphs with an underlying structure in clusters. This is the stochastic version of the block model used in the analysis. There are K blocks of respective sizes $n_1, \dots, n_K$. Nodes of the same block are connected with probability p while nodes in different blocks are connected probability q. Nodes in block k have label k. We denote by s_k the number of seeds in block k and by s the total number of seeds.

We first compare the performance of the algorithms on a binary classification task ($K = 2$) for a graph of $n = 10\,000$ nodes with $p = 10^{-2}$ and $q = 10^{-3}$, in two different settings:

- **Seed asymmetry:** Both blocks have the same size $n_1 = n_2 = 5000$ but different numbers of seeds, with ratio $s_1/s_2 \in \{1, 2, \dots, 10\}$ and $s_2 = 250$ (5% of nodes in block 2 are seeds).
- **Label asymmetry:** Both blocks have different sizes, with ratio $n_1/n_2 \in \{1, 2, \dots, 10\}$ and total size $n = 10\,000$, but the same number of seeds, $s_1 = s_2 = 250$ (5% of all nodes are seeds).

For each configuration, the experiment is repeated 100 times. Randomness comes both from the generation of the graph and from the selection of the seeds.

[2] https://perso.telecom-paris.fr/bonald/notebooks/diffusion.ipynb

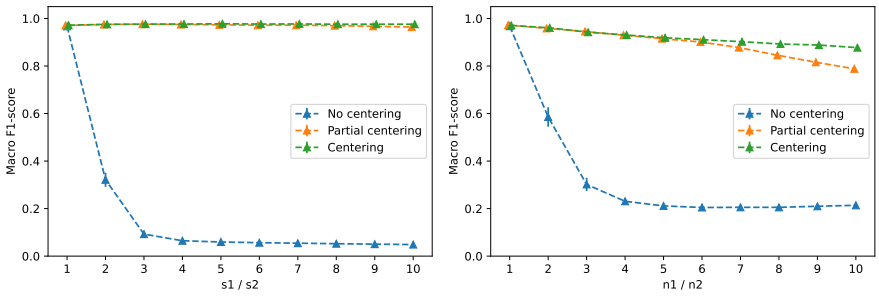


(a) Seed asymmetry (b) Label asymmetry

Fig. 2: F1 scores on the stochastic block model (2 labels).

We report the F1-scores in Figure 2 (mean ± standard deviation). Observe that the variability of the results is very low due to the relatively large size of the graph. As expected, the centered version is much more robust to both forms of asymmetry. The variant called *partial centering*, where the mean temperature is computed over free nodes only, tends to be less robust to label asymmetry.

We show in Figure 3 the same results for $K = 5$ blocks, still with $n = 10\,000$ nodes, $p = 10^{-2}$ and $q = 10^{-3}$. Blocks $2, 3, 4, 5$ have the same size and the same number of seeds. For the experiments on seed asymmetry, each block has $2\,000$ nodes and 5% of nodes in blocks $2, 3, 4, 5$ are seeds; we only vary the number of seeds in block 1. For the experiments on label asymmetry, there is the same number of seeds for each label, corresponding to an average proportion of 5% of all nodes. The performance metric is the F1-score averaged over the 5 labels. The conclusions are the same as with 2 labels.



(a) Seed asymmetry (b) Label asymmetry

Fig. 3: Macro F1-scores on the stochastic block model (5 labels).

5.2 Real data

We now focus on real datasets available from the SNAP collection[3] and the NetSet[4] collection, restricting to graphs having ground-truth labels. All graphs are considered undirected.

Table 1: Overview of the datasets.

Dataset	#nodes	#edges	#classes
Cora	2 708	5 278	7
Citeseer	3 264	4 536	6
PubMed	19 717	44 325	3
Email	1 005	16 385	42
PolBlogs	1 490	16 716	2
WikiSchools	4 403	100 329	16
WikiVitals	10 011	654 502	11
WikiVitals+	45 179	3 079 335	11

For each dataset, we select seeds uniformly at random. The process is repeated 100 times. The macro-F1 scores are shown in Table 2 for seeds representing $5\%, 10\%$ or 20% of the nodes. We see that the centered version outperforms the standard version over all datasets. The performance gains are substantial for the largest graphs, extracted from Wikipedia. The variance is also lower in all cases, showing the robustness of the algorithm. Additional results, not reported here, tend to show that the variant selected for temperature centering (based on either all nodes or free nodes) has a marginal impact on performance.

6 Conclusion

We have proposed a novel approach to node classification based on heat diffusion. Specifically, our technique consists in centering the temperatures of each solution to the Dirichlet problem before classification. We have proved the consistency of this algorithm on a simple block model and shown that the temperature centering brings significant performance gains on real datasets. This is a crucial observation given the popularity of the algorithm.

The question of the consistency of the algorithm when the mean temperature is computed over free nodes (instead of all nodes) remains open. Another interesting research perspective is to extend our proof of consistency of the algorithm to *stochastic* block models, where edges are drawn at random [1] .

[3] https://snap.stanford.edu/
[4] https://netset.telecom-paris.fr/

Appendix

A Proof of Lemma 1

Proof. In view of (2), we have:

$$(n_1(p-q)+nq)T_1 = s_1 p + (n_1 - s_1)pT_1 + \sum_{j\neq 1}(n_j - s_j)qT_j,$$
$$(n_k(p-q)+nq)T_k = s_1 q + (n_k - s_k)pT_k + \sum_{j\neq k}(n_j - s_j)qT_j,$$

for $k = 2, \ldots, K$. We deduce:

$$\begin{aligned}(s_1(p-q)+nq)T_1 &= s_1 p + Uq,\\ (s_k(p-q)+nq)T_k &= s_1 q + Uq \qquad \forall k = 2,\ldots,K,\end{aligned}$$

with

$$U = \sum_{j=1}^{K}(n_j - s_j)T_j.$$

The proof then follows from the fact that

$$n\bar{T} = s_1 + \sum_{j=1}^{K}(n_j - s_j)T_j = s_1 + U.$$

B Proof of Theorem 1

Proof. Let $\Delta_k^{(1)} = T_k - \bar{T}$ be the deviation of temperature of non-seed nodes of block k for the Dirichlet problem associated with label 1. In view of Lemma 1, we have:

$$\begin{aligned}(s_1(p-q)+nq)\Delta_1^{(1)} &= s_1(p-q)(1-\bar{T}),\\ (s_k(p-q)+nq)\Delta_k^{(1)} &= -s_k(p-q)\bar{T} \qquad k = 2,\ldots,K,\end{aligned}$$

For $p > q$, using the fact that $\bar{T} \in (0,1)$, we get $\Delta_1^{(1)} > 0$ and $\Delta_k^{(1)} < 0$ for all $k = 2, \ldots, K$. By symmetry, for each label $l = 1, \ldots, K$, $\Delta_l^{(l)} > 0$ and $\Delta_k^{(l)} < 0$ for all $k \neq l$. We deduce that for each block k, $x_i = \arg\max_l \Delta_k^{(l)} = k$ for each free node i of block k.

Table 2: Macro-F1 scores (mean ± standard deviation) without and with temperature centering.

(a) 5% of seeds

Dataset	No centering	Centering	Variation
Cora	0.69 ± 0.02	**0.71** ± 0.02	+2%
Citeseer	**0.48** ± 0.01	**0.48** ± 0.01	0%
PubMed	0.76 ± 0.01	**0.78** ± 0.01	+2%
Email	0.12 ± 0.04	**0.22** ± 0.03	+85%
PolBlogs	0.82 ± 0.12	**0.87** ± 0.01	+7%
WikiSchools	0.08 ± 0.06	**0.44** ± 0.03	+472%
WikiVitals	0.29 ± 0.06	**0.63** ± 0.02	+116%
WikiVitals+	0.31 ± 0.03	**0.65** ± 0.01	+112%

(b) 10% of seeds

Dataset	No centering	Centering	Variation
Cora	0.74 ± 0.02	**0.75** ± 0.01	+1%
Citeseer	**0.52** ± 0.01	**0.52** ± 0.01	0%
PubMed	0.78 ± 0.01	**0.79** ± 0.00	+1%
Email	0.21 ± 0.04	**0.31** ± 0.03	+43%
PolBlogs	0.86 ± 0.02	**0.87** ± 0.01	+1%
WikiSchools	0.13 ± 0.04	**0.50** ± 0.02	+295%
WikiVitals	0.43 ± 0.04	**0.67** ± 0.01	+57%
WikiVitals+	0.61 ± 0.01	**0.68** ± 0.01	+12%

(c) 20% of seeds

Dataset	No centering	Centering	Variation
Cora	**0.78** ± 0.01	**0.78** ± 0.01	0%
Citeseer	**0.57** ± 0.01	**0.57** ± 0.01	0%
PubMed	**0.80** ± 0.00	**0.80** ± 0.00	0%
Email	0.32 ± 0.03	**0.40** ± 0.02	+24%
PolBlogs	**0.87** ± 0.01	**0.87** ± 0.01	0%
WikiSchools	0.27 ± 0.03	**0.57** ± 0.02	+110%
WikiVitals	0.58 ± 0.02	**0.70** ± 0.01	+22%
WikiVitals+	0.65 ± 0.01	**0.71** ± 0.00	+9%

References

1. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. Journal of machine learning research (2008)
2. Berberidis, D., Nikolakopoulos, A.N., Giannakis, G.B.: Adadif: Adaptive diffusions for efficient semi-supervised learning over graphs. In: International Conference on Big Data. IEEE (2018)
3. Chung, F.R.: Spectral graph theory. American Mathematical Soc. (1997)
4. Donnat, C., Zitnik, M., Hallac, D., Leskovec, J.: Learning structural node embeddings via diffusion wavelets. In: International Conference on Knowledge Discovery & Data Mining. ACM (2018)
5. Kondor, R.I., Lafferty, J.: Diffusion kernels on graphs and other discrete structures. In: Proceedings of the 19th international conference on machine learning (2002)
6. Li, Q., An, S., Li, L., Liu, W.: Semi-supervised learning on graph with an alternating diffusion process. CoRR (2019)
7. Ma, H., King, I., Lyu, M.R.: Mining web graphs for recommendations. IEEE Transactions on Knowledge and Data Engineering (2011)
8. Rossi, E., Kenlay, H., Gorinova, M.I., Chamberlain, B.P., Dong, X., Bronstein, M.M.: On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features. In: Proceedings of Machine Learning Research (2022)
9. Thanou, D., Dong, X., Kressner, D., Frossard, P.: Learning heat diffusion graphs. IEEE Transactions on Signal and Information Processing over Networks (2017)
10. Tremblay, N., Borgnat, P.: Graph wavelets for multiscale community mining. IEEE Transactions on Signal Processing (2014)
11. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of anthropological research (1977)
12. Zhu, X.: Semi-supervised learning with graphs. Ph.D. thesis, Carnegie Mellon University (2005)
13. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International conference on Machine learning (2003)