Semi-supervised regression using diffusion on graphs

Mohan Timilsina, Alejandro Figueroa, Mathieu D'aquin, Haixuan Yang

Applied Soft Computing Journal 104 (2021) 107188

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Semi-supervised regression using diffusion on graphs

Mohan Timilsina [a,*], Alejandro Figueroa [b], Mathieu d'Aquin [a], Haixuan Yang [c]

[a] Data Science Institute, Insight Centre for Data Analytics, National University of Ireland Galway, Ireland
[b] Departamento de Ciencias de la Ingeniería, Facultad de Ingeniería, Universidad Andres Bello, Antonio Varas 880, Santiago, Chile
[c] School of Mathematics, Statistics and Applied Mathematics, National University of Ireland Galway, Ireland

ARTICLE INFO

Article history:
Received 2 December 2019
Received in revised form 12 January 2021
Accepted 9 February 2021
Available online 18 February 2021

Keywords:
Network
Boundary heat diffusion
Prediction
Label

ABSTRACT

In real-world machine learning applications, unlabeled training data are readily available, but labeled data are expensive and hard to obtain. Therefore, semi-supervised learning algorithms have gathered much attention. Previous studies in this area mainly focused on a semi-supervised classification problem, whereas semi-supervised regression has received less attention. In this paper, we proposed a novel semi-supervised regression algorithm using heat diffusion with a boundary-condition that guarantees a closed-form solution. Experiments from artificial and real datasets from business, biomedical, physical, and social domain show that the boundary-based heat diffusion method can effectively outperform the top state of the art methods.

1. Introduction

There are large amounts of unlabeled data available in real-world machine learning applications. The labeling data is often laborious, or expensive, as it requires the effort of human experts for annotation. A typical example is in speech recognition, it costs almost nothing to record huge amounts of speech, but labeling it requires humans to listen and transcribe. This process is burdensome and time-consuming. In such a case, "semi-supervised learning (SSL)" becomes handy to tackle the few labeled data and large unlabeled data. In the supervised learning framework, a set of l independently identically distributed (iid) examples $x_1, \ldots, x_l \in X$ with corresponding labels $y_1, \ldots, y_l \in Y$ are given. Furthermore, u unlabeled data $x_{l+1}, \ldots, x_{l+u} \in X$ are provided. SSL tackles this combined information to improve classification performance. The few prominent SSL methods that have been effectively used in various fields with notable results are Self-training [1], Co-training [2], Tri-training [3] and Transductive SVM (Support Vector Machines) [4]. These methods exploit as unlabeled data as possible and have produced quality results.

Semi-Supervised Classification (SSC) is famous due to its ability to solve pattern recognition problems [5–7]. Most studies deal with the application of SSC techniques in many real-world problems in contrast to Semi-Supervised Regression (SSR) [8], which is a more common but slightly explored case. In the SSC, the independent variable Y_i is constrained to have only a finite number of possible values, whereas, in the SSR, the Y_i is assumed to be continuous. Hence, SSC algorithms designed for graph min-cut [9] do not apply to the more general SSR problem. Other algorithms, such as Gaussian Fields [10], apply to both SSR and SSC by using graphs.

The above discussion commonly justifies the development of SSR using a graph-based propagation method. Each graph-based diffusion has a different method of representation. For instance, PageRank uses a geometrically weighted sum of random walks; Heat diffusion uses an exponentially weighted sum of random walks [11]. This kind of diffusion expression affect in the performances. Yang et al. [12] showed that heat style diffusion is robust to web spamming in comparison to PageRank style diffusion. Another important observation is that many graph-based label propagation algorithms suffer from the problem of continuous diffusion. It means the label density is infinitely propagated in the network until the convergence is guaranteed [13–15]. While performing a random walk with continuous diffusion, algorithm explores more of the network by walking deeply. If the underlying network is of poor quality, this behavior eventually tends to produce meaningless diffusion values, leading to a large prediction error. Therefore, we need a diffusion function which can control the propagation depth so that we can effectively and efficiently predict the value for nodes in a network.

Main idea: In this paper, we propose graph-based SSR algorithm. This algorithm is very intuitive and natural based on the physical heat diffusion system with boundary conditions. The heat flow between points in the network is captured by measuring the amount of heat added or removed from the system. The points represent nodes in a graph, and heat flow between the points is the edges. The injection and extraction points of the

* Corresponding author.
E-mail addresses: mohan.timilsina@insight-centre.org (M. Timilsina), alejandro.figueroa@unab.cl (A. Figueroa), mathieu.daquin@insight-centre.org (M. d'Aquin), haixuan.yang@nuigalway.ie (H. Yang).

https://doi.org/10.1016/j.asoc.2021.107188

heat are the boundary of the system, which controls the heat flow. The final temperature distribution of the nodes after the heat diffusion process makes this technique ideal for a regression problem.

Contributions. Our contributions of heat diffusion with boundary condition are summarized as follows:

1. Accuracy: Our algorithm achieves relatively good prediction accuracy on different label propagation in regression datasets.
2. Closed-Form: The heat diffusion with boundary condition has closed-form solutions on any graph structures.
3. Parameter estimation: Heat diffusion with boundary condition has just one parameter with a default value of 1. It means there is no need for parameter tuning.

Moreover, we performed the extensive experiments using seven different regression datasets from different domains: (i) Sales prediction using TV advertisement (ii) Boston housing price prediction, (iii) White wine alcohol volume prediction, (iv) Red wine alcohol volume prediction, (v) Parkinson's patient sound level prediction (vi) Airfoil self noise prediction and (vii) Bike-sharing rental count prediction. The results demonstrated that our algorithm often outperforms state of the art label propagation algorithm in terms of prediction accuracy.

Our algorithm is a novel label propagation method which is motivated from physics inspired boundary-based heat diffusion to handle the graph-based regression problem relying only on the graph structure. To the best of our knowledge, our algorithm is the first solution to handle transductive graph-based semi supervised regression without any parameters to tune.

Outline. The rest of the paper is organized as: related work, problem definition, method description, experimental analysis, and conclusion.

2. Related work

The label propagation (LP) technique has various names, including graph-based semi-supervised and transductive learning. LP methods use diffusion mechanisms to propagate labels from a small set of nodes with known class labels to the remaining nodes of the graph.[1] The label propagation algorithms learn the labels of unlabeled nodes by diffusing information about local label density through the network. This behavior makes these algorithms faster and scales to large networks.

In transductive learning, Graph-based Laplacian Regularization (GLR) is a widely used [16]. GLR is based on the manifold assumption, which states that if two points are on the same manifold, their corresponding values are similar. This idea is one of the vital assumptions in graph-based SSL. Belkin et al. [17] demonstrated two types of SSL algorithms based on graph regularization showing that the exploitation of unlabeled data enhances the predictive performance. Similarly, Laplacian Regularized Least Square Regression (LapRLSR) method concerning the SSR framework regularized with a graph Laplacian prior has also been extended to build an efficient regressor is called Temporal Laplacian Regularized LS Regression (TLapRLSR) algorithm [18] in an image sequences application problem. Doquire and Verleysen [19] proposed a variant of the Laplacian method for feature selection algorithm named SSLS (Semi-Supervised Laplacian Score), which blends both supervised and unsupervised Laplacian Score methods for regression problems. Zhao et al. [20] combined the LapRLS with SSL Discriminant Analysis methods (SDA) and creating an SSL dimensionality reduction in a regression setting. On a similar note, the study by Sheng and Zhu [21] applied a regularized regressor integrated with quadratic loss inside a LapRLS framework, studying the correlation of the convergence rate.

There are several algorithms proposed that can solve the node classification problem from the LP perspective. Zhu et al. [15] proposed LP, which is one of the most well-known graph-based SSL algorithms in the Artificial Intelligence (AI) community. Zhou et al. [22] proposed another popular algorithm called Local and Global Consistency (LGC). Local means nearby points are likely to have the same label. Global means the points on the same structure are likely to have the same label. Most of graph-based methods are believed to work better on low-dimensional feature data in comparison to high dimensional data [23]. It is due to the fact that the graph is affected by the influence of noisy features of high-dimensional samples. Yu et al. [23] propose semi-supervised ensemble based approach to tackle that problem in subspaces.

Adsorption [24] and Modified Adsorption [25] search for the fixed point state where many connected nodes have the same class labels. These algorithms work best on the *homophily* (similar nodes may be more likely to attach than different ones) labeled network. Heat diffusion [12,26–29] style propagation has also been used in the SSC task due to its intuitive interpretations in terms of random walks, electrical circuits, and other aspects of spectral graph theory [13]. Chen et al. [30] demonstrated the weighting samples of labeled and unlabeled data to improve the graph-based semi-supervised classification. Label noise is also one of the critical issues in SSC to degrade classification accuracy [31,32]. Similarly, Wang et al. [33] proposed the discriminative graph with constrained k-means approach to avoid misclassifying boundary samples of different classes.

The label smoothness and locally estimated label penalties assumption used in graph-based SSC are also used neural network based regression model [34]. The graph construction from a feature data is itself another important problem in SSC for accurate label prediction [35]. Currently, Graph Convolutional Networks (GCN) [36,37] have shown impressive results in SSC, due to its ability that nicely integrates graph and feature information in each layer. The progress of GCN has motivated many influential works [38,39] on graph. Although these neural-network-based models tend to have stronger modeling capabilities than the classical graph-based approach, they typically require an ample amount of labeled data for training and validation due to high model complexity, hence may not be label efficient [40]. The major caveats of GCN is that it requires many additional labeled data for validation and suffers from the localized nature of the convolutions filter [38]. However, from the computational perspective, heat diffusion methods are promising because they are fast to compute in the sparse graphs [41,42] and robust in memory usage [43].

Yamaguchi et al. [44] proposed OMNI-Prop that applies to both *homophily* and *heterophily* (different nodes may be more likely to attach than similar ones) labeled network. All the models above have been proven useful on a node classification problem; only a few of them applied in regression problems.

The study by Wasserman and Lafferty [47]; El Alaoui et al. [48]; Mai and Couillet [49] showed the graph-oriented SSL algorithms and establish prediction properties of semi-supervised estimators from the number of features in the data. Cohen [50] demonstrated SSL in a directed graphs based on distance diffusion. As they consider distances from unlabeled to labeled nodes, each instance is computationally intensive and requires an approximation scheme. The major concern in these models is the computational complexity because they are slow to converge and unstable [51,52]. To overcome this problem, Rosenfeld et al. [53] proposed SSL with competitive infection models that consider

[1] Note that the words "graph" and "network" are interchangeably used in the paper.

Table 1
Qualitative comparison between different label propagation algorithms.

	Mincut Blum et al. [45]	HMN Zhu et al. [14]	LGC Zhou et al. [22]	BP Gatterbauer et al. [46]	OMNI Yamaguchi et al. [44]	HD Yang et al. [12]	BHD
Closed form solution	✓	✓	✓	×	✓	✓	✓
Convergence	✓	✓	✓	?	✓	✓	✓
No parameter tuning	✓	✓	×	×	✓	✓	✓
Regression	×	✓	✓	×	×	×	✓

distances from labeled to unlabeled nodes, which can be computed efficiently. Most of the SSL methods use spectral diffusions [54]. These methods include label propagation [15] and label propagation using the normalized graph Laplacian [22]. They scale well into repeated averaging over neighboring nodes and are used on massive graphs with billions of edges [55]. Recent work by Budninskiy et al. [56] showed the discrete differential geometry approach in a graph-based SSL problem where label diffusion uses a Laplacian operator learned from the geometry of the input data. They used the biconvex loss function in terms of graph edge weights and inferred labels. The function minimization is achieved through alternating rounds of optimization of the Laplacian and diffusion-based inference of labels. Thus the optimized results of the Laplacian diffusion directionally adapt to the intrinsic geometric structure of the data and give high accuracy in classifying labels. However, it is unsure that the same principle can be applied to the regression problem.

In the context of practical application, SSR showed the usefulness for instance predicting the final grade of undergraduate students in a distance online course by using a small number of students data from previous years [57]. The co-training style algorithm developed by Zhou and Li [58] demonstrated useful in SSR using graphs. Similarly, Wang et al. [59] proposed an algorithm, which is about the kernel regression framework exploiting both labeled and unlabeled examples. These algorithms are essential in the regression problem. However, they are not applicable without parameter tuning.

Belief propagation (BP) [60] is the propagation algorithm for performing inference on graphical models. This algorithm has been implemented to deal with various problems related to graphs such as a random walk with restart and label propagation. Papaspiliopoulos and Zanella [61] showed the usability of sampling multilevel regression models using belief propagation. Although BP is beneficial; its recursive calculation does not have any guarantee to converge on arbitrary graphs [44].

Table 1 shows the qualitative comparison between our algorithm and the major state of the art graph-based methods. Our algorithm is parameter-free, provides closed-form solutions, and guarantees convergence.

3. Problem formulation

This section details some terms and also introduces the graph regression problem. Suppose $\mathcal{N}$ is the list of nodes and E is the number of edges. For undirected graph $G = (\mathcal{N}, E)$, we have $E \subseteq \mathcal{N} \times \mathcal{N}$ also $\mathcal{N}_i \subseteq \mathcal{N}$. The set of nodes is composed of two types of components $\mathcal{N} = \mathcal{N}^L \cup \mathcal{N}^U$ where $\mathcal{N}^L = \{n_1, \ldots, n_L\}$ is a set of L labeled nodes and $\mathcal{N}^U = \{n_{L+1}, \ldots, n_{L+U}\}$ is the list of unlabeled nodes. Let Y be the set of possible labels and $Y_L = \{y_1, \ldots, y_L\}$ are the labels assigned to the nodes in $\mathcal{N}^L$. Thus, the graph regression problem is expressed as follows:

Problem (Graph regression)

- **Available:** A partially labeled graph.
- **Score:** Find the score $S_{i,j}$ which corresponds to the value of the unlabeled node i through labeled node j.

Table 2
Symbols and Definitions.

Symbols	Definitions
W	Adjacency matrix
N,E	# of nodes, # of edges
L	# of labeled nodes
U	#of unlabeled nodes
t	time
D	Degree matrix
f_u	Temperature distribution of the unlabeled node
f_l	Temperature distribution of the labeled node

- **Estimates:** The function estimates of the response variable:

  $\hat{S} = MS$

  where $\hat{S}$ are the new estimates, S are the observations and M is a matrix which may be constructed based on the data.

4. Methodology

We are given an undirected graph constructed from data features by applying a distance similarity metric. We follow the same smoothness assumption made by Zhu et al. [15] that nodes close to each other have similar values. This idea also applies to the regression problem [62]. Table 2 shows the list of symbols we used in the paper.

4.1. Graph construction

There are different methods to construct the graphs. We took a generic method to construct the graph.

- **Fully connected graph:** In the fully connected graph, where every pair of vertices x_i, x_j is connected by an edge. An edge between two vertices x_i, x_j represents the similarity of the two instances. One popular weight w_{ij} function used in a semi-supervised machine learning task is given by:

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{1}$$

  This function is also called a Gaussian kernel or a Radial Basis Function (RBF) kernel [10]. The edge weight decreases as the Euclidean distance $\|x_i - x_j\|$ increases, σ is known as the bandwidth parameter and chosen as $\frac{1}{n}$ and n is the number of features. The weight $w_{ij} = 1$ when $x_i = x_j$, and 0 when $x_i - x_j$ approaches to ∞. The advantage of a fully connected graph is in edge weight learning, with a differentiable weight function, one can easily take the derivatives of the graph with respect to weight hyperparameters [10]. However, it has a disadvantage in computational cost. For huge graphs, this matrix will be dense, thus using such graphs for label propagation will entail a high time complexity. To store all nonzero elements in this matrix (i.e., fully connected graph), we require huge memory space. As a means of avoiding this problem, sparse graphs can be constructed via k nearest neighbors (k-NN).

- **k-NN graph:** One can create k-NN graphs where each node connects to only a few nodes. Optimal K can be chosen using cross-validation in training data and such sparse graphs are computationally cheaper and faster. In a k-NN graph the vertices x_i, x_j are connected by an edge if x_i is in x_j's k-nearest-neighborhood or vice versa. The edge weight w_{ij} be either the constant 1, in the case when the graph is unweighted, or a function of a distance as in Eq. (1). k is a hyper-parameter that controls the density of the graph. If k is chosen very small then it may result into a disconnected graphs.

The above described graph construction approaches are very generic technique. The better graphs can be constructed if one has knowledge of the problem domain, and can define better distance functions, connectivity, and edge weights.

4.2. Heat diffusion

Heat is an energy which propagates from a body with a high temperature to a low temperature. This energy propagation idea have been successfully used in various domains such as web spamming in web graph analysis [12], recommender systems [63] and disease gene prioritization [64]. For a known graph structure, the heat flow with initial conditions can be defined by the following second order differential equation:

$$\frac{\partial f(x,t)}{\partial t} - \Delta f(x,t) = 0 \tag{2}$$

where f(x, t) is the heat at location x at time t, and Δ f is the Laplace–Beltrami operator on a function f. The heat diffusion kernel $K_t(x, y)$ is a special solution to the heat equation with an initial condition having a unit heat source at position y and no heat in another end. Heat kernel [13] have been proven to be useful because of the physical interpretation of the optimization in label propagation in a semi-supervised machine learning [15]. The solution to the heat diffusion equation on a graph is [12,26]:

$$f(t) = e^{-\alpha t H} f(0) \tag{3}$$

The value $f(t)$ illustrates the heat at node v at time t, beginning from an initial distribution of heat given by $f(0)$ at time zero and H is the graph Laplacian, and α is the diffusion coefficient.

4.3. Heat diffusion in a boundary condition in graph (BHD)

In the context of our work, we are considering diffusion in a boundary condition. By boundary condition we mean that we have some information about the solution at the endpoints.

Let us suppose that there are l labeled and u unlabeled nodes and $N = l+u$ be the total nodes in the multiplex graph. Then $L = \{1, 2, \ldots, l\}$ corresponds to labeled nodes with labels $f_1, \ldots, f_l$, and nodes $U = \{l+1, l+2, \ldots, l+u\}$ refers to the unlabeled points. Our job here is to assign the labels for the nodes U. The edge of the graphs is a $n \times n$ weight matrix W also known as adjacency matrix.

Now to formulate our model, let us assume that, at time t, each node $i \in U$, receives a certain amount of heat $M(i, j, t, \Delta t)$ from its neighbor j during a period of Δt. The heat $M(i, j, t, \Delta t)$ is proportional to the time Δt and the heat difference $f_j(t)$ - $f_i(t)$. Due to this, the heat difference at node i between time $t+\Delta t$ and time t will be equal to the sum of the heat that it receives from all of its neighbors. This is expressed as:

$$f_i(t+\Delta t) - f_i(t) = \sum_{j=1}^{n} (f_j(t) - f_i(t)) W_{ij} \Delta t \tag{4}$$

Dividing Eq. (4) by Δt into both sides, and let $\Delta t \to 0$, we have

$$\frac{df_i}{dt} = W_{i,:} f - d_i f_i \tag{5}$$

In terms of matrix operations, we split the weight matrix W into 4 blocks after the Lth row and column:

$$W = \begin{bmatrix} W_{LL} & W_{LU} \\ W_{UL} & W_{UU} \end{bmatrix} \tag{6}$$

Note that $W_{U,:} f = [W_{UL}\, W_{UU}] \begin{bmatrix} f_L \\ f_U \end{bmatrix}$, and $\Delta_{UU} = D_{UU} - W_{UU}$. Here Δ is the combinatorial Laplacian which is given in the matrix form as $\Delta = D - W$ where $D = diag(d_i)$. The $diag(d_i)$ is the diagonal matrix with entries $d_i = \sum_j w_{ij}$ and $W = [w_{ij}]$ is the weight matrix.

We have a matrix form:

$$\begin{aligned} \frac{df_U}{dt} &= W_{U,:} f - D_{UU} f_U \\ &= W_{UL} f_L + W_{UU} f_U - D_{UU} f_U \\ &= W_{UL} f_L - \Delta_{UU} f_U \end{aligned} \tag{7}$$

Solving this linear differential equation which is the form of $dy/dx + Py = Q$ to find the closed form solution we have:

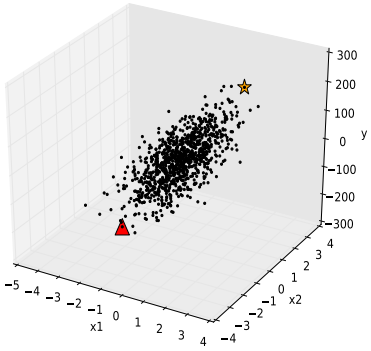
$$f_U = \Delta_{UU}^{-1} W_{UL} f_L + e^{-t\Delta_{UU}} C \tag{8}$$

This is the temperature distribution on the unlabeled nodes at time t, given the boundary condition f_L. This function is used to predict the labels for the unlabeled node. Given the initial condition $f_U|_{t=0} = f_U(0)$, $C = f_U(0) - \Delta_{UU}^{-1} W_{UL} f_L$. Note that, in the limit $t \to \infty$, $f_U = \Delta_{UU}^{-1} W_{UL} f_L$, which is the harmonic function.

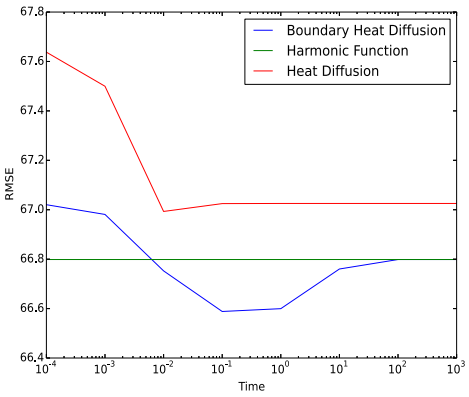
In order to intuitively interpret Eq. (8) and the heat diffusion with the boundary condition, we simulated the regression datasets with 1000 data points in two different data shapes one linear (standard deviation (σ)= 40) and another spiral. Both datasets contain two features and one target variable. The pattern of data is shown in Figs. 1(a), 1(c). We labeled two data points: a red triangle and orange star, and the rest of the data is unlabeled, which is in black. We employed the Gaussian RBF Kernel $w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ to construct the graph between these points and applied the closed-form equations for heat diffusion, harmonic function and boundary heat diffusion. Figs. 1(b), 1(d) show the performance of these algorithms. The y-axis is the Root Mean Square Error (RMSE), and the x-axis is the time. The harmonic function does not have the time component(t) in its equation in contrast to HD and BHD. We can learn from the curve that when t equals to 10^{-4}, both HD and BHD algorithms have the highest RMSE (see Figs. 1(b), 1(d)).

In the case of linear-shaped data, as time increases, HD and BHD both started to have a low RMSE. At time equals to 10^{-1}, HD started to converge (see Fig. 1(b)). There is no further reduction of the RMSE, whereas the BHD kept decreasing until the range 10^{-1} to 10^{0}. Beyond that interval, RMSE started to increase, and from 10^{2}, the BHD has a similar RMSE to harmonic function.

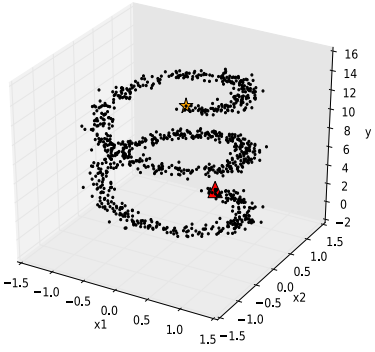
In the case of spiral-shaped data, RMSE for BHD rapidly decreases when time equals 10^{0} after that RMSE of BHD is similar to a harmonic function. HD does not change much in RMSE in these datasets and converges faster. In this dataset, we also observe for a higher value of time; the BHD will be the same as harmonic function favoring continuous or long-range propagation. Thus, BHD never looses with harmonic function because, in the infinite time stamp, BHD will ultimately become harmonic function.



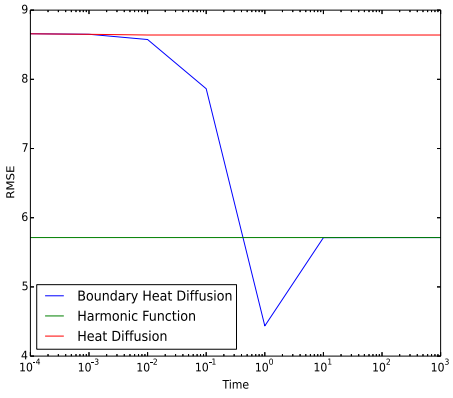
(a) Simulated linear pattern data with 1000 data points.



(b) Error curves.



(c) Simulated spiral pattern data with 1000 data points.



(d) Error curves.

Fig. 1. The error curves demonstrating the performance of different label propagation algorithm.

4.4. Computational complexity

In the solution provided by Eq. (8) we have two parts: (i) the harmonic part and (ii) the exponential part. When the graph is large, their computation will be time-consuming because both of them have a $O(n^3)$ complexity. To solve this, we took an iterative approach to compute the harmonic part provided by Zhu et al. [65], which is the same as a Random Walk with Restart (RWR) [66]. For the exponential part, we took the discrete approximations by Yang et al. [12]:

$$f(t) = \left(I - \frac{t}{M}\Delta_{UU}\right)^{M} f(0) \tag{9}$$

where I and M are the identity matrix and the number of iterations, respectively. The latter was set to 30 in conformity to [12]. t is the time. $f(0)$ is the initial temperature and $f(t)$ is the temperature at timestamp t. Specifically, after the discrete formalization of the complexity of exponential kernel in our model is given by $O(M|E|n)$ where M is the number of iterations, n is the number nodes and $|E|$ is the number of edges in the graph.

To demonstrate the time performance of our discrete approximation, we chose two different graphs (i) fully connected graph using the Gaussian kernel and (ii) sparse graph using the k-NN method. These graphs are constructed using simulated regression data with standard deviation (σ)= 100 and by varying input data points from 100 to 1000. We applied all the algorithms by using only 10% labeled data. The k in the k-NN graph is estimated by cross-validation in training sets. The performance of the BHD algorithm using 100 dedicated realizations along with other states

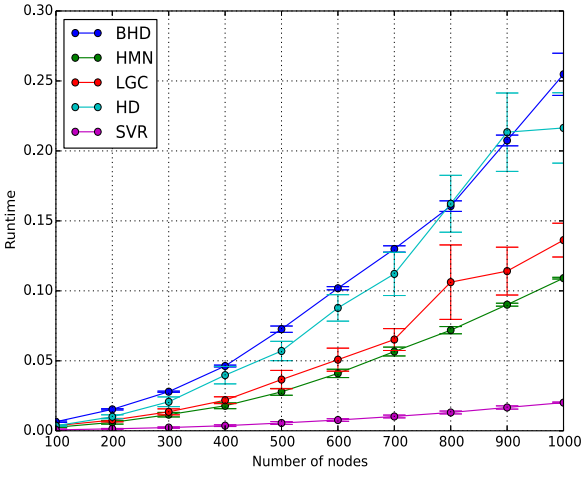


Fig. 2. Time performance in a fully connected graph.

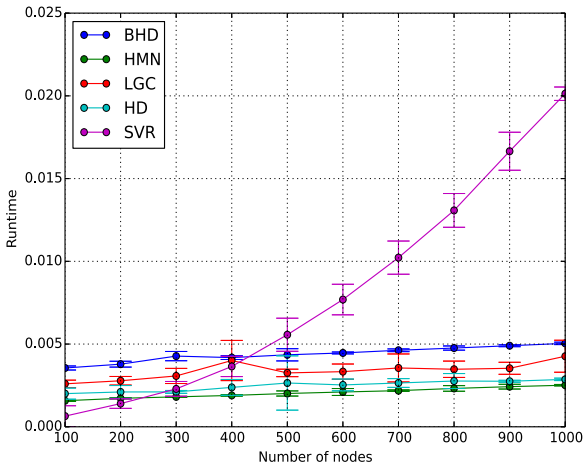


Fig. 3. Time performance in a k-NN graph.

of the art methods in one core machine is shown in Figs. 2 and 3. The x-axis is the number of nodes, the y-axis is the runtime, and the error bar is the standard deviation from 100 dedicated realizations.

In Fig. 2, we observe that by using a fully connected graph for BHD shown by a blue line, the computational cost goes rapidly high with the increase in the data points. It also holds for the other graph-based label propagation methods like HD, HMN, LGC. The non-graph-based method SVR also increases with the increase of the data points. In a fully connected graph, the matrix is dense and eventually leading the cubic time complexity $O(n^3)$, which is a worst-case even though we apply discrete approximation in a graph-based method. Thus a fully connected approach can be very time consuming for large graphs.

However, if we replace the fully connected graph by k-NN graphs where each node connects to only a few nodes, we can reduce the complexity of discrete approximation. The label propagation in such sparse graphs is computationally cheaper and faster. The time complexity, in this case, depends on k being chosen. If the chosen k is the same as the number of data points n, then it will also have the worst case. The optimum k can be chosen by using the cross-validation in training sets or by domain knowledge of the data. For small k, the graph will be sparse, which ultimately speeds up the computation time. In Fig. 3, we can see that using sparse graphs with discrete approximation; the computation overload can be reduced. We also see that all graph-based methods have very fast computation times with sparse graph implementations in comparison to non-graph-methods (SVR).

4.5. Space complexity

For a fully connected graph using Gaussian Kernel we need to store $|E|$ number of edges and n is the length of vectors for initial temperatures that means the space complexity (S) is: $S = O(|E|) + O(n) = O(n^2) + O(n)$.

In a real-world case scenario, when we have a large graph, then using a Gaussian Kernel is not efficient in terms of space. So we can replace the Gaussian Kernel by the k-NN approach. If k is chosen small then we can reduce the space complexity from $O(n^2) + O(n)$ to $O(kn) + O(n)$.

4.6. Temperature setting for initial conditions:

In order to propagate heat, we need to set the initial temperature. The initial temperature of a node is its labeled real value. These labeled nodes are then the training nodes.

Initial Temperature Setting in test set: If the quality of the network is poor, the ideal way to make inferences about the prediction of the node label values in the test set is to use the sample mean. It can be assumed that the nodes without any links are from a population with a mean μ. We can then make an inference of μ by a sample mean [67], which is used to set the initial temperature for an unlabeled node in a test set.

If the network contains valuable information for making the prediction, this initial estimation should be combined with diffusion along the edges in the network. Our method supports this property, as shown in Eq. (8), while the harmonic function ignores this initial estimation because of the continuous or global diffusion. Additionally, if the network is of poor quality, our boundary heat diffusion model has the freedom of choosing a small value of t.

5. Algorithm

The BHD has two parts harmonic and heat diffusion. For the harmonic part, we took the iterative approach provided by Zhu et al. [65]. This algorithm requires $n \times n$ transition matrix, $n \times 1$ label vector, $n \times n$ Laplacian matrix and M is the number of iterations. Once harmonic scores are determined, we need to calculate the constant C according to Eq. (8). C is obtained by subtracting an initial label score from a harmonic score. The initial label score has an initial temperature for each node. We imputed the values for the unlabeled nodes as the means of the labeled nodes. This C is the initial condition of the state vector $(n \times 1)$ for heat diffusion with boundary condition. Formally, the process is described in Algorithm 1.

Algorithm 1: Heat Diffusion with Boundary Condition for Regression Problem.

Input : The transition matrix T of size $n \times n$; initial label vector Y of size $n \times 1$; Laplacian matrix L; M is the number of iteration chosen as 30; I is the identity matrix of size $n \times n$

Output: State vector of size $n \times 1$

Initialize U = Y
repeat
  $Y^{k+1} \leftarrow TY^k$
  $Y^{k+1} \leftarrow Y^{k+1} + U$
  $Y^k = Y^{k+1}$
  $k = k + 1$
until error between Y^{k+1} and Y^k becomes sufficiently small
Initial_Temperature: Impute mean value for unlabeled nodes using labeled value
C = Initial_temperature - Y^K
State_Vector = C
t is a parameter in (0,1);
for $b = 1$ *to* M **do**
  $State\_Vector = Y^k + \left(I - \frac{t}{M}L\right) State\_Vector$
end
return *State_Vector*

6. Experiments

All the codes are written in Python and the datasets used in the experiment are available in the web.[2] From this experiment, we answer the following questions using real and synthetic datasets:

- Q1: *Parameter:* Does the parameter t affect the prediction performance of heat diffusion with boundary condition?

[2] https://github.com/timilsinamohan/SSR

Table 3
Regression Datasets.

Datasets	Domain	Number of Features	Number of Data Points
Advertisement	Business	1	200
Boston Housing	Business	13	506
Parkinson's Telemonitoring	Biomedical	16	5875
White Wine Quality	Business	10	1599
Red Wine Quality	Business	10	4898
Airfoil Self-Noise	Physical	5	1503
Bike Sharing	Social	16	17,389
3D Road Network	Engineering	4	434,874
Million Song Dataset	Business	90	515,345
Online Retail Dataset	Business	8	1,067,371

- Q2: *Accuracy:* How accurate BHD is in comparison to the state of the art label propagation algorithm?
- Q3: *Non-Label Propagation:* How well BHD performs in comparison to the state of the art non-label propagation algorithm?
- Q4: *Proportion:* How is the accuracy impacted to all the state of the art algorithms due to change of lab proportion in the graph?
- Q5: *Speed:* How fast is BHD in comparison to the state of the art algorithm?
- Q6: *Features* Does changing the features in the datasets affect the performance of BHD?
- Q7: *Performance* Does BHD has similar performance if we change the graph construction?

Real Datasets: We used ten regression datasets from different domains in our experiments. The datasets used in the experiments are shown in Table 3.

All the datasets used in the experiments are publicly available. Out of 7 datasets, 8 of the datasets [Parkinson, White wine, Red wine, Airfoil self-noise, Bike-sharing, 3D Road Network, Million Song Dataset and Online Retail Dataset] are from a UCI machine learning data repository.[3] The Boston housing data is from python open source scikit data repository.[4] Advertisement data is collected from the data repository[5] of the book "Elements of Statistical Learning" [68].

Brief descriptions of the datasets are as follows:

1. **Advertisement:** This data contains the advertising data sales (in thousands of units) for a particular product advertising budgets (in thousands of dollars) for TV, radio, and newspaper media. We use TV budgets to predict advertising sales.
2. **Boston Housing:** This data contains information collected by the U.S Census Service concerning housing in the area of Boston Mass [69]. This data has been used extensively throughout the literature to benchmark algorithms. We used this data to predict the price of the house.
3. **Parkinson Telemonitoring:** This data is composed of a range of biomedical voice measurements from 42 people with early-stage Parkinson's disease [70]. From this data, we predicted the UPDRS (Unified Parkinson's Disease Rating Scale) score of each patient.
4. **Red and White Wine Quality:** This data is composed of two different wines, i.e., red and white. These two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine [71]. From this data, we predicted the alcohol level in the wine.
5. **Airfoil Self-Noise:** This data is from NASA,[6] which comprises of different size airfoils at various wind tunnel speeds and angles of attack. From this data, we predicted the scaled sound pressure level [72].
6. **Bike Sharing:** Bike-sharing [73] is an automated bike rental system. A user can rent a bike from a particular location and return to another location by using these systems. These systems are getting popular due to their impact on traffic, health, and environmental issues. The bike-sharing systems generate data that make these systems attractive for artificial intelligence (AI) based research. The bike-sharing systems records, the duration of travel, departure, and arrival position. This property turns the bike-sharing system into a virtual sensor network. The data collected from these sensors are useful for identifying mobility in the city. From this data, we predicted the count of total rental bikes.
7. **3D Road Network:** This dataset is constructed by adding elevation information to a 2D road network in North Jutland, Denmark [74]. This dataset can be used by any applications which require to know very specific elevation information of a road network to perform task such as eco-routing, cyclist routes etc. From this data, we predicted the elevation of the road.
8. **Million Song Datasets:** The Million Song Dataset[7] is a freely-available collection of audio features and metadata for a million contemporary popular music tracks [75]. From this data, we predicted the year of the song released.
9. **Online Retail Datasets:** The Online Retail Dataset contains all the transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011 [76]. From this data, we predicted the product price of the item.

Artificial Datasets: We use six different simulated datasets in the study. Each of the datasets has 1000 data points. These datasets can be generated by Python api.[8] Brief description of the datasets are as follows:

1. **Simulated Regression Data:** This is generated from the random regression problem with input samples and the output values using standard deviation (σ)= (100).
2. **Spiral Data:** This data has a spiral shape. It is generated from the input samples with features and output values. The output values are the univariate position of the data points to the major dimension of the points in the manifold.
3. **Swiss Roll Data:** This data is generated using an algorithm provided by Marsland [77]. The algorithm generates the input samples with features and output values. The output values are the univariate position of the data points to the major dimension of the points in the manifold.
4. **Sparse Uncorrelated Data:** This data is generated by an algorithm provided by [78]. The algorithm generates a random regression problem with sparse uncorrelated design which has the input samples and the output values.
5. **Friedman Regression Data:** The data Friedman1 and Friedman2 are generated by an algorithm provided by [79,80]. The algorithm generates the input samples and the output values.

Metric: We chose the root mean square error (RMSE) to evaluate the performance of the algorithm. RMSE is a quadratic scoring

[3] https://archive.ics.uci.edu/ml/index.php
[4] https://scikit-learn.org/stable/datasets/index.html
[5] https://web.stanford.edu/~hastie/ElemStatLearn/data.html
[6] https://www.nasa.gov/
[7] http://millionsongdataset.com/
[8] https://scikit-learn.org/stable/modules/classes.html

Table 4
The average RMSE score obtained by means of 10 fold cross validation by state of the art methods. The error is the standard deviation obtained from the 10-fold cross validation.

	Advertisement	Boston Housing	Parkinson	White Wine	Red Wine	Airfoil Self Noise	Bike Sharing	3D Road Network	Million Song Datasets	Online Retail Datasets
HMN Zhu et al. [15]	12.78 ± 0.39	22.94 ± 0.38	10.68 ± 0.05	2.64 ± 0.01	2.50 ± 0.17	66.84 ± 2.64	200.48 ± 2.78	29.25 ± 0.05	103.09 ± 3.84	145.59 ± 8.90
LGC Zhou et al.[22]	14.03 ± 0.17	23.64 ± 0.20	28.30 ± 0.08	9.62 ± 0.01	9.55 ± 0.01	114.29 ± 0.12	246.40 ± 1.42	27.5 ± 0.03	73.09 ± 0.01	141.99 ± 11.52
HD Yang et al. [12]	14.41 ± 0.1	23.50 ± 0.23	29.21 ± 0.06	9.96 ± 0.01	9.86 ± 0.01	118.22 ± 0.09	251.04 ± 1.28	27.16 ± 0.03	72.02 ± 0.01	**138.89 ± 12.14**
SVR Drucker et al. [81]	**5.32 ± 2.14**	**10.96 ± 3.22**	10.85 ± 0.08	1.64 ± 0.58	1.64 ± 0.58	139.69 ± 0.51	185.68 ± 1.61	28.86 ± 0.06	**13.42 ± 3.08**	143.87 ± 9.05
BHD	5.42 ± 0.18	11.12 ± 0.33	**10.61 ± 0.03**	**1.23 ± 0.01**	**1.11 ± 0.01**	**11.33 ± 0.1**	**179.98 ± 0.95**	**26.31 ± 0.51**	70.87 ± 0.11	140.45 ± 12.32

rule that measures the average magnitude of the error. It is the square root of the average of squared differences between predictions and actual observations. The RMSE score is then given by:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{10}$$

where y_i is the observed value and $\hat{y}_i$ is the predicted value and n is the number of observations.

Q1: Parameter In this experiment, we assessed the prediction ability of our approach by varying the parameter t. The parameter t is varied from 0.0001 to 1. We report results for all regression data used in our experiments.

We applied the RBF kernel in the data points to construct the graph from these data. We varied the labeled nodes from 10% to 90%. We run the experiments and record the RMSE averaged over 100 trials for each labeled percentage for all the datasets.

From Fig. 4, we observed that when the percentage of the labeled node increases, the RMSE score decreases in all the values for the parameter t. It is because a majority of the nodes were already labeled so less heat required to label the remaining nodes. We observed that at $t = 1$, the RMSE score is minimum in comparison to other parameters, as shown by the light-green curve. It means we need maximum heat to perform diffusion in this kind of network, which in this case, is $t = 1$. This observation also means that we do not need to tune the parameter when using this algorithm. Hence, we use this default value $t = 1$ for all experiments.

Q2: Accuracy We compared our approach with state of the art label propagation algorithms namely: (i) harmonic function (HMN) [65], (ii) local and global consistency method (LGC) [22], and (iii) heat diffusion (HD) [26]. We also compared the accuracy with Support Vector Regression (SVR) using linear kernel which is a non-label propagation algorithm. It is because SVR has been chosen as a baseline to compare with the regression based label propagation method by the previous studies [14,82].

We split data with 10% for training, 90% testing, and apply the algorithms in 10 Folds cross-validation setting to record the average RMSE score.

From Table 4, we observe that the boundary-based heat diffusion has performed either at least equaling or exceeding the four state of the art methods. BHD has performed significantly better than the HMN, LGC, and HD in predicting the outcome values for Advertisement, Boston housing, White wine, Red wine, Airfoil Self Noise, Bike Sharing and 3D network datasets. However, in Parkinson's data, BHD has a marginal improvement over HMN. One of the reasons for this might be the nature of diffusion. In HMN diffusion, the information propagates infinitely favoring long-range interactions, and BHD also has a similar property. For a long-range diffusion, HMN equals to BHD, which is one of the vital property of BHD. In Advertisement and Boston Housing datasets, the/our SVR method performed marginally better than BHD. As this data has a strong linear association with outcome variable and SVR with linear kernel captures this better than BHD. In the Online retail dataset, HD performed better than other state of the art methods. In this dataset, it may be that local diffusion is favored in comparison to global diffusion and that might have affected the accuracy.

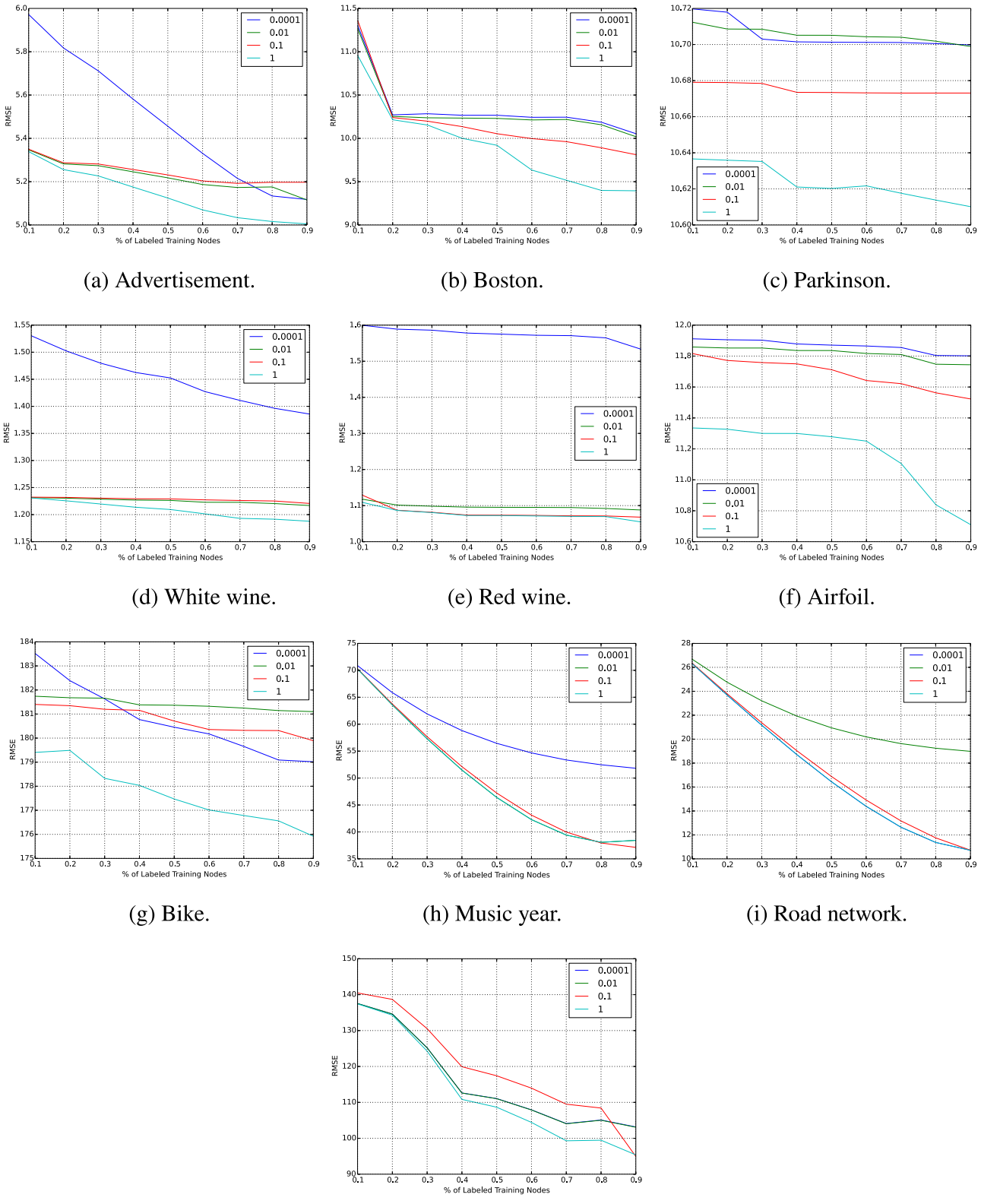
Table 5
P-values of the t-test at significance level $\alpha = 0.05$. The bold figures indicate significant p-value.

	HMN	HD	LGC	SVR
BHD (Advertisement)	**2.347e−12**	**2.56e−16**	**7.438e−16**	**1.213e−05**
BHD (Boston Housing)	**4.883e−14**	**<2.2e−16**	**3.319e−16**	0.6139
BHD (Parkinson)	**0.0002416**	**<2.2e−16**	**<2.2e−16**	**5.012e−06**
BHD (White Wine)	**4.24e−11**	**<2.2e−16**	**<2.2e−16**	0.2748
BHD (Red Wine)	**8.03e−08**	**<2.2e−16**	**<2.2e−16**	0.78
BHD (Airfoil Self Noise)	**4.116e−13**	**<2.2e−16**	**<2.2e−16**	**0.006906**
BHD (Bike Sharing)	**3.516e−13**	**<2.2e−16**	**<2.2e−16**	**0.009806**
BHD (3D Road Network)	**2.016e−11**	**<2.2e−16**	**<2.2e−16**	0.46101
BHD (Million Song)	**3.211e−11**	**<2.2e−16**	**<2.2e−16**	**<2.2e−16**
BHD (Online Retail)	**6.03e−11**	**<2.2e−16**	**<2.2e−16**	**0.00028**

We noted that the problem of SSR is more general than the SSC. In the latter case, the outcome variable is constrained to have only a finite number of possible values, whereas, in regression, the outcome variable is assumed to be continuous. Hence, LGC and HD algorithms might be more suitable for SSC tasks, whereas HMN and BHD can be applicable for SSR tasks, as shown by our results. Zhu et al. [15] showed that the Gaussian Fields apply to both SSR and SSC problems.

Different algorithms shared the same random trials. So, we could perform statistical tests. We applied the paired t-test to find out if there is a significant difference in the 10-fold cross-validation results between BHD and other states of the art methods using a significance (α) level of 0.05. The p-values of the t-test are reported in Table 5. We found that there is a significant difference between the prediction performed by BHD with other states of the art methods (p-values <0.05). In Boston housing, white wine, red wine and 3D road network dataset, we observed the p-value higher than 0.05 between BHD and SVR method. It suggests that there is no significant difference between the ten fold predictions between these methods.

We further assess whether the differences in performance across the compared algorithms are statistically significant in the ten folds cross-validation results. We used a Friedman Nemenyi test [35,83,84], which compares the algorithms in a pairwise way. For this task, algorithms are ranked according to their prediction accuracy, so that the best performing algorithm is ranked at the top, the second-best is at the second position, and so on. The null hypothesis for this test is that all methods have equal performance. The alternative hypothesis is that there is a difference in performance between the methods. The p-value of the test is demonstrated in Table 6.



(a) Advertisement. (b) Boston. (c) Parkinson.

(d) White wine. (e) Red wine. (f) Airfoil.

(g) Bike. (h) Music year. (i) Road network.

(j) Online retail.

Fig. 4. Impact of parameter t in regression datasets. X-axis is the percentage of labeled data. Y-axis is the RMSE score.

From Table 6, we come to the following conclusions. First, the Friedman Nemenyi test suggests that the proposed BHD performs significantly better against the majority of the other state of the art methods. In Bike Sharing datasets, our BHD has a significant difference with all baseline methods. Similarly, for 3D Road Network, Airfoil, and Parkinson datasets, BHD is significantly different to all the other methods. As for Online Retail Datasets, BHD is only significantly different to HMN methods,

Table 6
P-values of the Friedman and Nemenyi statistical test over all the 10 fold cross-validation at significance level $\alpha = 0.05$. The bold figures indicate significant p-value.

		HD	HMN	LGC	SVR
Advertisement	HMN	**0.0377**	–	–	–
	LGC	0.6184	0.6184	–	–
	SVR	**7.2e−06**	0.2109	**0.0037**	–
	BHD	**7.4e−06**	0.2109	**0.0035**	1.0000
Boston Housing	HMN	**0.03770**	–	–	–
	LGC	0.61845	0.61845	–	–
	SVR	**7.7e−07**	0.08083	**0.00072**	–
	BHD	**5.9e−05**	0.43571	**0.01599**	0.91532
Parkinson	HMN	**0.00021**	–	–	–
	LGC	0.61845	**0.03770**	–	–
	SVR	**0.03770**	0.61845	0.61845	–
	BHD	**1.5e−07**	0.61845	**0.00021**	**0.03770**
White Wine	HMN	**0.0248**	–	–	–
	LGC	0.6184	0.5261	–	–
	SVR	**3.5e−06**	0.2109	**0.0022**	–
	BHD	**3.0e−05**	0.4357	**0.0101**	0.9932
Red Wine	HMN	**0.03770**	–	–	–
	LGC	0.61845	0.61845	–	–
	SVR	**7.7e−07**	0.08083	**0.00072**	–
	BHD	**5.9e−05**	0.43571	**0.01599**	0.91532
Airfoil	HMN	**0.0377**	–	–	–
	LGC	0.6184	0.6184	–	–
	SVR	0.2758	0.9153	0.9800	–
	BHD	**3.5e−06**	0.1571	**0.0022**	**0.0160**
Bike Sharing	HMN	0.61845	–	–	–
	LGC	0.61845	**0.03770**	–	–
	SVR	**0.03770**	0.61845	**0.00021**	–
	BHD	**0.00021**	**0.03770**	**1.5e−07**	**0.61845**
3D Road Network	HMN	**0.03770**	–	–	–
	LGC	0.61845	0.61845	–	–
	SVR	0.61845	**0.00021**	**0.03770**	–
	BHD	**0.03770**	**1.5e−07**	**0.00021**	0.61845
Million Song Datasets	HMN	**0.03770**	–	–	–
	LGC	0.61845	0.61845	–	–
	SVR	**0.03770**	**1.5e−07**	**0.00021**	–
	BHD	0.61845	**0.00021**	**0.03770**	0.61845
Online Retail Datasets	HMN	**1.5e−05**	–	–	–
	LGC	0.3513	0.0248	–	–
	SVR	**0.0022**	0.7899	0.3513	–
	BHD	0.4357	**0.0160**	0.9999	0.2758

but not its performance wrt. SVR and LGC. In that dataset, HD outperformed all methods and is significantly different to them in terms of accuracy. With respect to the Advertisement and Boston Housing datasets, both SVR and BHD are significantly different in performance to HD and LGC, and between them, they are not significantly different. Whereas in the Million Song Datasets, SVR is significantly different to all other methods. We conjecture that this data must have strong linear relationships with the target value, which is better captured by SVR.

Q3: Comparison with non-label propagation algorithms Apart from the state of the art label propagation algorithm, we also compared the accuracy of the BHD with other non-label propagation algorithms, namely: Multi-scheme semi-supervised regression approach (MSSRA), [57,85], Semi-Supervised Random Forest [86], Co-Training style semi-supervised regression COREG, K-Nearest Neighbor (KNN) regression, and Linear regression. The result of the comparison is shown in Table 7.

We observed that in Boston Housing, Parkinson, White Wine, and Red Wine data MSSRA outperformed all the other methods. However, our BHD outperforms in Advertisement and Online Retail data. It is also clear from the table that there is no single method that beats all the other methods. In the majority of cases, MSSRA is performing better. One of the reasons for that is MSSRA is training on the arbitrary number of regressors whose predictions are filtered through a minimum range criterion for distinguishing the most accurate regressor to apply in the test sets. However, other methods only rely on a single regressor function. One of the caveats of MSSRA is that it trains multiple regressors, which might be computationally expensive to handle large datasets without running in multiple core machines.

Q4: Proportion To demonstrate the performance of all the algorithms, we used different percentages of the labeled data in training sets ranging from 10% to 90%. For each percentage of the labeled data, we ran 10 Fold cross-validation. The performance of the algorithm in different datasets is shown in Table 8.

We observe that SVR with linear kernel performed quite well regardless the fraction of labels for "Advertisement", "Boston housing" and "Million songs" datasets. One of the reasons for that is a linear association between predictors and target variables which is better captured by SVR than other graph-based label propagation methods. Another important observation about SVR is that with more labeled data its performance improved in comparison to the graph based methods particularly for White and Red wine datasets. It may be due to adding more labeled data decreases the marginal error in SVR which help to improve the predictions which is being trained on more examples.

Our BHD method performed better in "Parkinson", "Airfoil", "Bike sharing", "3D Road network" in all the label proportion. Whereas using fewer labeled training data in "Red wine" 10% and "White wine" (10%,20%,30% and 40%) BHD outperformed the other methods. In a "Million Song Datasets" BHD outperform to all the graph-based method by using only 10%, 20%, 30% and 40% labeled dataset. It means that BHD makes use of the graph structure to exploit the information of unlabeled data for the regression problem and improves the performance.

In an "Online retail" datasets, we observed that the HD performed better than rest of the algorithms regardless the fraction of labels. The heat diffusion has the property that the weights decay faster and it penalizes more heavily the shorter paths keeping the heat in a local neighborhood. This property may be feasible for this datasets which might have influence for better prediction than other algorithms.

Q5: Speed We performed the run time performance of the algorithm across all our regression datasets. Table 9 shows the computational time for the algorithms. We observed that in all the datasets SVR outperforms the state of the art methods. It may be due to the following reasons: (i) SVR with linear kernel does not require to construct graph from the data points which saves the extra computational time unlike other graph based method. (ii) Furthermore, training SVR with a linear kernel requires only one parameter (regularization parameter) to be optimized. On the other hand, some of the graph-based propagation methods require to do grid search to find optimum parameters which might have consume more computation time.

Q6: Features In this experiment, we assess the performance of BHD in various artificial datasets by varying features from [5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]. For each feature, we performed 10-fold cross-validation by dividing the data into 10% for training and 90% for testing and record the average RMSE score.

From Fig. 5, we observe that there are marginal changes in the RMSE when we change the number of features. In a simulated regression data (red curve (SRD)), changing the features from 5 to 10, we see a slight reduction of RMSE. It is maybe the fact that the features we added might have contributed to improve the predictions. Whereas, when we increase the features > 40 in LR, we do not see much reduction in RMSE. As we increase the number of irrelevant features, the signal in the data becomes weaker and has no contribution towards prediction. In such a case, we can reduce the number of irrelevant features by using

Table 7
The average RMSE score obtained by means of 10 fold cross validation by non-label propagation methods. The error is the standard deviation obtained from the 10-fold cross validation.

	Advertisement	Boston Housing	Parkinson	White Wine	Red Wine	Airfoil Self Noise	Bike Sharing	3D Road Network	Million Song	Online Retail
MSSRA	**5.42 ± 0.46**	**7.76 ± 0.01**	**3.62 ± 0.51**	**0.58 ± 0.52**	**0.63 ± 0.45**	12.05 ± 0.14	147.65 ± 0.42	29.12 ± 0.87	10.11 ± 0.42	149.21 ± 19.02
Random Forest	5.48 ± 0.14	10.99 ± 0.41	10.23 ± 0.58	1.15 ± 0.45	1.01 ± 0.24	7.81 ± 0.18	155.90 ± 0.81	17.78 ± 0.01	**9.38 ± 0.01**	143.66 ± 19.02
KNN regression	5.65 ± 0.14	9.78 ± 0.24	8.37 ± 0.27	1.25 ± 0.35	1.10 ± 0.42	**6.77 ± 0.47**	150.81 ± 0.43	**3.65 ± 0.04**	12.01 ± 0.48	169.91 ± 18.86
Linear regression	3.27 ± 0.32	38.67 ± 0.15	38.67 ± 0.24	0.61 ± 0.22	1.11 ± 0.14	8.29 ± 0.15	180.81 ± 0.21	18.37 ± 0.01	85.55 ± 0.01	143.78 ± 9.03
COREG	6.36 ± 0.49	16.34 ± 0.21	4.11 ± 0.29	0.78 ± 0.82	0.90 ± 0.14	5.57 ± 0.58	**135.41 ± 0.25**	29.31 ± 0.51	80.55 ± 0.01	150.55 ± 0.01
BHD	**5.42 ± 0.18**	11.12 ± 0.33	10.61 ± 0.03	1.23 ± 0.01	1.11 ± 0.01	11.33 ± 0.1	179.98 ± 0.95	26.31 ± 0.51	70.87 ± 0.11	**140.45 ± 12.32**

Table 8
Performance of the different state of the art algorithms with different label proportions.

Datasets	Algorithms	10%	20%	30%	40%	50%	60%	70%	80%	90%
Advertisement	HMN	12.22 ± 0.51	10.31 ± 0.43	9.05 ± 0.67	7.98 ± 0.65	7.13 ± 0.78	6.62 ± 0.88	5.27 ± 0.67	5.27 ± 0.88	5.34 ± 1.56
	LGC	14.06 ± 0.20	13.15 ± 0.16	12.26 ± 0.34	11.22 ± 0.48	10.32 ± 0.67	9.54 ± 0.68	7.99 ± 0.88	7.45 ± 0.96	6.88 ± 1.37
	HD	14.44 ± 0.16	13.89 ± 0.19	13.33 ± 0.26	12.65 ± 0.41	12.09 ± 0.51	11.58 ± 0.59	10.31 ± 0.85	10.03 ± 0.82	9.50 ± 1.17
	SVR	**5.32 ± 1.52**	**4.50 ± 0.99**	**4.24 ± 1.10**	**4.20 ± 0.95**	**4.19 ± 1.19**	**4.18 ± 1.70**	**4.18 ± 2.50**	**4.10 ± 1.27**	**4.05 ± 2.06**
	BHD	5.42 ± 0.17	5.25 ± 0.18	5.22 ± 0.20	5.17 ± 0.37	5.12 ± 0.49	5.06 ± 0.52	5.05 ± 0.55	5.03 ± 0.66	5.01 ± 1.07
Boston housing	HMN	22.24 ± 0.53	20.83 ± 0.52	19.97 ± 0.52	18.77 ± 0.72	18.00 ± 0.68	17.29 ± 0.95	16.63 ± 1.09	16.04 ± 1.11	15.11 ± 1.72
	LGC	23.40 ± 0.28	22.51 ± 0.37	21.83 ± 0.41	20.88 ± 0.63	20.19 ± 0.75	19.40 ± 1.02	18.58 ± 1.19	17.84 ± 1.16	16.68 ± 1.57
	HD	23.55 ± 0.23	22.80 ± 0.37	22.21 ± 0.37	21.33 ± 0.57	20.71 ± 0.75	19.92 ± 1.01	19.22 ± 1.13	18.35 ± 1.24	17.32 ± 1.34
	SVR	**8.89 ± 2.12**	**7.86 ± 1.18**	**7.39 ± 0.63**	**7.08 ± 1.31**	**7.01 ± 1.63**	**6.84 ± 5.03**	**6.66 ± 1.54**	**6.49 ± 1.59**	**6.51 ± 2.23**
	BHD	11.40 ± 0.21	11.24 ± 0.29	11.17 ± 0.36	11.03 ± 0.55	10.90 ± 0.57	10.77 ± 0.75	10.48 ± 0.96	10.35 ± 0.89	10.20 ± 1.05
Parkinson	HMN	10.68 ± 0.05	10.61 ± 0.04	10.59 ± 0.05	10.57 ± 0.06	10.58 ± 0.09	10.55 ± 0.11	10.54 ± 0.12	10.60 ± 0.13	10.55 ± 0.22
	LGC	28.34 ± 0.07	25.77 ± 0.10	23.30 ± 0.10	20.90 ± 0.15	18.62 ± 0.14	16.47 ± 0.15	14.60 ± 0.21	13.16 ± 0.17	12.07 ± 0.29
	HD	29.25 ± 0.05	27.55 ± 0.08	25.91 ± 0.09	24.26 ± 0.13	22.65 ± 0.13	21.03 ± 0.16	19.46 ± 0.21	18.03 ± 0.21	16.47 ± 0.35
	SVR	10.89 ± 0.08	10.77 ± 0.05	10.76 ± 0.07	10.70 ± 0.09	10.71 ± 0.10	10.64 ± 0.13	10.65 ± 0.16	10.68 ± 0.14	10.63 ± 0.32
	BHD	**10.64 ± 0.04**	**10.63 ± 0.04**	**10.63 ± 0.05**	**10.62 ± 0.06**	**10.62 ± 0.08**	**10.62 ± 0.11**	**10.61 ± 0.12**	**10.61 ± 0.13**	**10.60 ± 0.21**
White wine	HMN	2.66 ± 0.06	1.95 ± 0.05	1.75 ± 0.08	1.68 ± 0.08	1.62 ± 0.08	1.56 ± 0.09	1.47 ± 0.09	1.43 ± 0.13	1.39 ± 0.17
	LGC	9.63 ± 0.01	8.67 ± 0.01	7.74 ± 0.02	6.81 ± 0.01	5.93 ± 0.02	5.07 ± 0.02	4.28 ± 0.04	3.62 ± 0.08	3.16 ± 0.11
	HD	9.98 ± 0.01	9.36 ± 0.01	8.76 ± 0.02	8.15 ± 0.02	7.55 ± 0.02	6.93 ± 0.02	6.31 ± 0.04	5.73 ± 0.03	5.15 ± 0.05
	SVR	1.64 ± 0.49	1.61 ± 0.15	1.60 ± 0.34	1.59 ± 1.31	1.58 ± 0.28	1.57 ± 0.57	1.36 ± 0.41	1.35 ± 0.65	1.32 ± 0.49
	BHD	**1.23 ± 0.01**	**1.22 ± 0.01**	**1.21 ± 0.01**	**1.20 ± 0.01**	**1.20 ± 0.01**	**1.20 ± 0.02**	**1.19 ± 0.03**	**1.19 ± 0.05**	**1.18 ± 0.05**
Red wine	HMN	2.46 ± 0.20	1.92 ± 0.19	1.70 ± 0.14	1.65 ± 0.12	1.60 ± 0.13	1.47 ± 0.12	1.49 ± 0.15	1.45 ± 0.21	1.45 ± 0.37
	LGC	9.56 ± 0.02	8.64 ± 0.03	7.75 ± 0.03	6.87 ± 0.03	6.01 ± 0.03	5.19 ± 0.04	4.44 ± 0.06	3.78 ± 0.10	3.31 ± 0.21
	HD	9.87 ± 0.02	9.27 ± 0.03	8.67 ± 0.02	8.07 ± 0.03	7.47 ± 0.03	6.88 ± 0.06	6.29 ± 0.08	5.68 ± 0.06	5.11 ± 0.15
	SVR	1.64 ± 0.39	1.27 ± 0.25	1.16 ± 0.14	1.33 ± 0.36	1.39 ± 0.40	1.23 ± 0.33	1.31 ± 0.41	1.19 ± 0.23	1.28 ± 0.30
	BHD	**1.11 ± 0.01**	**1.11 ± 0.02**	**1.10 ± 0.03**	**1.09 ± 0.03**	**1.09 ± 0.03**	**1.09 ± 0.05**	**1.09 ± 0.07**	**1.09 ± 0.10**	**1.08 ± 0.11**
Airfoil	HMN	52.06 ± 5.59	27.77 ± 3.61	20.80 ± 2.40	15.96 ± 3.46	14.57 ± 3.37	13.48 ± 2.65	12.62 ± 2.45	11.83 ± 2.37	10.75 ± 3.66
	LGC	114.24 ± 0.16	103.46 ± 0.33	92.82 ± 0.31	81.84 ± 0.42	71.34 ± 0.67	61.33 ± 0.77	51.15 ± 0.95	41.56 ± 1.20	32.30 ± 1.48
	HD	115.44 ± 0.17	105.80 ± 0.36	96.42 ± 0.44	86.65 ± 0.46	77.29 ± 0.73	68.66 ± 0.89	59.46 ± 1.09	51.04 ± 1.67	43.00 ± 1.52
	SVR	124.95 ± 0.22	122.66 ± 0.85	120.02 ± 0.51	120.85 ± 0.52	107.70 ± 0.46	105.42 ± 0.17	100.42 ± 0.17	87.31 ± 0.85	76.71 ± 0.07
	BHD	**11.33 ± 0.15**	**11.32 ± 0.28**	**11.29 ± 0.37**	**11.29 ± 0.45**	**11.27 ± 0.64**	**11.25 ± 0.81**	**11.10 ± 1.02**	**10.83 ± 0.96**	**10.70 ± 1.19**
Bike sharing	HMN	244.57 ± 0.84	227.65 ± 1.80	210.56 ± 2.38	193.10 ± 2.68	175.49 ± 3.83	158.91 ± 3.67	139.37 ± 3.18	118.39 ± 4.43	95.06 ± 5.49
	LGC	259.73 ± 0.41	257.94 ± 1.20	255.62 ± 1.51	253.04 ± 1.80	250.14 ± 1.81	247.94 ± 2.12	245.51 ± 2.53	242.94 ± 2.68	240.58 ± 4.80
	HD	253.12 ± 0.70	244.55 ± 1.43	235.18 ± 1.85	225.51 ± 2.17	215.36 ± 2.72	205.70 ± 2.93	195.19 ± 2.69	183.77 ± 2.87	171.61 ± 4.18
	SVR	185.26 ± 0.82	184.75 ± 1.59	183.43 ± 1.85	182.21 ± 1.83	180.80 ± 1.61	179.66 ± 1.99	179.06 ± 2.59	178.48 ± 3.29	177.59 ± 4.61
	BHD	**179.98 ± 0.68**	**177.39 ± 0.53**	**171.63 ± 0.83**	**170.77 ± 1.32**	**169.45 ± 1.84**	**163.17 ± 2.10**	**156.65 ± 2.37**	**150.09 ± 3.00**	**142.72 ± 3.38**
3D Road network	HMN	29.26 ± 0.05	29.47 ± 0.08	29.54 ± 0.08	29.52 ± 0.08	29.48 ± 0.09	29.39 ± 0.08	29.31 ± 0.08	29.24 ± 0.10	29.07 ± 0.15
	LGC	27.53 ± 0.03	26.10 ± 0.03	24.71 ± 0.04	23.34 ± 0.04	22.00 ± 0.04	20.70 ± 0.05	19.45 ± 0.05	18.26 ± 0.07	17.12 ± 0.07
	HD	27.16 ± 0.04	25.38 ± 0.03	23.65 ± 0.04	21.94 ± 0.05	20.30 ± 0.05	18.72 ± 0.05	17.22 ± 0.06	15.85 ± 0.08	14.58 ± 0.09
	SVR	28.86 ± 0.07	25.47 ± 0.41	23.42 ± 0.50	19.42 ± 0.38	19.37 ± 0.59	19.36 ± 0.42	19.27 ± 0.22	19.05 ± 0.51	18.01 ± 0.35
	BHD	**26.30 ± 0.51**	**23.69 ± 0.08**	**21.16 ± 0.04**	**18.72 ± 0.05**	**16.44 ± 0.05**	**14.39 ± 0.06**	**12.64 ± 0.07**	**11.37 ± 0.10**	**10.71 ± 0.12**
Million song	HMN	103.55 ± 3.98	92.74 ± 3.07	86.25 ± 1.74	82.45 ± 0.85	80.29 ± 0.64	78.96 ± 0.28	78.22 ± 0.12	77.74 ± 0.13	77.38 ± 0.13
	LGC	73.09 ± 0.01	69.18 ± 0.01	65.47 ± 0.02	62.00 ± 0.03	58.83 ± 0.03	55.98 ± 0.03	53.54 ± 0.04	51.53 ± 0.11	50.03 ± 0.11
	HD	71.89 ± 0.01	66.89 ± 0.01	62.24 ± 0.03	58.02 ± 0.04	54.36 ± 0.03	51.33 ± 0.04	49.10 ± 0.05	47.73 ± 0.14	47.30 ± 0.16
	SVR	**15.80 ± 4.49**	**15.52 ± 2.43**	**15.32 ± 2.58**	**15.01 ± 9.51**	**14.51 ± 8.08**	**13.59 ± 7.07**	**12.55 ± 2.02**	**11.55 ± 8.18**	**10.11 ± 5.08**
	BHD	70.88 ± 0.01	65.86 ± 0.01	61.91 ± 0.01	58.82 ± 0.02	56.46 ± 0.02	54.67 ± 0.02	53.36 ± 0.05	52.43 ± 0.06	51.83 ± 0.08
Online retail	HMN	145.59 ± 8.90	146.26 ± 11.83	143.56 ± 13.73	139.44 ± 13.01	137.71 ± 16.07	138.96 ± 12.95	139.66 ± 17.85	139.80 ± 40.12	123.42 ± 56.20
	LGC	139.99 ± 11.52	136.36 ± 15.16	129.23 ± 14.95	120.61 ± 17.41	115.62 ± 18.38	112.09 ± 16.62	109.44 ± 19.47	111.67 ± 33.48	98.43 ± 35.49
	HD	**138.88 ± 12.15**	**134.54 ± 15.82**	**125.92 ± 15.59**	**115.28 ± 18.87**	**109.44 ± 20.28**	**104.53 ± 18.40**	**100.14 ± 20.66**	**101.73 ± 33.01**	**88.92 ± 30.10**
	SVR	143.87 ± 9.01	142.14 ± 12.79	139.72 ± 13.59	137.33 ± 14.13	134.07 ± 14.66	135.02 ± 12.87	138.14 ± 16.79	139.26 ± 39.66	123.54 ± 56.14
	BHD	140.47 ± 12.31	138.68 ± 14.93	130.53 ± 16.82	119.93 ± 17.85	117.41 ± 20.78	113.96 ± 16.83	109.50 ± 21.07	108.42 ± 33.07	94.95 ± 31.18

Table 9
Run time comparisons between state of the art algorithms on various regression datasets. s and m refers to seconds and minutes respectively.

	Advertisement	Boston Housing	Parkinson	White Wine	Red Wine	Airfoil Self Noise	Bike Sharing	3D Road Network	Million Song Datasets	Online Retail Datasets
HMN Zhu et al. [15]	0.011 s	0.081 s	20.86 s	11.4 s	1.06 s	0.30 s	23.46 s	25.46 s	28.16 s	34.16 s
LGC Zhou et al.[22]	0.0064 s	0.053	7.96 s	5.61 s	0.56 s	0.12 s	0.28 s	30.28 s	25.46 s	27.25s
HD Yang et al. [12]	0.0067 s	0.03 s	6.72 s	4.28 s	0.40 s	0.25 s	26.02s	28.02 s	30.12 s	31.12 s
SVR Drucker et al. [81]	**0.0012 s**	**0.0035 s**	**0.014 s**	**0.04 s**	**0.13 s**	**0.0075 s**	**0.25 s**	**10.14 s**	**14.14 s**	**22.78 s**
BHD	0.015 s	0.10 s	7.00 s	4.69 s	0.50 s	0.39 s	67.44 s	28.36 m	33.31 m	41.40 m

a dimensional reduction method and construct the graph using only relevant features. In Friedman1 (F1) data represented by the purple line, we see that adding features 5,10, 20 and 30 has helped to reduce RMSE. It means the added feature is reasonable and helps to strengthen the prediction performance. However, beyond 30 features, the RMSE score did not reduce. It means the feature added is irrelevant, and added features have no contribution towards prediction.

In geometrical manifold data like a spiral (blue line (SD)) and swiss roll (sky blue (SRL)) are created using three crucial

Table 10
Performance of BHD using fully connected and kNN graph based on Time and RMSE. s and m refers to seconds and minutes respectively.

Datasets	Time taken using fully connected graph	Time taken using kNN graph	RMSE fully connected graph	RMSE kNN graph
Advertisement	0.01 s	**0.00756 s**	**5.42 ± 0.18**	6.30 ± 0.30
Boston Housing	0.10 s	**0.022133 s**	**11.12 ± 0.33**	12.12 ± 0.85
Parkinson's	7.00 s	**1.56 s**	**10.61 ± 0.03**	10.88 ± 0.04
White Wine	4.69 s	**1.21 s**	**1.23 ± 0.01**	1.31 ± 0.03
Red Wine	0.50 s	**0.13 s**	**1.11 ± 0.01**	1.13 ± 0.01
Airfoil	0.39 s	**0.26 s**	11.33 ± 0.1	**10.16 ± 0.10**
Bike Sharing	67.44 s	**7.24 s**	179.98 ± 0.95	**179.90 ± 0.38**
Road Network	28.36 m	**2.65 s**	**26.31 ± 0.51**	26.99 ± 0.05
Million Song Datasets	33.3 m	**5.8 s**	70.87 ± 0.11	**69.67 ± 0.01**
Online Retail	41.40 m	**5.6 s**	140.45 ± 12.32	**139.45 ± 11.11**

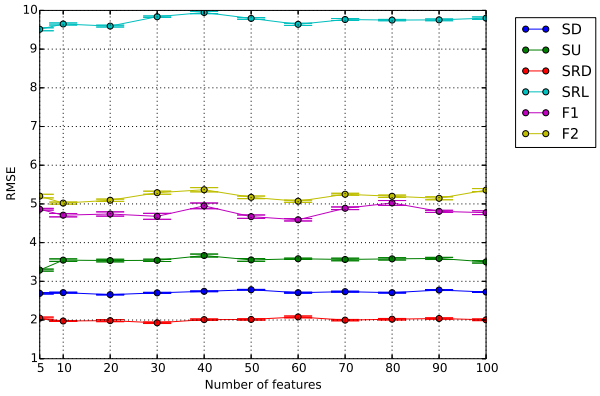


Fig. 5. Performance of BHD on artificial datasets by varying features. *X*-axis is the number of features and *Y*-axis is the average root mean square error using each features in a 10 fold cross validation setting. SD: Spiral Data (blue); SU: Sparse Uncorrelated (green); SRD: Simulated Regression Data (red); SRL: Swiss Roll Data (sky blue); F1:Friedman 1 (pink); F2: Friedman2 (yellow).

features. While adding irrelevant features in a spiral dataset, there is not much of a change in RMSE. Whereas, in SRL, we see RMSE marginally increased by adding the additional irrelevant features making the signal in the data weaker. We also observe a slight decrease in RMSE after features > 40 and remain constant throughout the additional irrelevant features. One of the reasons for this variation of RMSE between SRL and SD might be their topological difference, which might have affected their prediction performances.

Q7:Performance We assess the performance of BHD in terms of time and accuracy by means of two different generic graph construction methods: (i) fully connected graph using Gaussian Kernel; and (ii) k-NN graph. Table 10 shows the results of the comparison using 10% labeled data in a 10 Fold cross validation settings. In our datasets, we observed that in a fully connected graph, the computation cost increases when the size of the graph also increases. We can see for 3D Road Network, Million Song, and Online Retail datasets the computation time increased very rapidly. It is due to the cubic complexity experienced by an algorithm by using a fully connected graph. However, if we replace the fully connected by the k-NN graph, we can see that the computation time is reduced significantly.

The parameter k controls the density of the graph. For a small k, the number of edges in the graph will be small, which allows us to speed up the computational time. Small k may result in disconnected graphs, although it is not a problem for BHD if each connected component has some labeled points. We chose k by cross-validation of training nodes. We also observe from Table 10 that it is almost similar or a marginal difference in accuracy between these two graph-construction methods. In Advertisement, Boston Housing, Parkinson's, White Wine, Red Wine, and 3D Road Network datasets, the fully connected graph has outperformed by a small margin to k-NN. It may be due to effective weight learning of the edges between the nodes using a fully connected graph. The graph needs to be weighted so that similar nodes have large edge weights between them. The fully connected graph captures this property with the expense of high computation cost than a k-NN graph.

7. Conclusion

We presented the application of boundary heat diffusion in an SSR problem. We applied these algorithms in different domains, such as business, biomedical, physical, and social domain data. The main idea of our method is to assign a node with the initial temperatures. The initialized temperatures act as the boundary and diffuse the heat in the network. The advantages of our algorithm are:

1. Accuracy: it outperforms or equals the state of the art algorithms in label propagation on graph-based semi-supervised regression tasks: [Tables 4, 8].
2. Parameter Free: It has just one parameter with the effective default value 1. [Fig. 4].

We employed BHD for the real-valued labels in a graph constructed from manifold data. The method outperformed some of the states of the art methods, but in some data, the support vector regression (SVR) method performed better. One of the reasons for this might be a strong linear association between an outcome variable and predictors, and SVR with linear kernel captures this better than BHD. One way to handle this problem is to construct a better graph from the manifold data. We used the Gaussian kernel, which provided us with a fully connected graph. Of course, better graphs can be constructed if one can define better distance functions, connectivity, and edge weights. It is another critical challenge in a graph-based semi-supervised regression problem.

One of the significant strengths of the heat diffusion with boundary condition is computational complexity. We showed that the boundary-based heat diffusion could be computed using discrete approximation. It will have an advantage in the scalability issues in a large graph constructed using k-NN method because the complexity is linear in the number of edges in the graph. This property makes the method suitable for bigger graph regression problems. We believe that our proposed approach provides a simple but effective method to estimate the real values for performing semi-supervised graph-based regression. In our future work, we would like to extend our BHD method to a semi-supervised classification problem in different label correlation problems such as homogeneous, heterogeneous, and mixed labeled prediction tasks.

CRediT authorship contribution statement

Mohan Timilsina: Conducted experiments, Formal analysis, Writing - original draft. **Alejandro Figueroa:** Guidance, Writing - review & editing. **Mathieu d'Aquin:** Guidance, Writing - review & editing. **Haixuan Yang:** Guidance, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to acknowledge Science Foundation Ireland (SFI/12/RC/2289_P2) for funding this research.

References

[1] Corinna Cortes, Mehryar Mohri, On transductive regression, Adv. Neural Inf. Process. Syst. 19 (2006) 305–312.
[2] Avrim Blum, Tom Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, 1998, pp. 92–100.
[3] Zhi-Hua Zhou, Ming Li, Tri-training: Exploiting unlabeled data using three classifiers, IEEE Trans. Knowl. Data Eng. 17 (11) (2005) 1529–1541.
[4] Thorsten Joachims, Transductive inference for text classification using support vector machines, in: Icml, Vol. 99, 1999, pp. 200–209.
[5] Daoqiang Zhang, Zhi-Hua Zhou, Songcan Chen, Semi-supervised dimensionality reduction, in: Proceedings of the 2007 SIAM International Conference on Data Mining, SIAM, 2007, pp. 629–634.
[6] Masashi Sugiyama, Tsuyoshi Idé, Shinichi Nakajima, Jun Sese, Semi-supervised local Fisher discriminant analysis for dimensionality reduction, Mach. Learn. 78 (1–2) (2010) 35.
[7] Daniel Palomera, Alejandro Figueroa, Leveraging linguistic traits and semi-supervised learning to single out informational content across how-to community question-answering archives, Inform. Sci. 381 (2017) 20–32.
[8] Georgios Kostopoulos, Stamatis Karlos, Sotiris Kotsiantis, Omiros Ragos, Semi-supervised regression: A recent review, J. Intell. Fuzzy Systems 35 (2) (2018) 1483–1500.
[9] Avrim Blum, Shuchi Chawla, Learning from Labeled and Unlabeled Data Using Graph Mincuts, figshare, 2001.
[10] Xiaojin Jerry Zhu, Semi-Supervised Learning Literature Survey, Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 2005.
[11] Fan Chung, The heat kernel as the pagerank of a graph, Proc. Natl. Acad. Sci. 104 (50) (2007) 19735–19740.
[12] Haixuan Yang, Irwin King, Michael R. Lyu, Diffusionrank: a possible penicillin for web spamming, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2007, pp. 431–438.
[13] Risi Imre Kondor, John Lafferty, Diffusion kernels on graphs and other discrete structures, in: Proceedings of the 19th International Conference on Machine Learning, Vol. 2002, 2002, pp. 315–322.
[14] Xiaojin Zhu, Andrew Goldberg, Semi-Supervised Regression with Order Preferences, Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 2006.
[15] Xiaojin Zhu, Zoubin Ghahramani, John D. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions, in: Proceedings of the 20th International Conference on Machine Learning, ICML-03, 2003, pp. 912–919.
[16] Mingrui Wu, Bernhard Schölkopf, Transductive classification via local learning regularization, in: Artificial Intelligence and Statistics, 2007, pp. 628–635.
[17] Mikhail Belkin, Partha Niyogi, Vikas Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (Nov) (2006) 2399–2434.
[18] Ling Xie, Miguel A. Carreira-Perpinán, Shawn Newsam, Semi-supervised regression with temporal image sequences, in: 2010 IEEE International Conference on Image Processing, IEEE, 2010, pp. 2637–2640.
[19] Gauthier Doquire, Michel Verleysen, A graph Laplacian based approach to semi-supervised feature selection for regression problems, Neurocomputing 121 (2013) 5–13.
[20] Mingbo Zhao, Tommy WS Chow, Zhou Wu, Zhao Zhang, Bing Li, Learning from normalized local and global discriminative information for semi-supervised regression and dimensionality reduction, Inform. Sci. 324 (2015) 286–309.
[21] Baohuai Sheng, Hancan Zhu, The convergence rate of semi-supervised regression with quadratic loss, Appl. Math. Comput. 321 (2018) 11–24.
[22] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, Bernhard Schölkopf, Learning with local and global consistency, in: Adv. Neural Inf. Process. Syst., 2004, pp. 321–328.
[23] Guoxian Yu, Guoji Zhang, Zhiwen Yu, Carlotta Domeniconi, Jane You, Guoqiang Han, Semi-supervised ensemble classification in subspaces, Appl. Soft Comput. 12 (5) (2012) 1511–1522.
[24] Shumeet Baluja, Rohan Seth, D Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, Mohamed Aly, Video suggestion and discovery for youtube: taking random walks through the view graph, in: Proceedings of the 17th International Conference on World Wide Web, ACM, 2008, pp. 895–904.
[25] Partha Pratim Talukdar, Koby Crammer, New regularized algorithms for transductive learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2009, pp. 442–457.
[26] Haixuan Yang, Michael R. Lyu, Irwin King, A volume-based heat-diffusion classifier, IEEE Trans. Syst. Man Cybern. B 39 (2) (2009) 417–430.
[27] Mohan Timilsina, Mathieu d'Aquin, Haixuan Yang, Heat diffusion approach for scientific impact analysis in social media, Soc. Netw. Anal. Min. 9 (1) (2019) 16.
[28] Mohan Timilsina, Haixuan Yang, Ratnesh Sahay, Dietrich Rebholz-Schuhmann, Predicting links between tumor samples and genes using 2-Layered graph based diffusion approach, BMC Bioinf. 20 (1) (2019) 462.
[29] Mohan Timilsina, Meera Tandan, Mathieu d'Aquin, Haixuan Yang, Discovering links between side effects and drugs using a diffusion based method, Sci. Rep. 9 (1) (2019) 1–10.
[30] Xia Chen, Guoxian Yu, Qiaoyu Tan, Jun Wang, Weighted samples based semi-supervised classification, Appl. Soft Comput. 79 (2019) 46–58.
[31] Fabricio A. Breve, Liang Zhao, Marcos G. Quiles, Particle competition and cooperation for semi-supervised learning with label noise, Neurocomputing 160 (2015) 63–72.
[32] Karl Øyvind Mikalsen, Cristina Soguero-Ruiz, Filippo Maria Bianchi, Robert Jenssen, Noisy multi-label semi-supervised dimensionality reduction, Pattern Recognit. 90 (2019) 257–270.
[33] Jun Wang, Guangjun Yao, Guoxian Yu, Semi-supervised classification by discriminative regularization, Appl. Soft Comput. 58 (2017) 245–255.
[34] Hiroshi Ohno, Neural network-based transductive regression model, Appl. Soft Comput. 84 (2019) 105682.
[35] João Roberto Bertini Junior, Maria do Carmo Nicoletti, Liang Zhao, Attribute-based decision graphs: a framework for multiclass data classification, Neural Netw. 85 (2017) 69–84.
[36] Thomas N. Kipf, Max Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
[37] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Ondrej Chum, Label propagation for deep semi-supervised learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5070–5079.
[38] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, Maosong Sun, Graph neural networks: A review of methods and applications, 2018, arXiv preprint arXiv:1812.08434.
[39] Ziwei Zhang, Peng Cui, Wenwu Zhu, Deep learning on graphs: A survey, IEEE Trans. Knowl. Data Eng. (2020).
[40] Qimai Li, Xiao-Ming Wu, Han Liu, Xiaotong Zhang, Zhichao Guan, Label efficient semi-supervised learning via graph filtering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9582–9591.
[41] Simon Bourigault, Cedric Lagnier, Sylvain Lamprier, Ludovic Denoyer, Patrick Gallinari, Learning social network embeddings for predicting information diffusion, in: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, 2014, pp. 393–402.
[42] Dorina Thanou, Xiaowen Dong, Daniel Kressner, Pascal Frossard, Learning heat diffusion graphs, IEEE Trans. Signal Inf. Process. Netw. 3 (3) (2017) 484–499.
[43] Awad H. Al-Mohy, Nicholas J. Higham, Computing the action of the matrix exponential, with an application to exponential integrators, SIAM J. Sci. Comput. 33 (2) (2011) 488–511.
[44] Yuto Yamaguchi, Christos Faloutsos, Hiroyuki Kitagawa, Omni-prop: Seamless node classification on arbitrary label correlation, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[45] Avrim Blum, John Lafferty, Mugizi Robert Rwebangira, Rajashekar Reddy, Semi-supervised learning using randomized mincuts, in: Proceedings of the Twenty-First International Conference on Machine Learning, ACM, 2004, p. 13.
[46] Wolfgang Gatterbauer, Stephan Günnemann, Danai Koutra, Christos Faloutsos, Linearized and single-pass belief propagation, Proc. VLDB Endow. 8 (5) (2015) 581–592.

[47] Larry Wasserman, John D. Lafferty, Statistical analysis of semi-supervised regression, in: Adv. Neural Inf. Process. Syst., 2008, pp. 801–808.
[48] Ahmed El Alaoui, Xiang Cheng, Aaditya Ramdas, Martin J Wainwright, Michael I Jordan, Asymptotic behavior of l_p -based laplacian regularization in semi-supervised learning, in: Conference on Learning Theory, 2016, pp. 879–906.
[49] Xiaoyi Mai, Romain Couillet, A random matrix analysis and improvement of semi-supervised learning for large dimensional data, J. Mach. Learn. Res. 19 (1) (2018) 3074–3100.
[50] Edith Cohen, Semi-supervised learning on graphs through reach and distance diffusion, 2016, arXiv preprint arXiv:1603.09064.
[51] Frank Lin, William W. Cohen, The multirank bootstrap algorithm: Self-supervised political blog classification and ranking using semi-supervised link classification., in: ICWSM, 2008.
[52] Peter A Lofgren, Siddhartha Banerjee, Ashish Goel, C Seshadhri, FAST-PPR: scaling personalized pagerank estimation for large graphs, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 1436–1445.
[53] Nir Rosenfeld, Amir Globerson, Semi-supervised learning with competitive infection models, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2018, pp. 336–346.
[54] Fan R.K. Chung, Fan Chung Graham, Spectral Graph Theory, (92) American Mathematical Soc., 1997.
[55] Sujith Ravi, Qiming Diao, Large scale distributed semi-supervised learning using streaming approximation, in: Artificial Intelligence and Statistics, 2016, pp. 519–528.
[56] Max Budninskiy, Ameera Abdelaziz, Yiying Tong, Mathieu Desbrun, Laplacian-optimized diffusion for semi-supervised learning, Comput. Aided Geom. Design (2020) 101864.
[57] Georgios Kostopoulos, Sotiris Kotsiantis, Nikos Fazakis, Giannis Koutsonikos, Christos Pierrakeas, A semi-supervised regression algorithm for grade prediction of students in distance learning courses, Int. J. Artif. Intell. Tools 28 (04) (2019) 1940001.
[58] Zhi-Hua Zhou, Ming Li, Semi-Supervised Regression with Co-Training, in: IJCAI, Vol. 5, 2005, pp. 908–913.
[59] Meng Wang, Xian-Sheng Hua, Yan Song, Li-Rong Dai, Hong-Jiang Zhang, Semi-supervised kernel regression, in: Sixth International Conference on Data Mining, ICDM'06, IEEE, 2006, pp. 1130–1135.
[60] Judea Pearl, Reverend Bayes On Inference Engines: A Distributed Hierarchical Approach, Cognitive Systems Laboratory, School of Engineering and Applied Science, 1982.
[61] Omiros Papaspiliopoulos, Giacomo Zanella, A note on MCMC for nested multilevel regression models via belief propagation, 2017, arXiv preprint arXiv:1704.06064.
[62] Mugizi Robert Rwebangira, John Lafferty, Local Linear Semi-Supervised Regression, Vol. 15213, School of Computer Science Carnegie Mellon University, Pittsburgh, PA, 2009.
[63] Hao Ma, Haixuan Yang, Michael R. Lyu, Irwin King, Mining social networks using heat diffusion processes for marketing candidates selection, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, ACM, 2008, pp. 233–242.
[64] Daniela Nitsch, Joana P Gonçalves, Fabian Ojeda, Bart De Moor, Yves Moreau, Candidate gene prioritization by network analysis of differential expression using machine learning approaches, BMC Bioinf. 11 (1) (2010) 460.
[65] Xiaojin Zhu, Zoubin Ghahramani, Learning from Labeled and Unlabeled Data with Label Propagation, Technical Report, Citeseer, 2002.
[66] Hanghang Tong, Christos Faloutsos, Jia-Yu Pan, Fast random walk with restart and its applications, in: Sixth International Conference on Data Mining, ICDM'06, IEEE, 2006, pp. 613–622.
[67] Irwin Miller, Marylees Miller, John E. Freund, John E. Freund's Mathematical Statistics, Prentice Hall, 1999.
[68] Trevor Hastie, Robert Tibshirani, Jerome Friedman, James Franklin, The elements of statistical learning: data mining, inference and prediction, Math. Intell. 27 (2) (2005) 83–85.
[69] A.Myrick Freeman, Hedonic prices, property values and measuring environmental benefits: a survey of the issues, in: Measurement in Public Choice, Springer, 1981, pp. 13–32.
[70] Athanasios Tsanas, Max A Little, Patrick E McSharry, Lorraine O Ramig, Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests, IEEE Trans. Biomed. Eng. 57 (4) (2009) 884–893.
[71] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, José Reis, Modeling wine preferences by data mining from physicochemical properties, Decis. Support Syst. 47 (4) (2009) 547–553.
[72] Thomas F. Brooks, D.S.tuart Pope, Michael A. Marcolini, Airfoil self-noise and prediction, 1989.
[73] Hadi Fanaee-T, Joao Gama, Event labeling combining ensemble detectors and background knowledge, Prog. Artif. Intell. 2 (2–3) (2014) 113–127.
[74] Manohar Kaul, Bin Yang, Christian S. Jensen, Building accurate 3d spatial networks to enable next generation intelligent transportation systems, in: 2013 IEEE 14th International Conference on Mobile Data Management, Vol. 1, IEEE, 2013, pp. 137–146.
[75] Brian McFee, Thierry Bertin-Mahieux, Daniel PW Ellis, Gert RG Lanckriet, The million song dataset challenge, in: Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 909–916.
[76] Daqing Chen, Sai Laing Sain, Kun Guo, Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining, J. Database Mark. Customer Strateg. Manage. 19 (3) (2012) 197–208.
[77] Stephen Marsland, Machine Learning: an Algorithmic Perspective, CRC press, 2015.
[78] Gilles Celeux, Mohammed El Anbari, Jean-Michel Marin, Christian P Robert, et al., Regularization in regression: comparing Bayesian and frequentist methods in a poorly informative situation, Bayesian Anal. 7 (2) (2012) 477–502.
[79] Jerome H. Friedman, Multivariate adaptive regression splines, Ann. Stat. (1991) 1–67.
[80] Leo Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[81] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, Vladimir Vapnik, Support vector regression machines, in: Adv. Neural Inf. Process. Syst., 1997, pp. 155–161.
[82] Andrew B. Goldberg, Xiaojin Zhu, Stephen Wright, Dissimilarity in graph-based semi-supervised classification, in: Artificial Intelligence and Statistics, 2007, pp. 155–162.
[83] Janez Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.
[84] Qingyao Wu, Jian Chen, Shen-Shyang Ho, Xutao Li, Huaqing Min, Chao Han, Multi-label regularized generative model for semi-supervised collective classification in large-scale networks, Big Data Res. 2 (4) (2015) 187–201.
[85] Nikos Fazakis, Stamatis Karlos, Sotiris Kotsiantis, Kyriakos Sgarbas, A multi-scheme semi-supervised regression approach, Pattern Recognit. Lett. 125 (2019) 758–765.
[86] Jurica Levatić, Michelangelo Ceci, Dragi Kocev, Sašo Džeroski, Semi-supervised classification trees, J. Intell. Inf. Syst. 49 (3) (2017) 461–486.